



# ADELI

ASSOCIATION POUR LE DÉVELOPPEMENT DE LA LOGIQUE INFORMATIQUE

## LA LETTRE DE L'ADELI et

### L'ADELIEN

N° 14, Novembre 1990

#### Sommaire:

- 1-"Merise, son vocabulaire, sa logique" (Georges ICKOWICZ)
  - 2-"Bâtir son AGL ? Pourquoi pas...Oui, mais comment ?  
Commission "Les préalables à la mise en place d'un AGL"  
(Pierre FISCHOF, Philippe GAUCHET, Alain COULON)
  - 3-"Pourquoi, en France exploite-t-on si lentement et si  
partiellement les apports de J.D.Warnier ?" (Alain COULON)
  - 5-" Warnier: sagesse et questionnements d'un scientifique  
humaniste" (Pierre FISCHOF)
  - 6-"Warnier: un message simple et pourtant immortel..." (Pierre  
FISCHOF)
  - 7-Présentation des 4mes Journées "Pratique des méthodes et  
outils logiciels d'aide à la conception de systèmes  
d'information" (Henri HABRIAS)
  - 8-Les citations de l'Adélien (Henri HABRIAS)
  - 9-Nous avons lu (Henri HABRIAS)
  - 10-"Langage C et approche "orientée" objet (Philippe DRIX) avec  
l'aimable autorisation de l'auteur du livre "Langage C.Norme C.  
Vers une approche orientée objet (Masson éditeur) et de  
l'éditeur
  - 11-"Mais que fait donc l'Adéli ? (Comité de l'Adéli)
  - 12-"Logical Construction of Software" (Chand, Yadav)" extrait des Communications  
of the ACM
- Rédaction:H.Habrias,IUT, 3 rue MI Joffre 44041 Nantes Cedex 01  
tel:(33) 40 30 60 52,fax:(33)40 30 60 01  
Imprimé par l'IUT de Nantes

## LES CITATIONS DE L'ADELIEN:

"Quand on a beaucoup de moyens, on perd ses moyens"  
(auteur inconnu)

"Aimez-vous le poulet rôti ? Le comprenez-vous"  
(Jean Cocteau, France-Culture)

"Si l'on examine un cochon d'Inde, on s'aperçoit avec stupeur que ce n'est pas un cochon et qu'il n'est pas d'Inde. Seul le "D" est authentique"  
(Cavanna cité par R.Moreno dans sa TBA.;voir plus loin)

"Concept. Tresseur de pont ou sceptre de con ?"  
"Norme morne"  
(Michel Leiris, Langage tangage ou ce que les mots me disent", Gallimard, 1985)

"Voici une invention bien curieuse que l'on vient de présenter à l'Institut, c'est le nouveau boomerang français dont le bois est taillé de telle sorte que l'instrument, une fois jeté sur l'adversaire, ne revient pas à celui qui l'a lancé. On évite ainsi tout risque d'accident." G. de Pawlowsky

"Je refuse d'appartenir à un club qui m'accepterait pour membre"  
(Groucho Marx)

"Je ne suis ni pour ni contre, bien ou contraire" (A.Allais)

"Un couteau sans lame auquel il manque le manche" (G-C Lichtenberg)

## UNE VRAIE HISTOIRE BELGE:

"-Ne saviez-vous pas que les Français utilisent une méthode qui s'appelle MERISE ?

-Non. Mais ça veut dire quoi ?

"-Méthode Eprouvée pour Ralentir Indéfiniment la Sortie des Etudes"

**Notre cours d'anglais pour informaticiens:**

**" LISP : Lots of Irriting Silly Parenthesis"**  
(auteur inconnu)

**"Mathematicians are like Frenchmen: whatever you say to them they translate into their own language and forthwith it is something entirely different..."**

(Goethe, cité par B.Meyer dans "Introduction to the theory of programming language" ,Prentice-Hall éditeur)

**NOUS AVONS LU:**

**-Le premier article de la série "Practical Programmer" dans les Communications of the Acm (october 90):**

il s'agit de présenter des solutions pratiques tirées de l'expérience relatives à la construction de logiciels.

**-P.Jaulent: "Génie logiciel. Les méthodes: Sadt,Sa,E-A,SA-RT,SREM,SYS-PO,OOD,HOOD" , Armand Colin, 1990**

Extra pour faire le tour de la question. Voir aussi l'excellente présentation de M.Lissandre dans les Actes des 4mes journées LIANA.

**-B.Meyer: "Introduction to the theory of Programming Languages", Prentice-Hall,1990**

**-Roland Moreno: "Théorie du Bordel Ambient" (Pierre Belfond),1990**

Hors de propos ,n'est-ce pas !

**-Dans le dernier N° d'Afcet Interfaces, les résultats d'une enquête sur le Génie logiciel**

**PRESENTATION DES 4MES JOURNEES "METHODES ET OUTILS  
LOGICIELS  
D'AIDE A LA CONCEPTION DES SYSTEMES D'INFORMATION  
Henri HABRIAS**

nb: on peut se procurer les actes (700 pages) en les commandant à l'IUT  
département Informatique, 3 rue Ml Joffre 44041 Nantes Cedex 01

Après les journées de septembre 87 ,88,et 89 le LIANA, laboratoire  
d'informatique appliquée de l'IUT de Nantes, présente les 4mes journées  
qui pourraient avoir pour titre:

***"Méthodes sans frontière".***

Sans frontières en effet car sont traitées des méthodes intéressant aussi  
bien des "informaticiens de gestion" que des "informaticiens industriels",  
aussi bien des applications "par lot" que des applications "temps réel",  
aussi bien des applications avec parallélisme que sans parallélisme.

Sans frontière en effet car tout le "cycle de vie du logiciel" est couvert  
par les exposés.

Sans frontière en effet car diverses écoles de pensée s'expriment.

Sans frontières en effet car les conférenciers viennent de l'Université et  
des entreprises et beaucoup travaillent de part et d'autre de cette  
frontière.

Sans frontière en effet car les conférenciers viennent de divers pays.

Sans frontière en effet car les méthodes présentées ne sont pas  
seulement d'origine française.

Sans frontière, en effet car ces journées réunissent comme les années précédentes des "professionnels" et des enseignants.

Cette année, elles comprennent des cours :

- un cours sur la méthode JSD avec passage à plusieurs langages "classiques" , avec parallélisme et "orientés objets"
- un cours sur la méthode HOOD (Hierarchical object oriented design) définie à la demande de l'agence spatiale européenne
- un cours sur la méthode de spécification formelle "Me too" créée dans le cadre du projet Alvey en Grande-Bretagne
- un cours sur la démarche et l'outil ISW
- un cours sur la méthode Sys\_P\_D

Elles comprennent aussi des exposés, des démonstrations de logiciels et des communications.

M.Lissandre (Société Véridatas) introduit ces journées par un panorama des méthodes de "conception temps réel" en s'appuyant sur un exemple. Son exposé illustre remarquablement bien l'esprit de ces journées:

- illustrer les propos par des exemples d'applications afin d'éviter la "langue de bois" chère aux informaticiens
- analyser les méthodes et outils du marché
- dégager les principes théoriques sous-jacents.

Il traite l'exemple avec "SA/RT" et ses variantes (Ward et Mellor, Hatley et Pirbhai, ESML...), avec LDS, SADT, HOOD, JSD, EPOS et pour les approches synchrones, SAGA.

Nous reconnaissons avoir été séduits par les écrits de Michaël Jackson et de John Cameron, sans doute parce qu'ils étaient pour nous un bon miroir, nous renvoyant notre pensée en la précisant. Aussi nous sommes reconnaissant à M.Jackson d'avoir présenté une conférence lors des journées de 1987 et de 1989. Le lecteur trouvera des textes en français sur la méthode Jackson de conception de système dans les Actes des 2mes et 3mes Journées. Le hasard a voulu que Gwin Jones (NEWI Polytechnic, Pays de Galles) qui est auteur d'un livre sur JSP (la méthode de

programmation de Jackson, proche de LCP de J.D.Warnier) et qui prépare une thèse sur JSD, soit en poste dans un établissement universitaire Gallois avec lequel notre I.U.T. collabore dans le cadre des échanges Erasmus. R.G.Jones est un excellent pédagogue et praticien de JSD. C'est avec lui que nous avons lancé l'idée de traiter sur un cas le passage d'une spécification JSD à l'écriture en différents langages. Le cas traité est le cas "Widget". Le texte de ce cas se trouve dans le livre de M.Jackson et dans celui de J.Cameron. Nous remercions les auteurs pour nous avoir autorisé à utiliser ce cas.

G.Jones traite de l'implantation en langage séquentiel orienté objet (Turbo-Pascal 5.5. de Borland) et en un langage parallèle (Pascal Plus).

En 1987, B.Meyer nous fournissait dans son texte : "Les objets arrivent", un aperçu de son livre sur la programmation "objet". En 1989, Ph.Drix (ESEO, Angers) nous faisait un excellent cours sur la "conception de logiciel orientée objet". Cette année, il a bien voulu traiter le cas "Widget" en l'implantant avec le langage Eiffel. Il s'est servi de la spécification J.S.D., mais la méthode J.S.D. n'utilisant pas l'héritage comme moyen de classification des objets, il a suivi une méthode en s'inspirant de Yourdon, Booch et Meyer. Il montre la démarche en laissant apparaître les hésitations et les retours en arrière. Cela lui a paru préférable à une présentation académique et idéalisée". Ph.Drix a bien compris l'esprit de ces journées !

Il utilise les principes de la programmation "contractuelle" exposés par B.Meyer. Ph.Drix nous fournit l'ensemble du codage en Eiffel. Le lecteur dispose ainsi d'un excellent document de travail.

En 1989, M.Gauthier (Université de Limoges), auteur d'un important livre sur ADA, avait fait une conférence sur " *Programmation Objet? ADA ? rien à f..... des querelles de chapelle !*", J.M.Filloque et R.Ogor nous avaient fourni un texte sur " *ADA, langage d'implémentation pour la méthode J.S.D.*". Cette année, M.Gauthier a bien voulu traiter le cas "Widget".

Avec M.Barbier, R.Romphoux, G.Mitillon et J.Meynard nous travaillons à la mise au point de la méthode "MOON" qui fait la synthèse entre JSD et l'approche statique de type "bases de données". Dans ces actes, M.Barbier

et R.Rampoux proposent une solution en Cobol Microsoft au cas "Widget".

Robert Cuesta (Université de Manchester) a programmé des analyseurs lexicaux en Smalltalk dans le cadre d'un stage effectué en 89 au Liana. Il a bien voulu participer à notre projet. Il nous fait part de ses remarques et nous fournit le code en Smalltalk/ V 286.

Remarquons que Drix, Gauthier et Cuesta n'avaient pas de formation en JSD . Cependant ils ont pu facilement comprendre la spécification fournie...ce qui montre l'intérêt de JSD.

Remarquons aussi que les auteurs n'ayant pas lu les ouvrages de Jackson et de Cameron, n'ont pas eu connaissance du pourquoi de choix délibérés de la méthode J.S.D. C'est en tenant compte de cette remarque qu'il faut lire ce que dit M.Gauthier de l'hypothèse de files infinies en JSD: "*faible de la méthode JSD*". Or voici ce qu'écrit M.Jackson page 36 de son livre:

*"But implementation is an integral part of JSD, ....A satisfactory method must not only deliver a specification. It must also provide a negotiable path from that specification to an executable system....*

*Another reason for dealing explicitly with implementation techniques in JSD ...is that we have relegated to the implementation stage many decisions which traditionally come much earlier in the development. So we need to demonstrate that these decisions have not been ignored or made harder.."*

M.Jackson reconnaît dans son livre l'influence des premiers travaux sur les processus séquentiels communicants (CSP) de Hoare. Sridhar et Hoare ont étudié l'utilisation de CSP comme base théorique de JSD en l'appliquant aux études de cas de Jackson. Yeung et Topping du Staffordshire Polytechnic (R.U.) présentent l'automatisation du processus de traduction d'un texte en notation JSD en un texte en notation CSP.

Breiteneder et Mück nous proposent dans le cadre de l' "approche base de données" classique, une démarche de conception interactive de schéma de base de données et leur outil YADD (Yet Another Database Designer !).

A la suite d'un appel d'offre en 1987 de l'Agence Spatiale Européenne est née la méthode HODD (Hierarchical Object Oriented Design). M.Laf de CISI-Ingénierie a réalisé avec Hypercord un outil support. Il nous

présente HOOD et cet outil nommé Hyperhood ainsi que leur utilisation dans le projet DIVA au Groupe de Reconnaissance Acoustique du GERDSM au Brusuc près de Toulon.

7.

Une utilisation de la méthode Jackson nous est présentée par Ph.Caillared, B.Soleiman et J.P.Gervat (Institut d'Informatique Industrielle de Brest). Il s'agit d'une application de classification d'objets physiques. la modélisation est faite avec JSD et par graphes de Pétri. Le codage a été fait en Ada.

O.Barros (Université de Santiago, Chili) présente une approche de définition de système orientée objet.

F.Brissoud (IMAG, Grenoble), dans le cadre du projet "Aristote", propose une étude générale de la prise en compte des associations dans différentes approches de modélisation de systèmes d'information et les caractéristiques d'un modèle à objets.

En 1989, G.Laffitte (Insee) nous a présenté une approche formelle de la construction des logiciels ainsi que l'outil "B" de J.R.Abriol, approche et outil qu'il a utilisé pour la préparation des logiciels de dépouillement du recensement général de la population de 1990. Cette année, nous nous sommes intéressé à "Me too", une méthode et un langage de spécification basé sur les concepts mathématiques d'ensemble et de fonction (en fait proche de la méthode NIAM qui fait l'objet d'autres présentations dans ces journées). Ce langage est exécutable. La méthode utilise la spécification formelle et le prototypage rapide, mettant plus l'accent sur la validation par test que par la preuve. Des exemples d'applications sont disponibles. Un des auteurs des travaux et du livre sur "Me too", Valeria Jones a bien voulu nous présenter en 25 pages les principes, la notation et un exemple d'application de la méthode. Nous avons aussi été autorisés à publier une spécification d'un système de vente parue dans l'ICL Technical Journal.

M.Theys de l'Université de Bruxelles nous propose une méthode de développement et de maquettage de système temps réel. Il s'appuie sur un exemple de P.Ward.



S.Sabbagh (Bull-Cedrag) présente Sesame, un générateur d'interface en langage naturel pour bases de données relationnelles.

P.Kokol et ses collègues de l'Université de Maribor (Yougoslavie) nous présentent l'A.G.L. "PIA-Case" qui transforme une spécification opérationnelle JSP en programme.

C.Sibertin-Blanc (Université de Toulouse 1) et R.Bastide (Techlog, Toulouse) nous présentent la conception par objets de systèmes parallèles. Ils utilisent les Réseaux de Pétri à objets. Avec de tels réseaux, les places peuvent contenir des valeurs qui sont soit des constantes soit des objets, lesquels, à la différence des jetons, sont identifiables.

Dans un deuxième article, ils nous présentent un exemple complet d'utilisation du formalisme des objets coopératifs pour la modélisation d'un atelier flexible.

P.Jaulent (Société Microtools) expose la méthode SYS\_P\_O (Systems Project Object) selon une présentation anthropomorphique.

En 1988, nous avons consacré une journée à la méthode et au formalisme NIAM pour la conception de bases de données. Cette année, deux communications portent sur des outils associés à la méthode.

ISW de la société ITS de Bruxelles est un AGL utilisant les diagrammes de flots de données, les réseaux de Pétri et la méthode NIAM. La cohérence entre les différents niveaux d'abstraction est assurée. Une étude de cas complète a été préparée pour ces journées. Elle illustre parfaitement la méthode et les remarquables fonctionnalités de l'outil.

L'outil RIDL, autre support de la méthode NIAM, est présenté par ses concepteurs, une équipe de l'Université de Tilburg (Pays-Bas). La place de RIDL dans le projet européen Sprite est décrite.

R.Stemper et une équipe de l'Université de Twente (Pays-Bas) cherchent à spécifier un système social comme un système de normes. Ils considèrent que les langages de spécification existants sont sous-tendus par une position philosophique qui fait qu'est formalisée une vue du monde

comme une réalité objective. Mais les systèmes sociaux sont construits par leurs participants. Aussi introduit-il deux postulats:

- il n'y a pas de réalité sans agent
- l'agent ne connaît le monde que par les actions.

Basé sur ces deux postulats, un formalisme a été créé pour représenter des systèmes de normes sociales.

Dans ce système, le sens est vu comme une relation entre signe et comportement. Les propos sont illustrés par une étude de cas: le cas IFIP déjà utilisé dans les journées de 1988.

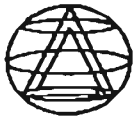
Limonde, Mille et Niel (IPCI et INSA de Lyon) nous donnent un exemple de construction d'ateliers flexibles et de leur environnement en utilisant l'atelier de génie logiciel (AGL) Teamwork, support de la méthode Yourdon.

Alain Griffault (Nakache SA), nous présente MEC, un outil de validation pour les processus communicants. Cet outil est issu d'une collaboration entre le laboratoire LABRI associé au CNRS et la société Nakache. il est basé sur le modèle ARNOLD-NIVAT dans lequel les processus sont représentés par des systèmes de transitions enrichis par des marques liées aux états et des marques liées aux transitions. L'auteur illustre son propos par un exemple de contrôle de ressources: suivi d'une trajectoire qui lui permet de montrer les différentes étapes de validation de spécification avec le produit MEC

n.b. Notre propos concerne l'ensemble des textes reçus à la date de rédaction de cette introduction.

Nantes, le 10 août 90

Henri Habrias



# ADELI

## MERISE.

### SON VOCABULAIRE, SA LOGIQUE

#### INTRODUCTION

Le formalisme de Merise peut être considéré comme clair en ce qui concerne les données.

Il n'en est pas de même pour les traitements : il nous a donc paru utile de resituer les mots et la logique de Merise avec le souci de mettre en évidence des liens formels entre des mots, et des relations entre concepts.

#### 1. RETOUR SUR LE CYCLE D'ABSTRACTION

Le cycle est défini dans Merise par trois niveaux :

**Niveau conceptuel.**

On pose la question : quel ? Que veut-on ? quels sont les objectifs ?  
On répond en cherchant à identifier les grandes fonctions de l'entreprise et on ébauche une définition d'ensemble en décrivant les processus et les opérations.

**Niveau organisationnel.**

On affine et complète le niveau précédent en répondant aux questions : qui? où? quand?

On identifie les acteurs, les postes de travail, les lieux, les moments, les traitements manuels ou informatiques.

**Niveau opérationnel.**

On répond à la question : comment? de quelle façon ?

Mais on s'intéresse uniquement au système d'information : tout ce qui concerne le système opérant a été vu au niveau organisationnel.



## 2. CINEMATIQUE DU SYSTEME TOTAL.

Elle est décrite par la circulation des flux entre les postes de travail, les acteurs, et la circulation des flux à l'intérieur du système d'information.

Au niveau opérationnel on ne considère que les flux de données.

Le niveau organisationnel décrit les flux réels et les interfaces entre :

- . le système opérant.
- . le système d'information.
- . le système de pilotage.

## 3. EVENEMENT ET FLUX.

La notion d'événement prête à confusion . Prenons un exemple :

Des factures sont éditées (Phase A) puis vérifiées (Phase B).

Si nous appelons ces deux événements : EDITION et VERIFICATION nous ne voyons, pas au niveau formel, qu'il s'agit dans l'un et l'autre cas de factures.

Si nous appelons ces deux événements FACTURES nous voyons au niveau formel qu'un même flux concerne deux phases mais nous ne pouvons plus dire qu'il s'agit (formellement) d'événements différents.

Pour conserver la logique de Merise on pourrait dire qu'un événement est composé d'une condition temporelle et d'une chose qui est décrite sans référence au temps, qui conserve son identité dans le temps, que nous appellerons FLUX.

De plus on pourrait compléter la définition d'un événement en associant un nom indiquant l'origine des flux : machine, acteur interne ou externe, etc. On aura donc .

**Evénement = condition temporelle + flux + origine.**

La condition temporelle s'apparente à la condition logique

( IF (condition logique) THEN ....) : de même que la condition logique peut déclencher ou non un traitement à l'intérieur de l'ordinateur de même la condition temporelle déclenche ou non une action.

Si nous acceptons ces modifications, nous faisons apparaître les liens formels décrits au niveau des phases et des procédures entre les trois systèmes à l'intérieur du système total, mais également les liaisons entre les flux de données et les autres flux.



## 4. CONSEQUENCE POSSIBLE AU NIVEAU CONCEPTUEL.

Processus et operation sont définis dans Merise en tenant compte de la notion d'événement.

il semble qu'on puisse les définir uniquement à partir de la notion de flux.

C'est ce que nous ferons pour les raisons suivantes :

- . Les discussions relatives au temps peuvent encombrer le niveau conceptuel avec des questions de détail alors qu'à ce niveau des orientations majeures sont prises.
- . La notion de flux est concrète et claire.
- . Les questions relatives au temps (celui du futur système) sont toutes traitées au niveau organisationnel de façon cohérente.
- . On obtient une définition formellement claire des différences et des liens entre : processus, procédure, opération, phase.

## 5. STRUCTURE DU VOCABULAIRE ET CYCLE D'ABSTRACTION.

### 5.1 Formalisme de présentation.

Un exemple permettra de le préciser.

Soit :

```
SOCIETE
  DEPARTEMENT
    SERVICE
  DOMAINE
    PROCESSUS
```

signifie : dans la définition de SOCIETE entrent les définitions de DEPARTEMENT et de DOMAINE. Dans celle de DEPARTEMENT et de DOMAINE entrent respectivement les définitions de SERVICE et de PROCESSUS.

Mais il peut signifier, aussi, qu'une chose est contenue dans une autre. Ce sera le cas si un service appartient à un département et un seul. Ce ne sera pas le cas si un même processus est présent dans plusieurs domaines.



## 5.2 Vocabulaire et cycle d'abstraction.

Niveau conceptuel.

GROUPE

ENTREPRISE

DEPARTEMENT

SERVICE

DOMAINE

PROCESSUS

OPERATION

FLUX

Niveau organisationnel.

DOMAINE

APPLICATION

PROCEDURE

PHASE

TACHE MANUELLE

PROGRAMME

FICHER

ENTITE

ECRAN

ENTITE

ETAT

ENTITE

SYNCHRONISATION

EVENEMENT

CONDITION TEMPORELLE

FLUX

ACTEUR

POSTE DE TRAVAIL

PERIODICITE

DUREE



Niveau opérationnel.

PROGRAMME

PROGRAMME

FICHIERS

ETAT

RUBRIQUE

ECRAN

RUBRIQUE

MODULE

MODULE

RUBRIQUE

## 6. SENS ET COHERENCE FORMELLE.

Il n'est pas question ici de redéfinir tous les mots, mais d'attirer l'attention sur le sens de certains mots, et leurs relations, en vue de constituer le système des mots, le concept contenu dans Merise, relatif à la modélisation du système d'information.

Niveau conceptuel :

FLUX :

nous avons préféré à ce niveau le mot FLUX au mot EVENEMENT

OPERATION :

une opération se déduit d'une PHASE en ne conservant que la notion de flux donc en éliminant les notions d'acteurs, d'espace et de temps.

D'où : à plusieurs phases peut correspondre une seule opération.

PROCESSUS :

se déduit d'une procédure comme une opération d'une phase. Du temps on ne conserve que la notion d'enchaînement.



Niveau organisationnel .

PROCEDURE .

enchaînement de phases.

PHASE .

certaines auteurs ignorent le mot phase et semblent désigner par procédure ce que nous appelons phase.

Chaque phase est déterminée par une synchronisation d'événements et un poste de travail où s'effectue une transformation.

C'est au niveau de la procédure qu'on peut connaître les différents postes de travail, où s'effectuent les opérations concernant un même processus ( cette phrase montre que les définitions qui ont été données sont conformes à leur usage dans le langage habituel).

Une même opération effectuée à des postes de travail différents, ou avec des synchronisations différentes, conduit à des phases de noms différents.

On peut, pour se convaincre du sens des mots, réfléchir aux différences qu'on introduirait en parlant, par exemple dans le cas de l'ouverture d'un compte dans une banque, en utilisant les mots : processus, opération, procédure, phase.

POSTE DE TRAVAIL :

Il n'y a qu'un poste de travail par phase. La notion de poste de travail implique des moyens et des hommes. Cependant ceux-ci ne sont pas considérés comme des acteurs car les événements qu'ils peuvent déclencher (en faisant leur travail) sont attachés au poste de travail et non à eux-mêmes.

ACTEUR :

personne qui décide et dont la décision intervient dans le pilotage d'un système.

PROGRAMME, ECRAN, ETAT :

Ils sont simplement nommés au niveau organisationnel. La description détaillée est faite au niveau opérationnel.

Cependant leur présence au niveau organisationnel est indispensable pour indiquer de façon formelle les flux de données.





## Niveau opérationnel.

Il concerne la définition détaillée de chaque programme.

Des liens marqués par les noms de programmes, d'état ou d'écrans, existent avec le niveau organisationnel

## CONCLUSION.

La conduite de projets, la documentation, la maintenance, dans une grande ou moyenne organisation, requièrent l'intervention de plusieurs personnes, à des moments différents, parfois éloignés dans le temps, nécessitant une approche tantôt montante, tantôt descendante, pour des questions de caractère général ou pour des problèmes limités.

Une approche formelle rigoureuse est nécessaire pour donner et garder une cohérence formelle d'abord, réelle ensuite, à la modélisation du système d'informations.

Mais il faut être d'accord - préalablement - sur le sens des mots.



**Bâtir son "A. G. L." ?**

**Pourquoi pas ...**

**Oui, mais comment ?**

La commission  
"Les préalables à la mise en oeuvre  
d'un Atelier de Génie Logiciel"

a achevé la première étape de ses travaux en apportant quelques éléments de réponse.

*Documentation complète sur demande*

Ces éléments sont contenus dans 4 documents dont vous trouverez les résumés dans les pages suivantes.

Chaque texte (d'une vingtaine de pages) est disponible, pour chacun de vous, en version intégrale, sur simple demande écrite auprès du secrétariat de l'ADELI.

*Présentation-débat en janvier 1991*

La commission organisera une matinée de présentations et de discussions, suivie d'un repas, à la mi-janvier 1991.

La date exacte et le contenu de l'ordre du jour vous seront communiqués dans quelques semaines par un courrier séparé.



## DOCUMENT D'ORIENTATION

### *Introduction*

### *La problématique*

Les questions

#### Place et rôle des ateliers de fabrication de logiciels

Généralités

Eléments pour une typologie des besoins pris en charge par l'ingénierie informatique

- traitement de l'information et accès à l'information
- le champ d'application de l'ingénierie informatique

La structuration d'un système d'information

Conception, spécification et réalisation des systèmes d'information

Structuration, spécification et fabrication d'un logiciel

Les systèmes d'information sont des instruments

Organisation de l'espace d'utilisation du système et préparation de sa mise en oeuvre

#### Composants du génie informatique

Généralités

Le cadre méthodologique

Les savoir-faire théoriques : les Méthodes & Techniques

Généralités

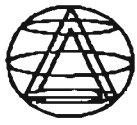
Méthodes & Techniques de développement

Méthodes & Techniques d'assurance-qualité

Méthodes & Techniques de conduite de projet

Le savoir-faire pratique

Les outils



## *Le groupe de travail ADELI - P.A.G.L.*

Les objectifs actuels du groupe de travail

Destinataires des travaux du groupe

Les membres de l'entreprise  
Les prestataires

Participation aux travaux du groupe

Les participants  
Les types de participation

Structures à mettre en place

Programmes

Programme général  
Programme détaillé  
Lancement

Orientations générales

Modalités de fonctionnement

Les réunions  
Les documents de synthèse

## *Terminologie*

Ce document d'orientation, réalisé en octobre 1988, fournit le cadre de la commission et indique son mode de fonctionnement prévisionnel.

A quelques détails près, actualisés empiriquement, ce document, constitue en permanence, la trame de notre travail.

Il est nécessaire de vous y reporter pour mieux situer les éléments que la Commission propose à votre réflexion.



## APPROCHE DU SYSTEME INFORMATIQUE PAR L'INTERFACE D'UTILISATION

Comment sont construits les systèmes informatiques actuels ?

Trop souvent, les Informaticiens partent de ce que peuvent faire leurs plateformes et proposent aux Utilisateurs les fonctionnalités qu'ils savent réaliser. Dans cette démarche, la maîtrise des technologies monopolise toute leur attention et commande la solution qu'ils imposent.

Cette approche par les contraintes techniques est-elle incontournable ?

### 1. Les travaux de l'entreprise

Les travaux de l'entreprise s'étudient de façon hiérarchique. On distingue arbitrairement six niveaux qui ont reçu des noms conventionnels :

- activités générales
- activités particulières
- activités élémentaires
- tâches générales
- tâches particulières
- tâches élémentaires

Chaque niveau se décompose en plusieurs occurrences de niveau inférieur. Ainsi, une activité élémentaire se compose de plusieurs tâches générales.

### 2. L'aspect externe du système informatique

En regard des travaux à effectuer, l'aspect externe du système informatique se compose lui aussi de six niveaux :

- fonctions générales
- fonctions particulières
- fonctions élémentaires
- fonctionnalités générales
- fonctionnalités particulières
- fonctionnalités élémentaires



Chaque niveau se décompose en plusieurs occurrences de niveau inférieur. Ainsi, une fonction élémentaire se compose de plusieurs fonctionnalités générales.

### 3. *L'architecture des logiciels*

Les logiciels correspondants se composent de niveaux imbriqués :

- sous-systèmes
- progiciels
- programmes
- macro-modules
- modules
- macro-instructions

### 4. *Les correspondances travaux - interfaces - logiciels*

De là à envisager des bijections entre les travaux et les logiciels via les interfaces (aspects externes du système) il n'y a qu'un grand pas.

### 5. *Les qualités des interfaces*

En outre, toute interface d'utilisation devrait offrir les propriétés suivantes :

- convivialité :
  - . menu
  - . infocentre
  - . bloc-note
  - . gestion des travaux
  - . transfert
- protection
- macro-commande
- documentation

La documentation utilisateur, partie intégrante du logiciel, doit comporter les éléments suivants :

- . guide d'utilisation
- . didacticiel
- . manuel de référence
- . aide directe
- . guide de macro-langages

De plus, la documentation de fabrication est indispensable à chaque modification en atelier du logiciel.



## DE L'ENTREPRISE AU SYSTEME D'INFORMATION

1. Un schéma, qui ne pêche plus par son originalité, nous affirme que *toute entreprise se compose de trois systèmes complémentaires.*

- le système de production
- le système de pilotage
- le système d'information
- les muscles,
- le cerveau,
- le système nerveux.

Une analyse plus poussée démontre que le système de production (encore appelé système opérant) est un opérateur, au sens mathématique, qui transforme :

- des intrants (besoins du marché, matières premières, énergie, main d'oeuvre,...)
- en extrants (solutions - produits et services).

Le système de pilotage définit, en permanence, l'organisation de cet opérateur et la fait évoluer en fonction de l'environnement.

Toute décision du système de pilotage modifie l'organisation du système opérant.

Tout système de production contient un système administratif qui manipule les informations nécessaires aux transformations des intrants en extrants.

C'est le système d'information d'aide à la production.

D'autre part, le système de pilotage doit s'informer du fonctionnement du système de production et lui transmettre les ordres qui modifient son organisation.

C'est le système d'information d'aide à la décision.

2. *Avant d'automatiser tout ou partie d'un système d'information, il faut le rationaliser.* Ce faisant, on réduit le système d'information naturel réel à un système d'informatique assisté par ordinateur (S.I.A.O.).

Cette transformation s'effectue par des actions successives :

- sélection des éléments de connaissance et des comportements jugés utiles et modélisables,
- modélisation des informations et des règles de gestion,
- automatisation de certains traitements autour d'une base de données.



Toutes ces actions (sélection, modélisation, automatisation) s'appuient sur des décisions humaines, donc totalement subjectives et arbitraires.

3. *Le système informatique assisté par ordinateur s'articule en trois parties :*

- infrastructure matérielle et logicielle (plate-forme),
- logiciel applicatif,
- organisation de l'interface système informatique - système de production (postes de travail, saisies, restitutions,...).

4. *Il existe 8 différents types de systèmes informatiques selon le caractère individuel ou collectif (partageable) :*

- des données,
- des traitements,
- des ressources (moyens nécessaires aux traitements : processeurs, consommables).

Les cas-limites sont :

- le système informatique de production (informatique de labeur !) dont les données, traitements et ressources sont collectives.
- la micro-informatique individuelle (données, traitements, ressources individuelles)

5. *Le passage de l'état actuel à l'état futur impose une démarche de rénovation par étapes.*

(chaque étape constate formellement une progression harmonieuse des trois cycles (vie, abstraction, décision) :

- le schéma directeur du système d'information de l'entreprise,
- l'étude préalable d'un domaine de l'entreprise,
- l'étude détaillée d'une application d'un domaine,
- la réalisation technique d'un projet applicatif d'une application,
- la mise en oeuvre (par projet applicatif, application, domaine),
- l'évolution-adaptation d'un système.

6. *Un lexique commun définit les termes utilisés.*



# ORGANISATION DE L'ENTREPRISE ET PREALABLES AU GENIE LOGICIEL

Introduction : Pourquoi un document sur l'organisation ?

1. *La connaissance de l'organisation de l'entreprise est-elle un préalable à la mise en oeuvre de l'A.G.L.*  
(Qu'appelle-t-on organisation ?)

L'entreprise et son environnement

L'entreprise vue de l'intérieur

Entreprise et niveaux d'abstraction

Subdivision hiérarchique de l'entreprise

Définition

2. *Inconvénients à faire du génie logiciel sans étudier l'organisation*

Inconvénients d'une non-étude de l'organisation existante (passé)

Inconvénients d'une non-étude de l'organisation future

Inconvénients d'une non-étude de l'organisation transitoire présente (plan de migration)

3. *Comment est-il possible de modéliser (décrire) l'organisation d'une entreprise ?*

### 3.1 Six concepts

La notion de Tiers / Acteur de l'entreprise

La notion d'Objet d'Echange / Ressource / Prestation

La notion de Flux / Echange



La notion de Moyen / Processeur de l'entreprise

La notion de Transformation / Activité de l'entreprise

La notion de Connaissance / Compétence / Capacité de l'entreprise

### 3.2 Trois niveaux de description de l'entreprise

Conceptuel

Organisationnel ou Logique

Physique, Opérationnel ou Technique

### 3.3 Aspects statiques et dynamiques de description de l'entreprise

### 3.4 Vers une approche objets de description de l'entreprise

### 3.5 Modélisation d'ensemble de l'organisation

Conclusion : La juste place de l'organisation dans le génie logiciel

*Pourquoi, en France, exploite-t-on  
si lentement et si partiellement  
les apports de J.D. WARNIER ?*

Le jeudi 9 août, quelques-uns d'entre nous ont assisté aux obsèques de Jean-Dominique WARNIER, décédé brutalement, à l'âge de 70 ans. La presse informatique s'est montrée particulièrement discrète sur la disparition d'une personnalité qui échappait aux stéréotypes de la profession.

Cependant, qu'on le veuille ou non, adulé par quelques disciples, quelquefois trop zélés, sujet de vives polémiques, marginalisé par son comportement atypique, contesté par des professionnels agacés, WARNIER aura laissé une empreinte indélébile dans l'histoire de l'informatique.

De nombreux Adéliens n'évoqueront pas sans émotion la mémoire de J.D. WARNIER. En revanche, d'autres Adéliens jugeront déplacée toute espèce de citation, considérée a priori comme un hommage excessif.

Rappelons que l'Association ADELI a été créée en 1978 par un groupe de praticiens, réuni dans une structure permanente, pour échanger des expériences concrètes sur la mise en oeuvre des concepts exprimés par J.D. WARNIER

En complément d'autres témoignages, ces quelques lignes se limiteront à une réflexion autour de trois questions.

### *1° Qu'a apporté WARNIER à l'informatique ?*

#### *1 - L.C.P. (Logique de Construction de Programmes)*

Dès la fin des années 60, alors que les programmeurs s'ingéniaient à échafauder des châteaux de cartes qui résistaient difficilement aux secousses des modifications, WARNIER a énoncé, dans le scepticisme général, les bases de la programmation structurée.

Un programme se compose de structures alternatives et répétitives.  
La structure d'un programme se déduit de la structure des données utilisées.  
La structure des données utilisées se déduit de l'étude hiérarchique des données de sortie et des données internes.

Le temps des polémiques a suivi celui du scepticisme et des railleries. Il a fallu attendre une demi-douzaine d'années avant que les principes qu'il avait énoncés soient officiellement admis dans la culture informatique, sous les traits de la programmation structurée, en provenance, cette fois, du monde anglo-saxon !



## 2 - L.C.S. (Logique de Conception de Systèmes)

Dès le début des années 70, WARNIER a exprimé deux nouveaux concepts :

- flux d'échanges dans l'entreprise

Une entreprise échange des objets (produits et/ou services) avec des Tiers, externes et internes, Fournisseurs et Clients.

- approche d'un système par les bases logiques de données

Les traitements du système informatique de l'entreprise s'articulent autour de l'ensemble, plus stable, de ses données.

Chaque ensemble (au sens mathématique) de données doit être accompagné de sa définition en compréhension (expression explicite de la caractéristique commune des éléments de l'ensemble).

Parler de fichiers logiques de base alors que la technologie ne nous offrait encore que des gestions de fichiers séquentiels, (indexés pour les plus perfectionnés), était une gageure.

La notion de référentiel "entreprise" (qui désigne un ensemble de moyens sous une responsabilité opérationnelle) permettait une décentralisation, encore bien utopique dans le contexte organisationnel et technologique de l'époque, où l'informatique d'une firme ne se concevait pas en dehors d'un ordinateur central.

Ramener l'étude du système informatique d'une entreprise à l'analyse des ensembles de personnes physiques et morales qui échangent des produits ou des services avec elle, semblait une démarche trop généraliste pour englober toutes les activités économiques (commerce, industrie, banque, assurance, administration, ... ) et trop éloignée de ce l'on considérait comme la véritable nature des problèmes informatiques (le fameux "troisième bit à gauche").

Il a fallu attendre l'apparition et la diffusion :

- des bases de données relationnelles,
- des systèmes départementaux,

pour que les notions :

- de modèles logiques et conceptuels,
- d'unités géographiques et fonctionnelles (tiers),
- de flux entre acteurs,

soient officiellement admises comme composants incontournables de la conception de tout système informatique.

### 3 - L.D.R. (Langage de Description des Résultats)

Au milieu des années 70, WARNIER émet une nouvelle idée révolutionnaire.

Il appartiendrait désormais aux Utilisateurs de définir, eux-mêmes, les spécifications fonctionnelles de leur outil informatique. Ces fonctionnalités se confondent avec l'expression formelle des résultats de sortie.

Une sortie (état imprimé ou écran) se compose :

- d'une structure de sous-ensembles de rubriques;
- à chaque sous-ensemble est associée une condition de sortie
- chaque rubrique provient d'une origine (transfert ou calcul au moyen d'un algorithme)

Sans précipitation, petit à petit, ici et là, on a vu des Comités d'Utilisateurs donner un avis, de plus en plus pertinent, sur les caractéristiques fonctionnelles de leur outil informatique.

### 4 - L.C.E. (Logique de Conduite de l'Exploitation - S.D.D.)

A la fin des années 70, le Système de Données sur les Données préfigure l'amorce du Dictionnaire de Données.

Il s'accompagne d'une tentative d'automatisation de l'exploitation par programmation logique du langage de commande (J.C.L.). Cette initiative avorte devant la réticence des expérimentateurs potentiels, limités par les possibilités restreintes du langage (absence de répétitive, entre autres).

Là encore, la technique utilisée ne permettait pas de réaliser, immédiatement, un outil performant traduisant ce concept.

Il faut attendre une dizaine d'années pour que les encyclopédies, les bases de spécifications, systèmes de gestion d'objets, .. trouvent leur place centrale au sein des A.G.L. dignes de ce nom.

## 2 Pourquoi ces apports aujourd'hui familiers et incontestés ont-ils eu une trajectoire aussi lente ?

On peut s'étonner de l'absence d'impact instantané de ces idées simples. Pourtant, ces principes ont révélé leur pleine efficacité dès que l'évolution des technologies et des mentalités ont autorisé leur utilisation.

Peut-on en tirer quelques leçons pour l'avenir, en recherchant et identifiant les obstacles qui ont retardé ces progrès ?



## *La défiance des Français vis-à-vis des compétences de leurs compatriotes*

Nul n'est prophète en son pays et les travaux français sont souvent plus connus à l'étranger qu'en France.

Lors du Comité de programme des Journées de Génie Logiciel de Toulouse qui réunissait le gratin des grosses têtes pensantes sur le sujet, de nombreux participants n'avaient jamais entendu parler ni de la Logique Informatique, ni de J.D. WARNIER, alors qu'aucun d'eux ne se serait autorisé à ignorer la dernière publication issue d'un quelconque Di-aïe-i-ti (Dallas Institute of Technology).

Pourquoi faudrait-il toujours attendre la caution anglo-saxonne pour accepter un progrès lié à un domaine technologique ?

## *La crainte de la simplicité*

Les idées simples sont les plus difficiles à diffuser.

Toute simplicité fait peur. Surtout quand elle entraîne l'inconfort d'un changement d'habitude.

Songons à l'angoisse de nos ancêtres qui ont subi la transition des chiffres romains aux chiffres arabes. Il leur a fallu passer de la multiplication de XXIX par LIV que l'on était si fier de maîtriser à l'énigmatique 29 x 54.

Plus près de nous, depuis trente ans, les nouveaux francs permettent pour la même dépense de manipuler des nombres plus condensés. Ils sont ne sont encore reconnus par la clientèle des animateurs des jeux télévisés.

"Pourquoi faire simple quand on peut faire compliqué ?" affirmaient les petits personnages d'une série télévisée. Beaucoup de téléspectateurs ont pris cette plaisanterie au premier degré !

## *Les effets pervers de la pédagogie*

Les ouvrages de J.D. WARNIER, d'une grande rigueur, ont emprunté à la pédagogie scolaire quelques-unes de ses techniques (conclusions en caractères gras, graphiques clairs, résumés, )

Toutes techniques qui rajeunissent le lecteur.

Celui-ci, offense d'être pris pour un enfant, regrette

- les développements confus,
- dans un charabia diffus,
- illustrés de diagrammes touffus

qui par leur complexité valorisent à la fois l'enseignant et l'enseigné.

Que dire aussi de l'angoisse de ne pouvoir comprendre des choses aussi simples que la fameuse histoire (\*) des consommateurs bénéficiaires d'une remise, partiellement détournée par un garçon de café indélicat.

\* Trois consommateurs règlent, 15 F. chacun, leur boisson favorite. Sur ces 45 F., le patron leur ristourne 5 F.; somme sur laquelle le garçon prélève 2 F. en ne rendant qu'une pièce d'1 F. à chacun. (suite et fin à la page suivante)



# ADELI

Le choix d'exemples, volontairement simplifiés pour ne pas alourdir la durée de la formation, induit que la méthode exposée ne permet de résoudre que des cas élémentaires.

On a entendu et répété :

L.C.P. ne marche que pour des fichiers séquentiels,  
L.C.P. ne marche qu'en COBOL,  
L.C.P. n'est pas adapté au temps réel,

...

parce qu'en 1970, les premiers exemples traitaient en COBOL, en temps différé, des fichiers séquentiels.

On vit la cruelle illustration de la sagesse chinoise :  
"Quand l'on montre du doigt, le sot regarde le doigt"

### 3 . Quelles sont les idées de WARNIER qui n'ont pas encore été exploitées qui ne sont pas encore entrées dans la culture informatique ?

Si les concepts mathématiques ou méthodologiques exprimés par J.D. WARNIER sont aujourd'hui fréquemment exploités (dans l'oubli du nom de leur auteur) par les techniques informatiques; en revanche les préceptes humanistes restent, encore, lettre morte, bien à l'abri dans les pages de ses ouvrages.

#### 1. La finalité de l'informatique

Pourquoi, pour qui, informatise-t-on ?  
L'informatique est-elle faite pour satisfaire les besoins des Utilisateurs ou les ambitions des Informaticiens ?  
Que faut-il informatiser et quand ?  
Quels sont les critères de progrès ?  
La rentabilité consiste-t-elle uniquement à supprimer des emplois ?

#### 2. La place respective de l'homme et de la machine

La machine doit être au service de l'homme et non l'inverse.  
N'essayons pas de transférer aux machines les travaux qui nécessitent des qualités humaines (la décision) en laissant aux hommes les travaux répétitifs (la saisie des données et le classement des archives produites par la machine).  
Ce n'est pas à la machine d'imposer ses contraintes mais aux concepteurs de machines d'intégrer dans leur cahier des charges, les exigences des utilisateurs.  
L'intelligence n'est pas artificielle; "Pour être (quelquefois) intelligent, il faut être (souvent) bête !" et seul l'homme a le privilège de l'erreur et de la correction, donc du progrès.

Chacun a payé 14 F. : 9 fois 14 = 126; 126 + 2 = 128; Au voleur, il manque 1 F. !!!



### 3. La formation à la logique

La formation de base à l'utilisation des systèmes informatiques n'est pas la formation technique mais la formation au raisonnement logique.

N'avivons pas les échecs douloureusement répétés des plans informatiques scolaires (Informatique Pour Tous - ou pour les adroits Commerçants retenus).

On a mesuré l'effort en budgets d'achat de matériels sophistiqués rapidement périmés; mais était-il besoin d'un matériel coûteux pour mettre en évidence l'utilisation des mathématiques (qui furent dites modernes) pour apprendre à poser un problème ?

### 4. La répartition des bénéfices des progrès

Un progrès technique qui enrichit une minorité en asservissant une majorité est-ce un progrès social ?

Inonder les pays en voie de développement d'ordinateurs inutilisés, n'est-ce pas élargir le fossé qui les sépare des pays industrialisés ?

Quelques-uns d'entre vous souriront ou hausseront les épaules à la lecture de ce texte, en y trouvant les traces d'une manipulation, destinée à déstabiliser nos confortables valeurs économiques et sociales.

Ne nous méprenons pas. Il ne s'agit pas de préoccupations altruistes, "dégoulinantes de bons sentiments".

Un raisonnement individualiste vous montrerait qu'il s'agit tout simplement, au lieu de se complaire dans la bonne conscience béate des technocrates repus, de faire évoluer, avant qu'il soit trop tard, les conditions de notre survie, de votre survie.

Combien faudra-t-il laisser encore de temps au temps pour que ces idées deviennent enfin des évidences ?

Par quel chemin parviendront-elles enfin jusqu'à nous ?

Alain COULON - octobre 1990