Université de NANCY 1 — UER Sciences Mathématiques

Centre de recherche en Informatique de NANCY (CRIN)

" Analyse comparative de quelques méthodes de conception de systeme d'information. Définition d'un cadre de comparaison "

DIPLOME DE RECHERCHES DOCTORALES

Présenté et soutenu publiquement le 20 JANVIER 1989

à l'Université de NANCY 1

par

NDONG BIRAMA

Président	:	J.P.	FINANCE
Examinateur	:	J.C.	DERNIAME
Rapporteur	:	O.	THIERY

Université de NANCY 1

UER
Sciences Mathématiques

Centre de recherche en Informatique de NANCY (CRIN)

" Analyse comparative de quelques méthodes de conception de systeme d'information. Définition d'un cadre de comparaison "

DIPLOME DE RECHERCHES DOCTORALES

Présenté et soutenu publiquement le 20 JANVIER 1989

à l'Université de NANCY 1

par

NDONG BIRAMA

Président	:	J.P.	FINANCE
Examinateur	:	J.C.	DERNIAME
Rapporteur	:	O.	THIERY

Remerciements.

Je tiens à remercier du fond de mon coeur et de façon particulière Madame ODILE THIERY Professeur à l'Université de NANCY II, qui pendant un peu plus de trois ans, a accepté non seulement de suivre et de corriger mes travaux , mais aussi d'effectuer toutes les démarches administratives (inscriptions et autres) pour moi qui me trouvais à des milliers de kilomètres de NANCY,

Je tiens à remercier :

- J.P. FINANCE, Président de ce jury, Directeur du CRIN, Professeur à l'Université de NANCY I, grâce à qui j'ai pu m'inscrire à ce diplôme de troisième cycle, et qui n'a jamais ménagé ses efforts pour le développement et l'évolution de l'IAI (Institut Africain d'Informatique),

- J.C. DERNIAME, Professeur à l'Université de NANCY I, qui m'a accueilli et mis à l'aise au sein du CRIN, et dont les efforts inlassables pour le développement de l'Informatique en AFRIQUE ne se sont jamais démentis,

- l'IAI qui m'a offert le cadre de travail nécessaire pour passer dans de bonnes conditions ce diplôme de recherches doctorales.

Je remercie également :

- Monsieur TANKOANO JOACHIM, Professeur à l'IAI, dont les conseils amicaux et désintéressés m'ont beaucoup servi,

- tous les étudiants de l'IAI qui, d'une manière ou d'une autre, m'ont aidé pour les travaux de saisie,

- ma femme qui a repris en clair mes gribouillages et n'a eu de cesse de m'encourager.

II

à mes très chers parents

El hadj NDONG MOUNDOR

Hadja DIOKH MAïMOUNA

III

S O M M A I R E.

IV

V

INTRODUCTION.

1/ OBJECTIF.

Le travail présenté dans ce mémoire se situe dans le courant de recherches concernant la conception des systèmes d'information (SI). Celle-ci a connu d'importants développements depuis les années 1970. Des centaines de méthodes de conception de SI ont vu le jour. Cette étude portera sur la définition d'un cadre qui permette d'effectuer une comparaison entre diverses méthodes reconnues comme étant les plus représentatives de la conception des SI.

2/ DEFINITION D'UN SI.

Qu'entend - on par SI ? Ce concept joue un rôle clé en informatique de gestion d'où la nécessité d'en montrer les divers aspects. A ce sujet plusieurs définitions liées à différentes approches particulières ont été proposées :

2-1 APPROCHE TRADITIONNELLE PAR L'ANALYSE DES SYSTEMES.

La démarche systémique [MOI 77] est une méthode d'investigation au sein des organisations : toute organisation est considérée comme un système global au sein duquel agissent en interrelation dynamique plusieurs sous-systèmes (production, finances, administration....), ceci dans le but de répondre à des objectifs prédéfinis. Les différents sous-systèmes une fois mis en evidence, considérés individuellement de façon intrinsèque, constituent eux-mêmes des systèmes mettant en oeuvre d'autres sous-systèmes. Il devient alors possible par ce procédé de pousser la décomposition aussi loin que les besoins de l'analyse le demanderont. Cette approche permet de mieux comprendre le fonctionnement d'une organisation par l'étude des sous-systèmes qui la composent et de leurs interrelations.

En règle générale on distingue trois sous-systèmes dans une organisation :

(i) le système de pilotage dont les fonctions sont la définition des politiques de l'organisation et de la vérification de leur réalisation,

(ii) le système opérant avec comme fonctions la mise en oeuvre des politiques définies par le système de pilotage et la production au sens large,

(iii) le système d'information qui s'intercale entre les deux systèmes précédents et leur sert ainsi de courroie de transmission.

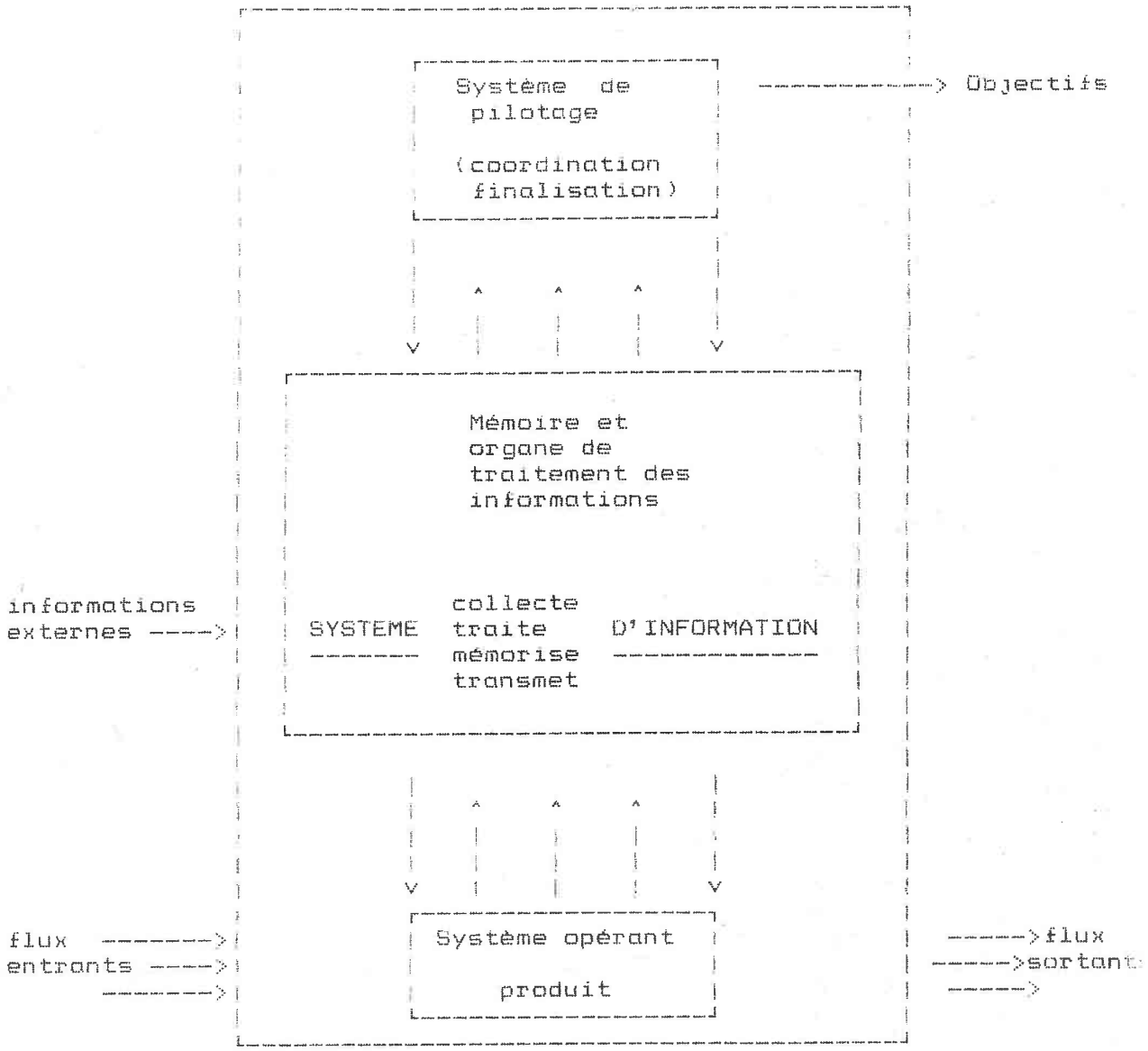
Par définition tout SI assume quatre fonctions principales relatives à l'information :

(i) la fonction de collecter l'information,

(ii) la fonction de stockage de cette information,

(iii) la fonction de transformation de cette information,

(iv) et la fonction qui consiste à communiquer cette information à toutes les autres entités de l'organisation.



Le système d'information dans l'organisation

NB : pour le schéma ci-dessus et les schémas suivants de ce chapitre :

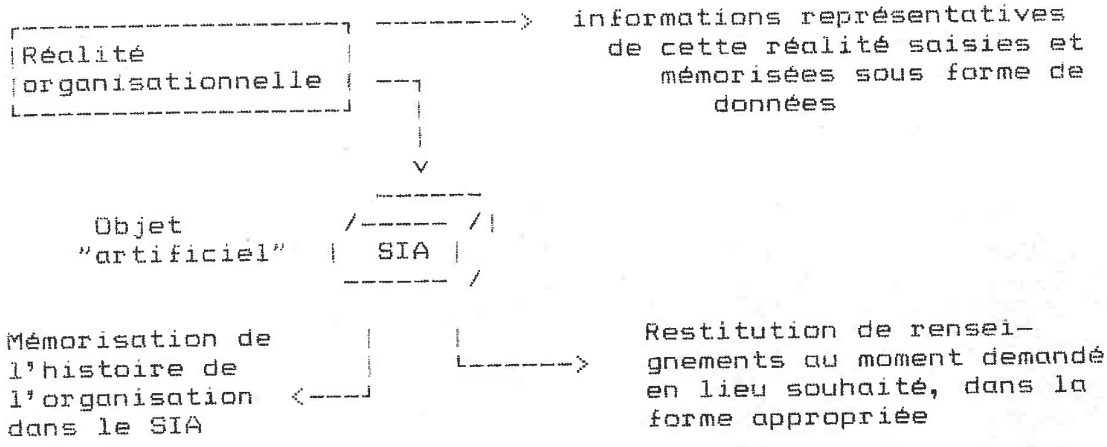
----> indique un flux d'information,

┌─────┐
└─────┘ indique un ensemble donné ou un sous-système,

<-----> indique une circulation de l'information dans les deux sens.

2-2 LE SI VU COMME OBJET ARTIFICIEL.

Une approche actuelle qui enrichit la démarche systémique permet de définir le SI comme objet artificiel construit pour représenter le SI réel et agir sur lui. Le SI réel ou réalité organisationnelle [THI 85] est appelé entité naturelle par le fait qu'il a une existence physique déterminée dans son environnement qu'est l'organisation. L'exemple le plus caractéristique de la représentation d'un SI sous la forme d'un objet artificiel est le SI automatisé (SIA).



Le SIA grâce à des moyens tels que la saisie en temps réel, collecte des informations représentatives de l'activité de l'organisation, mémorise des informations primaires et des informations élaborées dans des supports tels que les disques magnétiques, les traite grâce à l'ordinateur et les communique sous forme d'états ou d'images écran aux diverses entités de l'organisation concernées par le travail informatique accompli dans des conditions de lieu et de temps déterminés. Pour effectuer les fonctions d'automatisation, le SIA utilise plusieurs types de représentations [FOU 82] :

(i) les données qui peuvent contenir des rubriques codées sont regroupées en fichiers qui ne sont en fait que des représentations partielles d'ensembles ou de sous-ensembles de l'organisation,

(ii) les instructions d'entrée-sortie, de calcul, de transfert... qui représentent les différents traitements à appliquer sur les données pour obtenir les résultats intermédiaires ou définitifs,

(iii) les "commandes dynamiques" ou conditions de déclenchement des divers traitements (définition des périodicités, et différentes phases d'exploitation...) qui représentent les modalités de fonctionnement du SIA sur une configuration donnée.

3/ LES SOLUTIONS DE CONCEPTION DE SI.

Une fois le SI défini comme objet artificiel, les questions que l'on se pose sont les suivantes :

(i) comment passer du SI objet naturel au SI objet artificiel ? ,

(ii) comment créer un objet artificiel représentatif sur le plan abstrait du SI entité naturelle ? ,

Telles sont les questions centrales de la conception des SI. Plusieurs approches ont été proposées, nous n'en examinerons que quelques unes :

3-1 APPROCHE ORIENTEE "ANALYSE DES BESOINS" .

Elle consiste comme son nom l'indique à débuter la conception du SI par la prise en compte des besoins formulés par les différents utilisateurs [THI 85]. Par la suite le problème sera de déterminer le processus par lequel sont obtenus les résultats (ou divers besoins des utilisateurs définis sous forme d'états de sortie). On utilise pour cela des outils tels que les grilles d'information [REIX 74]. L'étude détaillée du processus d'obtention des résultats permet d'obtenir les données d'entrée primaires sur lesquelles seront appliqués les traitements adéquats pour obtenir les résultats escomptés. Ces données primaires seront regroupées en fichiers qui ne seront en fait que des structures de données établies en fonction de l'usage qu'on en fait c'est à dire du traitement propement dit. Il y'a donc une dépendance très nette entre données et programmes.

C'est là une des grandes lacunes de cette méthodologie de "conception" de SI ; on peut citer d'autres critiques :

(i) il y'a un nombre important de fichiers plus ou moins spécialisés par application et une duplication de rubriques dans plusieurs fichiers,

(ii) les besoins des utilisateurs sont divers et évoluent, ils ne sont pas limités par exemple à l'automatisation de simples processus administratifs, ou à l'édition d'états de gestion, ils concernent des systèmes ouverts où les besoins formulés ne sont pas aussi explicites (exemples : réapprovisionnement automatique du stock entrepôt, automatisation intégrée d'une législation ...),

(iii) en outre la multiplication des fichiers rend difficiles les interrogations ponctuelles à la demande des utilisateurs.

Ces inconvénients illustrent bien le caractère partiel des représentations du SI entité naturelle obtenues par cette démarche :

(i) les fichiers figurent l'état du SI à un instant donné (état des clients d'une organisation à une certaine date par exemple),

(ii) la connaissance du passé (non prise en compte du temps) n'est pas non plus exhaustive : les historiques conservés dans les fichiers recouvrent des images historiques partielles (exemple de l'historique des mises à jour intervenues dans tel

fichier à partir de telle date) en général limitées aux données sans prise en compte des mutations des programmes et des commandes dynamiques au cours du temps.

3-2 APPROCHE "BASES DE DONNEES".

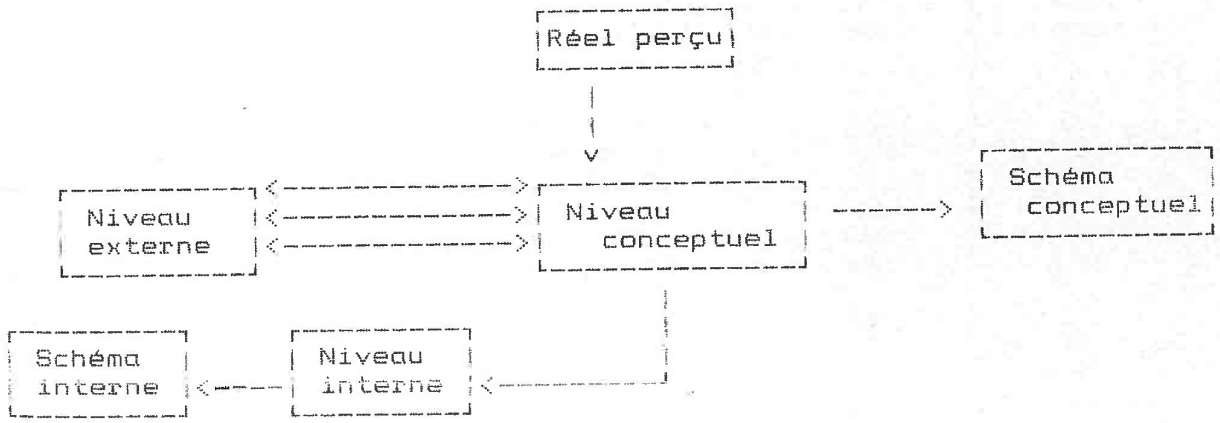
Les bases de données depuis le rapport d'étude ANSI/SPARC de 1975 [ANS 75] ont permis en développant de nouveaux concepts, de pallier une partie des inconvénients des SI traditionnels en particulier l'aspect dépendance données-programmes. Plusieurs niveaux ont été identifiés pour améliorer la représentation des données :

- le réel perçu est la représentation que le système de pilo - tage se construit pour son usage. Il s'agit d'une activité définie par le système de pilotage et que l'on peut décrire à partir des fonctions du système opérant,

- un niveau conceptuel ou niveau d'abstraction le plus élevé qui débouche sur un schéma conceptuel des données, représentation "quasi stable" du SI de l'organisation,

- un niveau interne basé sur le schéma conceptuel des données où sont prises en compte les contraintes d'exploitation et d'implantation,

- un niveau externe qui permet de décrire des sous-schémas ou schémas externes qui réflètent les préoccupations des utilisateurs. Ces sous-schémas ne sont qu'une partie de la structure des données dédiée à des applications particulières.



Les bases des données ont contribué de façon certaine à l'amélioration de la conception des SI. Elles privilégient cependant l'aspect données au détriment des traitements et des commandes dynamiques [FOU 82].

3-3 PRISE EN COMPTE DE LA DYNAMIQUE.

Que l'on procède par l'approche "analyse par les besoins" ou par l'approche "base de données" c'est l'aspect données qui est toujours privilégié sans prise en compte de la dynamique interne du SI et des relations avec son environnement [FOU 82].

En effet les organisations bien que régies par des règles de fonctionnement précises subissent des mutations et évoluent (changements d'état de l'organisation, les transitions de cette organisation d'un état à un autre ...). Les nouvelles approches de conception des SI, tout en considérant comme acquis le principe de représentation abstraite, se préoccupent d'intégrer dans leur conception toute cette dynamique des organisations. On peut citer dans ce cas REMORA [ROL 82]. Seront par conséquent pris en compte les changements d'état de l'organisation ainsi que ses transitions d'un état donné à un autre. La dynamique peut aussi refléter les réactions de cette organisation face à des événements synchronisés ou isolés dont la présence permet le déclenchement de certaines opérations ou traitements.

4/ VERS UNE ETUDE COMPARATIVE.

Ce survol des différentes approches montre l'étendue et la complexité de la conception des SI. Telle méthode peut avoir une orientation particulière mettant l'accent soit sur les données, soit sur les traitements ou proposer une intégration complète de la dynamique, soit effectuer une combinaison entre les aspects données-traitements-dynamique. Cette diversité rehausse l'intérêt d'une étude comparative entre diverses méthodes afin d'en tirer des conclusions qui contribuent à améliorer la conception des SI.

Des rencontres internationales ont été consacrées à ce sujet comme celles que l'IFIP a organisées à partir de 1979. Au départ deux thèmes : "conception et évaluation des SI", "développement des approches d'analyse, de spécification et d'évaluation assistées par ordinateur" furent proposés par le groupe d'étude. Les principes théoriques fondamentaux des méthodes devaient être exposés avec clarté, mais en plus ces méthodes devaient résoudre un énoncé en l'occurrence l'organisation d'une conférence internationale de style IFIP.

La première conférence d'études comparatives [CRIS-1] eut lieu en MAI 82 en vue d'expliciter, parmi toutes les propositions reçues (vingt-cinq), les méthodes les plus représentatives de la conception des SI. Sept méthodes (ACM/PCM, ISAC, D2S2, REMORA, NIAM, USE et DRAFT PROPOSAL) furent ainsi sélectionnées et détaillées, sept autres furent également présentées (CIAM [GUS 82], EDM [RZE 82], IML [RIC 82], ISSM [SOL 82], SDLA [KNU 82], SYSDOC [ASH 82], D2S2 [MAC 82]).

Une deuxième conférence [CRIS-2] qui était le prolongement de CRIS-1 eut pour objet l'étude comparative approfondie de ces sept méthodes.

Une troisième conférence [CRIS-3] s'est déroulée à Paris en OCTOBRE 84 avec pour point principal une présentation des méthodes d'analyse MERISE [LE 84], IDA [BOD 84] et AXIAL [AX 84].

Depuis lors d'autres conférences CRIS ont eu lieu mais elles ont surtout constitué des compléments par rapport aux congrès précédents.

5/ OBJECTIFS ET PLAN DU MEMOIRE.

5-1 OBJECTIFS.

Notre but est d'effectuer une étude comparative de certaines des méthodes traitées par CRIS-1, CRIS-2 et CRIS-3 en étudiant ces méthodes sur la base des critères évoqués précédemment (les différents niveaux d'abstraction, les types d'approche) ainsi que sur la base d'autres critères tels que ceux relatifs aux techniques de construction des méthodes ou aux différents cycles de vie du logiciel par exemple.

5-2 PLAN DU MEMOIRE

Le plan de l'étude se présente comme suit :

1ère partie : étude des divers concepts liés aux trois niveaux d'abstraction (niveau conceptuel, niveau physique, niveau logique), des paramètres associés aux différents cycles de vie du logiciel...,
définition d'un cadre de comparaison.

2ème partie : situation de chacune des méthodes étudiées par rapport au contenu de la première partie du plan et comparaison des différentes méthodes retenues, élaboration d'un tableau de synthèse,

3ème partie : à partir des constatations effectuées dans la deuxième partie dégager un certain nombre d'idées générales, essayer en conclusion de déterminer quelles doivent être les caractéristiques d'une bonne méthode de conception de SI et jeter ainsi les bases d'élaboration d'une nouvelle méthode de conception de SI.

PREMIERE PARTIE

DEFINITION DU CADRE DE COMPARAISON ET PRINCIPAUX THEMES A ETUDIER.

I - 1 INTRODUCTION .

Plus d'une centaine de méthodes se proposent d'apporter des solutions originales aux problèmes posés par la conception des SI. Leurs approches sont souvent divergentes, le vocabulaire utilisé ainsi que les symboles de représentation ne sont pas toujours identiques. Cette diversité rend difficile à priori tout exercice de comparaison. Cependant un certain nombre d'éléments de référence peuvent être dégagés.

Définir dans ces conditions un cadre de comparaison n'est pas chose aisée : pour ce faire nous nous reporterons auparavant aux travaux de BODART ET AL [BOD 83] dans CRIS-2 [IFI 83] qui serviront de base a notre réflexion.

I - 2 TRAVAUX DE BODART ET AL SUR LES DIFFERENTS CYCLES.

Ces auteurs partent du constat qu'un SI, comme tout être vivant, a un cycle de vie. Il est d'abord conçu avant sa naissance proprement dite, vit un certain temps et meurt. La phase de conception correspond à la phase de gestation du SI (elle débute à partir de l'étude d'opportunité jusqu'au stade où le produit est prêt à être réalisé), la phase d'implémentation annonce la naissance du SI qui entame alors une vie active marquée par des périodes de naissance e d'évolution. Celles-ci s'achèvent par une mort due à une obsolescence des techniques utilisées par rapport aux nouvelles techniques disponibles ou à une inadaptation de ce SI à l'évolution des besoins.

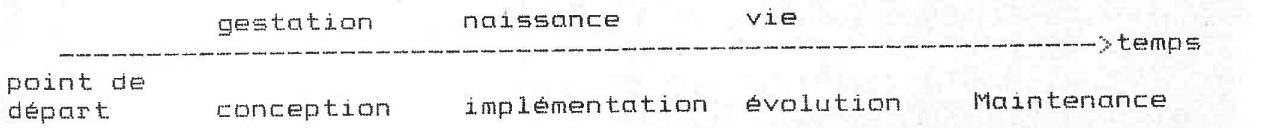


Figure 1 : Processus de développement d'un SI.

Le processus de développement du SI recouvre les différentes étapes de son cycle de vie y compris la maintenance ; il se déroule selon une séquence de cycles qui correspondent aux versions successives du SI à travers le temps.

I-2-1 LES CLASSES DE PROBLEMES ET LES CYCLES CORRESPONDANTS.

Pour définir un cadre d'analyse du développement d'un SI BODART et AL formulent l'hypothèse de l'existence de trois classes de problèmes intrinsèquement indépendantes les unes des autres, problèmes que l'on rencontre tout au long du processus de développe ment d'un SI. Ces classes de problèmes sont de trois types :

- les problèmes d'abstraction relèvent de la nature même du SI considéré comme artificiel représentant une réalité organisationnelle donnée,

- les problèmes de décision naissent du fait de l'interférence entre l'objet artificiel et l'objet naturel du SI ; ils doivent vérifier la consistance de l'objet artificiel par rapport à l'objet naturel,

- les problèmes de contrôle émergent lorsqu'il s'agira de vérifier tout au long du cycle de vie du SI s'il y a conformité par rapport aux normes et standards requis.

En faisant un parallèle avec la vie d'un SI caractérisée par une séquence de cycles, BODART et AL émettent l'hypothèse de l'existence du cycle d'abstraction, du cycle de décision et du cycle de contrôle. Ces trois cycles en interrelation pendant la durée du cycle de vie sont gérés par un cycle appelé cycle de management. A chacun de ces quatres cycles seront associés divers paramètres.

Pendant la phase de conception du SI le cycle d'abstraction traite des problèmes relatifs aux divers types d'abstraction correspondant aux niveaux d'abstraction unanimement reconnus par la communauté scientifique. Le concepteur s'intéressera, pendant le déroulement de ce cycle, aux paramètres tels que les concepts attachés à chaque niveau d'abstraction, aux caractéristiques du schéma d'abstraction.

Le cycle de décision s'intéresse à tous les choix à effectuer et aux décisions de toute nature (décisions de modification des structures, décisions de management des ressources ...) prises durant le cycle de vie du SI. Comme paramètres associés à ce cycle on peut citer les critères d'évaluation d'une décision prise, les éléments qui conditionnent une prise de décision....

Le cycle de contrôle fait état des divers contrôles appliqués au SI et des moments de réalisation de ces contrôles.

Trois classes de problèmes (contrôle du contenu du SI à divers moments, contrôle de la conduite du SI, contrôle du processus du développement du SI) sont les paramètres associés à ce cycle.

Le cycle de management gère les trois cycles ci-dessus en définissant les différentes étapes et les outils associés à chacun d'eux. Comme paramètres dont le concepteur devra tenir compte, on aura le niveau d'interaction des trois cycles, les divers outils de planification et de contrôle des tâches, les modalités de participation des utilisateurs....

I - 3 DEFINITION DU CADRE DE COMPARAISON RETENU.

Nous appuyant sur le cadre global ci-dessus défini par BODART et AL, nous allons étudier un certain nombre de thèmes qui constitueront nos éléments de comparaison des différentes méthodes de conception de SI.

Notons au préalable que nous n'aborderons pas le cycle de décision qui traite de problèmes de management (budgets par exemple), de problèmes technico-économiques (choix d'ordinateurs),

de définitions d'objectifs divers relatifs au SI...problèmes qui sont éloignés de nos préoccupations.

1) Au niveau du cycle d'abstraction nous ferons une étude des trois niveaux d'abstraction ou niveaux de description en centrant notre analyse sur le niveau conceptuel qui offre plus d'enseignements s'agissant d'une étude comparative des méthodes de conception de SI.

a) Nous nous intéresserons par conséquent au niveau d'abstraction atteint par la méthode c'est à dire au niveau de spécification que couvre cette méthode indépendamment de la réalisation du SI.

b) De même nous verrons si la méthode couvre ou non tout ou partie du cycle de vie du SI, quels moyens elle utilise pour répondre aux qualités du schéma conceptuel. Ces moyens se composent :

b-1) du modèle de spécification proposé par la méthode : quels sont les concepts qui sous-tendent ce modèle, sont-ils clairs, explicites ?

Traitent-ils des aspects statiques et dynamiques du SI ? Sous quelle forme font-ils la représentation historique du SI ?

Ont-ils un bon pouvoir de représentation ? Produisent-ils un schéma conceptuel de qualité ? [THI 85].

Sont-ils associés à une méthode (au sens de démarche aidant à la modélisation de l'organisation) ?,

b-2) du ou des langages de spécification proposés par la méthode:

Quel est leur niveau, leurs caractéristiques procédurales ou prédicatives ?, Forment-ils un tout intégré ?,

Utilisent-ils une représentation graphique ?, Conduisent-ils à des spécifications de qualité ?, Sont-ils orientés utilisateurs ?. [THI 85].

2) Au niveau du cycle de contrôle nous étudierons les outils de spécification proposés par la méthode. Comprennent-ils des outils de contrôle et d'aide à la conception ? Quels sont la nature et le niveau de ces outils ? Qu'offrent-ils comme possibilité pour valider la spécification formelle obtenue ? Quels sont les logiciels proposés ? [THI 85] .

Ces points de comparaison seront précédés d'une présentation succincte de la méthode avec indication de son ou de ses auteurs, du lieu de développement de cette méthode et de son environnement.

Un tableau dont le modèle est présenté ci-après permettra d'effectuer une synthèse des différents éléments de comparaison.

critères ---->	origine		orientation		concepts			langage spécification			outils			étapes		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
METHODES v																

LEGENDES :

1 : origine universitaire ; 2 : origine extra-universitaire;

3 : orientation conception; 4 : couverture autre que conception;

5 : concepts explicites; 6 : concepts relatifs à la statique;

7 : concepts relatifs à la dynamique;

8 : langages de spécification graphiques;

9 : nom et niveau des langages de spécification;

10 : nature des langages de spécification;

11 : outils de contrôle; 12 : outils de documentation;

13 : autres types d'outils;

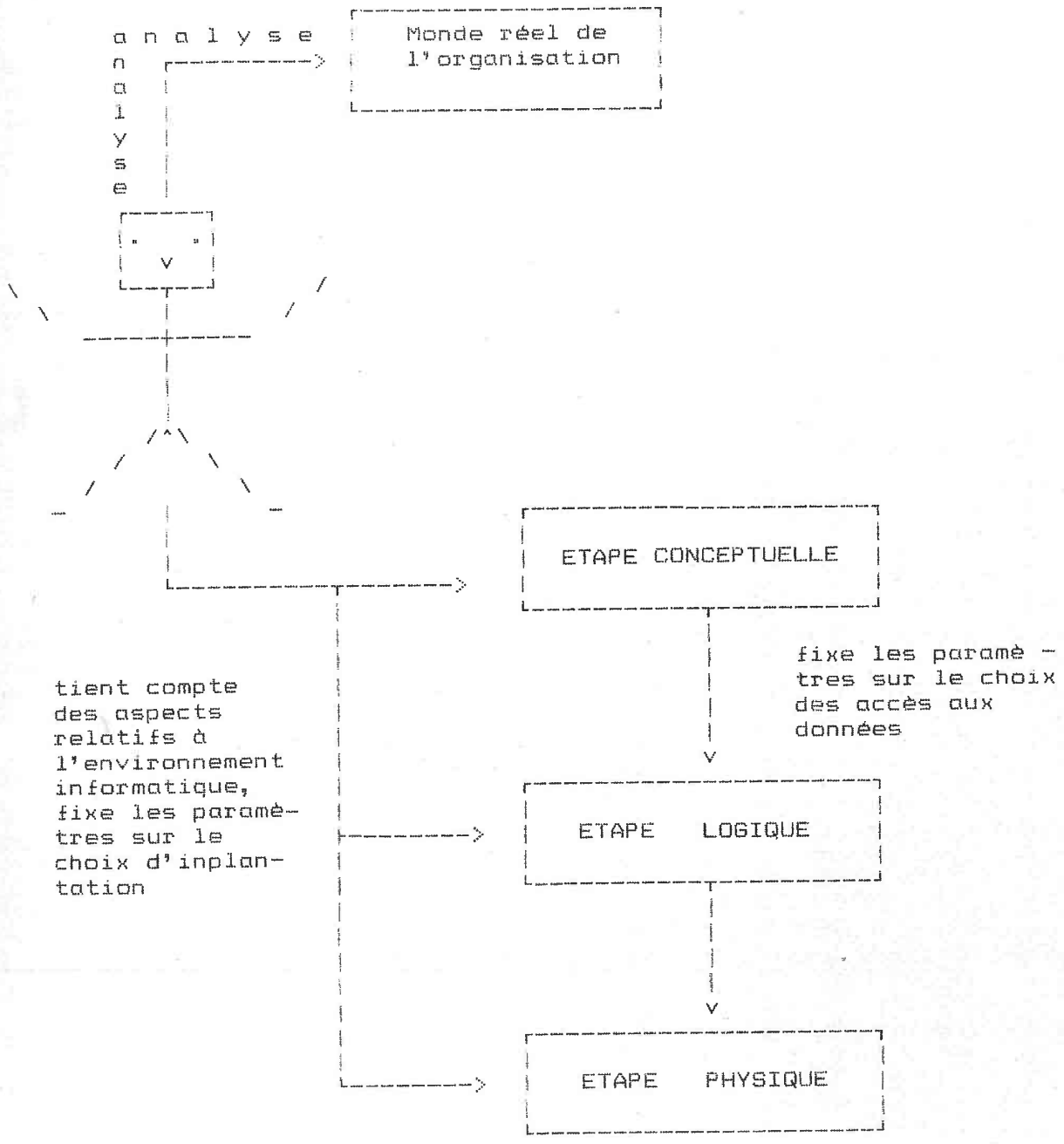
14 : étape conceptuelle; 15 : autre étape (logique en particulier);

16 : cycle complet.

Nous allons détailler avant toute analyse les différents points de notre cadre de comparaison :

I-3-1 NIVEAUX D'ABSTRACTION OU NIVEAUX DE REPRESENTATION.

Depuis le rapport ANSI/SPARC [ANS 75] les chercheurs en conception de base de données et de SI ont fait l'unanimité sur une démarche en trois étapes dénommées conceptuelle, logique et physique, démarche représentée par le schéma suivant :


analyse
analyse
Monde réel de l'organisation
ETAPE CONCEPTUELLE
fixe les paramè - tres sur le choix des accès aux données
tient compte des aspects relatifs à l'environnement informatique, fixe les paramè- tres sur le choix d'inplan- tation
ETAPE LOGIQUE
ETAPE PHYSIQUE

Le résultat de chacune des étapes définies dans la figure précédente est un schéma qui est une structure de données particulière. Ce schéma doit être complet, détaillé et documenté de telle manière que lorsque l'on est dans une étape on ne soit pas obligé de revenir en arrière soit pour mieux spécifier la solution définie antérieurement, soit pour la remettre en cause [ROL 79].

Le schéma conceptuel est l'expression conceptuelle du SI ; il doit traduire la sémantique du réel perçu grâce à une structure de données. Il doit aussi refléter tous les phénomènes réels qui surviennent dans l'organisation et que l'on doit représenter dans le SI.

Le schéma logique est le résultat de l'étape logique, il doit eproduire la totalité de la sémantique contenue dans le schéma conceptuel.

Le schéma physique quant à lui est l'expression définitive de la solution technique élaborée à partir des résultats de l'étape logique.

I-3-1-1 ETUDE DU NIVEAU CONCEPTUEL.

Le niveau conceptuel correspond à la représentation abstraite du réel perçu. Il met en évidence les diverses composantes du réel perçu et les relations que ces composantes entretiennent entre elles ceci dans le cadre des objectifs de connaissance que l'organisation s'assigne sur ce réel perçu.

Le niveau conceptuel assume trois rôles distincts :

1) Le niveau conceptuel s'intéresse aux données ou invariants statiques manipulés par l'organisation ainsi qu'aux aspects dynamiques du réel perçu ou traitements effectués sur les données à partir de règles de fonctionnement bien définies.

Pour être représentatif, le niveau conceptuel devra tenir compte du passé et de l'évolution de l'organisation c'est à dire intégrer le temps.

2) La conceptualisation débouche sur une expression abstraite du réel perçu qui doit être indépendante de tout aspect technique et doit restituer une image fidèle et cohérente du réel perçu.

3) Le niveau conceptuel constitue un point de rencontre entre informaticiens et gestionnaires dans la mesure où il reflète une réalité connue des gestionnaires. Un dialogue doit s'instaurer entre concepteurs et utilisateurs pour une meilleure connaissance des règles de gestion, des modalités d'application de celles-ci, de leur application effective au cours du temps, et des goulots d'étranglement. Le but final recherché étant une meilleure connaissance de l'organisation en vue de son amélioration.

Pour que ces trois rôles soient assumés de façon efficace, le schéma conceptuel doit posséder un certain nombre de qualités [THI 76] :

- il doit être minimal : il ne doit pas comporter de redondance,

- il doit être fidèle : il ne doit pas déformer la réalité,

- il doit être cohérent : il ne doit pas comporter de contradictions,

- il doit être complet : il doit prendre en compte tous les aspects pertinents du réel perçu de façon à permettre aux utilisateurs de retrouver dans ce schéma leurs diverses préoccupations,

- il doit être souple : pour intégrer les évolutions et les modifications des organisations,

- il doit être indépendant des usages : la représentation des règles de gestion au niveau de la dynamique doit être effectuée indépendamment de leur cadre d'application relatif au fonctionnement de l'organisation,

- il doit être compréhensible par les utilisateurs : Les concepts proposés par la méthode doivent être clairs ; la démarche de construction du schéma s'appuyant sur des outils de conception - réalisation adaptés au niveau de compréhension de l'utilisateur afin de favoriser le dialogue et permettre la validation des solutions proposées.

I-3-1-2 ETUDE DU NIVEAU LOGIQUE.

Nous distinguerons les données et les traitements :

- Au niveau des données, en l'absence d'un système de gestion de bases de données (SGBD) correspondant au formalisme conceptuel, on applique un certain nombre de règles au niveau conceptuel pour aboutir au niveau logique. Ces règles de passage ou règles systématiques dont l'application peut se faire de façon totalement algorithmique varient en fonction des formalismes utilisés au niveau conceptuel, elles permettent d'obtenir un nouveau formalisme adapté aux contraintes du traitement informatique sans perdre la signification exprimée au niveau conceptuel. La prise en compte des besoins peut modifier le schéma des données et partant le schéma logique, dans ce cas l'application des règles devient manuelle.

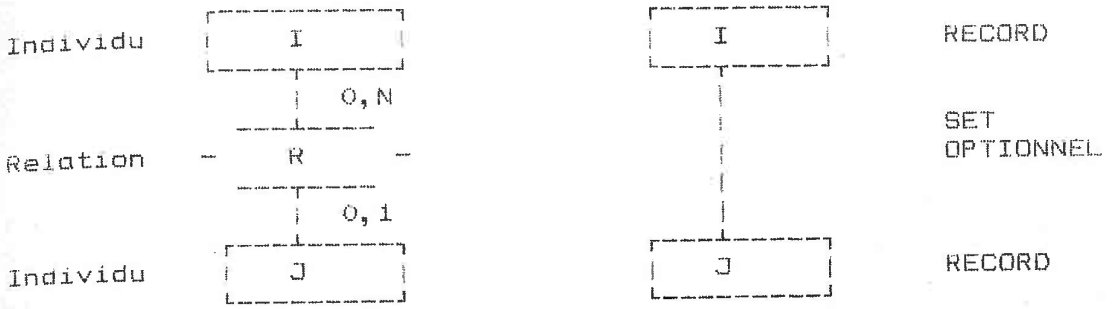
Comme exemples de formalismes nous prendrons celui du formalisme individuel [TAR 74] (individu-relation) et celui de REMORA.

- Dans le formalisme individuel les règles de transformation peuvent déboucher sur un schéma de type CODASYL; ainsi :

- "toute propriété devient un champ",

- "tout type d'individu devient un type de record, l'identifiant de l'individu devient la clé du record" ; de même les relations avec leurs cardinalités subissent des transformations :

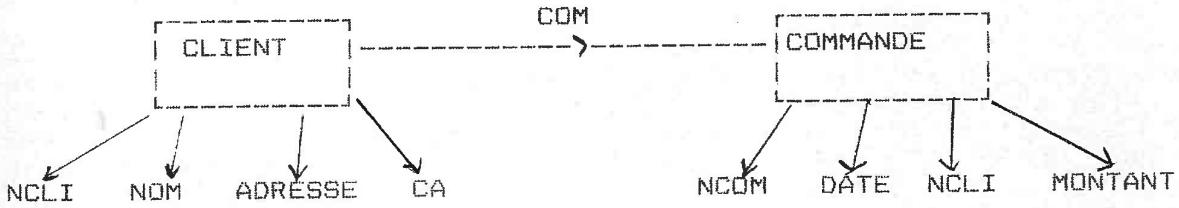
exemple : "toute relation binaire de cardinalité 0-N ; 0-1 ou 1-n ; 0-1 devient un type de set optionnel. Les propriétés de la relation R migrent alors dans le record j".



- Dans REMORA on trouve l'équivalent de ce formalisme pour faire le passage relationnel à un schéma logique standard [CABA 76] dont le modèle logique standard "sert de référence pour analyser et comprendre tous les modèles logiques des SGBD existants" [ROL 87].

Ce schéma logique standard est adaptable ensuite à tout type de SGBD dit "traditionnel".

EXEMPLE :



------ fonction d'accès
COM désignation de la fonction d'accès.

Commentaire : toute réalisation du type-article CLIENT permet, grâce aux réalisations du type-fonction d'accès COM, d'accéder aux réalisations correspondantes du type-article COMMANDE.

En résumé le niveau logique a comme objectifs au point de vue des donneés de :

- décrire les accès logiques aux données,

- définir pour les données participant à la partie automatisée du SI des structures logiques d'implantation ainsi que des hiérarchies d'accès logique,

- prendre en compte les besoins.

- Au niveau des traitements le niveau logique permet la définition d'une solution technique qui prend en compte non seulement les paramètres techniques mais également les paramètres organisationnels du traitement informatique. Par exemple dans REMORA le sous-schéma des traitements est un sous-schéma transactionnel. Pour les produire il faudra dans un premier temps interpréter le

sous-schéma conceptuel dynamique et ensuite dériver le sous-schéma logique définitif [ROL 87]. Ce dernier n'est pas automatisable complètement du fait de la prise en compte des nouveaux besoins qui peuvent le modifier.

I-3-1-3 ETUDE DU NIVEAU PHYSIQUE.

C'est un raffinement du niveau logique précédent, il résulte de décisions prises en fonction d'objectifs et de contraintes déterminées (caractéristiques d'un SGBD, systèmes de fichiers, caractéristiques des supports magnétiques...). Il se fixe comme objectif d'optimiser l'utilisation des ressources techniques en fonction des moyens disponibles. Le niveau physique sert de base aux activités de programmation, il est pris en charge en général de façon automatique.

I-3-2 LES MOYENS.

Les moyens répondent aux qualités du schéma conceptuel telles que nous les avons définies. Ils doivent permettre de satisfaire celles-ci.

I-3-2-1 LES MODELES CONCEPTUELS DE SI.

Les modèles servent à l'élaboration du schéma conceptuel. Pour spécifier le schéma conceptuel il est nécessaire de disposer de concepts et d'un formalisme élaborés ; le schéma conceptuel doit permettre de représenter de façon abstraite et indépendante des usages les données, les traitements, et les commandes dynamiques, ceci de manière intégrée et par rapport au temps [FOU 82].

Nous allons maintenant effectuer une analyse de divers modèles existant à l'heure actuelle suivant les quatres composantes de la modélisation que nous avons déjà définies (données, traitements, commandes dynamiques, intégration par rapport au temps). Pour cela nous nous servirons de la grille d'évaluation des modèles proposée par O. FOUCAUT [FOU 82] où sont croisées dans un tableau les caractéristiques de la modélisation plus la prise en compte du temps, et les composantes de la modélisation.

	Prise en compte du temps	Formalisation	Orientation Conception
Données			
Traitements			
Commandes dynamiques			
Intégration des trois Composantes			

Cette grille définit 10 critères précis d'étude du modèle du SI, le temps est mis en colonne et nous donnera des indications sur sa prise en compte au niveau des données et/ou des traitements ; de même on saura si un modèle propose une représentation formelle des données et/ou des traitements.

Nous distinguons trois groupes de modèles apparus chronologiquement :

- les modèles orientés traitement,
- les modèles de données,
- les modèles orientés dynamique.

Une étude chronologique sera menée à l'intérieur de chacun de ces groupes.

I-3-2-1-1 LES MODELES DE SI ORIENTES TRAITEMENT .

Les études portant sur ces modèles se sont déroulés antérieurement aux études portant sur les modèles de données. Ils comprennent :

- les modèles des méthodes d'analyse,
- les modèles de SI orientés traitements.

a) les modèles des méthodes d'analyse : elles utilisent les chaines de traitement. La conception d'un SI se déroule en trois grandes étapes : étude de l'existant, analyse fonctionnelle, analyse organique. Les modèles inclus implicitement dans ces modèles d'analyse ne sont pas suffisamment représentatifs d'un SI tel que nous l'avons défini :

- le critère "formalisation" n'est pas satisfait au niveau de l'étude de l'existant qui est une description informelle de tout ou partie de l'organisation avant la phase d'automatisation,
- le critère "orientation conception" n'est pas non plus satisfait au niveau de l'analyse fonctionnelle dont la définition générale est indépendante des besoins exprimés par les utilisateurs.

Ces insuffisances ont donné lieu aux :

b) modèles de SI orientés traitement: les recherches dans ce domaine privilégient la recherche d'outils permettant une certaine automatisation de l'analyse. On peut citer le projet LAPAGE [CAV 79] dont le modèle de SI est en fait un modèle de données complété par un modèle de traitement original.

Dans le contexte des "nouvelles méthodes" AXIAL [PEL 86], ISAC [LUN 82], ... présentent ce type d'approche.

Les deux modèles vus ci-dessus présentent l'inconvénient de ne pas formaliser les liens qui existent entre les données et les traitements; ils ne satisfont pas par conséquent la quatrième ligne de notre tableau.

I-3-2-1-2 LES MODELES DE SI ORIENTES DONNEES.

Nous distinguerons :

- le modèle infologique de LANGEFORS [LAN 66],

- les modèles conceptuels de données,

- les approches utilisant les types abstraits.

a) LE MODELE INFOLOGIQUE DE LANGEFORS.

Ce modèle qui a des particularités propres est antérieur à tous les autres. Basée sur une étude de l'information l'approche infologique dissocie l'aspect "infologique" du SI (celui de sa définition à partir des besoins de l'utilisateur) de l'aspect "datalogique" (celui de la représentation par des données)" [LAN 66].

Une telle approche permet une représentation élémentaire de l'information et une représentation dans le temps de cette information. Elle est basée sur un concept : le " e-message" qui est un triplet (x, y, t) représentant respectivement l'objet d'une classe (x), la propriété de l'objet (y) et l'instant (t). En utilisant une telle approche on peut représenter complètement l'historique des donnée dont la connaissance est fondamentale pour les gestionnaires.

Sur la grille d'information ce modèle satisfait tous les critères de la première ligne : Données-temps-formel-orientation-conception.

SUNDGREEN [SUN 73] a introduit le concept de e-processus pour compléter le modèle infologique. Un e-processus est défini comme "un processus qui à la fin d'une exécution couronnée de succès produit un e-message toujours du même type". L'introduction de ce concept permet au modèle de satisfaire les critères de la seconde ligne du tableau.

b) LES MODELES CONCEPTUELS DE DONNEES.

Développés depuis 1970 ces modèles sont aujourd'hui largement utilisés. Beaucoup de travaux leur ont été consacrés en particulier concernant le modèle relationnel. Ces travaux se poursuivent de nos jours.

b-1 LE MODELE RELATIONNEL.

Il a été introduit par CODD en 1970 [COD 70] et a été repris dans plusieurs études théoriques et pratiques. Le processus de modélisation s'appuie sur des schémas de relation qui traduisent la réalité organisationnelle. Ces collections ou schémas de relation sont transformées en plusieurs étapes successives appelées étapes de normalisation.

Le concept de base est la relation au sens de CODD c'est à dire que tout phénomène réel, quelle que soit sa catégorie, est représenté dans le modèle par tout ou partie d'une ou plusieurs relations [FOU 82].

Ce modèle satisfait aux critères de la première ligne du tableau à l'exception du critère "prise en compte du temps" [COD 70,71,74].

b-2 LES MODELES DERIVES DU MODELE RELATIONNEL.

On distingue le modèle de CHEN [CHEN 76], le modèle de FLORY [FLO 77], le modèle individuel de TARDIEU [TAR 74], tous ces modèles relevant des mêmes idées.

- le modèle entité - association.

Il a été introduit par CHEN [CHEN 76] ; il essaie de décrire aussi fidèlement que possible les objets manipulés dans l'organisation ceci avec des concepts simples. Pour décrire le monde réel trois types de classification sont effectués : type d'attribut, type d'entité et type d'association.

On désignera par objet ou entité tout constituant concret ou abstrait ayant une existence intrinsèque du point de vue d'un individu ou d'un groupe. Le point commun entre tous ces objets est leur appartenance pour un temps donné à l'organisation ;

On désignera par catégorie les ensembles concrets (clients, produits abstraits ou immatériels) manipulés par une organisation. Chacun de ces ensembles sera appelé catégorie : exemple : catégorie objet client ou catégorie objet salaire.

DESCRIPTION DES CATEGORIES.

Elle s'inspire des travaux effectués par les chercheurs en base de données [BEN 76], [KEN 76], [CHEN 76].

1 - définition de l'association.

C'est un ensemble de deux ou plusieurs objets appartenant à la même catégorie, ou à des catégories différentes, objets en activité dans un environnement où chacun joue un rôle spécifique.

2 - définition de la propriété.

C'est une qualité d'un phénomène ou d'une association de phénomènes réels, qualité identifiée par une référence, un nom différents de ceux des autres phénomènes ou des autres associations.

3 - définition de la classe.

C'est un ensemble de phénomènes ou d'associations de phénomènes ayant un sens identique, c'est à dire ayant en commun certaines propriétés et jouant le même rôle dans l'organisation.

4 - définition d'un identifiant.

C'est une propriété particulière d'une catégorie

donnée dont la valeur ne change pas au cours du temps.

ETAT D'UN OBJET.

état existant : l'objet existe réellement dans l'organisation lorsque celle-ci prend connaissance de l'existence de l'objet,

état inexistant : lorsque l'organisation n'a pas encore pris connaissance de l'existence de l'objet.

En résumé on peut dire que le modèle de CHEN, de TARDIEU, de FLORY apportent une amélioration par rapport au modèle relationnel en faisant une distinction entre les relations existant au niveau des objets et les relations existant au niveau des associations d'objets.

b-3 LES MODELES BINAIRES.

Ces modèles se proposent de décrire le monde réel avec le concept de relation. Le concept de table utilisé par CODD se révélant sémantiquement pauvre, les modèles binaires comblent cette lacune en privilégiant les associations binaires entre ensembles. Une relation sera définie par deux fonctions inverses l'une de l'autre. ABRIAL [ABR 74], BRACCHI [BRA 76], BUBENKO [BUB 77], SENKO [SEN 75], CREHANGE [CRE 75], FALKENBERG [FAL 75] ont proposé des modèles binaires.

b-4 LES MODELES SEMANTIQUES.

Ces modèles enrichissent les modèles binaires en permet - tant une représentation des faits de l'organisation sans biais et sans déformation. Cet enrichissement se traduit par une représentation des types et des occurrences de types sur un même schéma et la distinction de plusieurs partitionnements dans un même type [BOUZ 83]. On peut citer les travaux de SMITH [SMI 77] sur les concepts de généralisation, de classification, d'agrégation.

b-5 LES MODELES REPOSANT SUR LA LOGIQUE ET LE CALCUL DES PREDICATS.

La logique utilisée pour définir les langages d'interrogation et de manipulation de données sert ici à exprimer le schéma conceptuel qui se présente alors sous la forme d'une collection de formules d'un système formel basé sur le calcul des prédicats du premier ordre. On peut citer les travaux de NICOLAS [NIC 79], de WARREN [WAR 81]...

b-6 LES MODELES FONCTIONNELS.

Ces modèles ont pour objet de représenter et de manipuler les bases de données. Ils suggèrent que les objets et les relations qui les lient peuvent être représentés par des fonctions mathématiques définies sur des domaines de base ou des domaines construits. Dans ces modèles inspirés des travaux de BACKUS [BACK 78] et des concepts des langages tels que LISP et APL, concepts et langages sont imbriqués contrairement aux modèles

précédents où ils sont distincts. On peut citer comme travaux sur ces modèles les travaux de BUNEMAN et SHIPMAN [SHIP 81].

Ces modèles (de b-2 à b-6) s'ils présentent un certain formalisme ne remplissent pas toutes les conditions pour remplir une ligne de notre tableau. Voyons comment certains parmi ces modèles prennent en compte le temps ou les traitements.

- Modèles prenant en compte le temps.

En plus des concepts propres à la modélisation des données (entités, association, propriété) BUBENKO [BUB 77] a introduit le concept d'événement pour prendre en compte le temps lors de cette modélisation. L'événement étant défini ainsi :

" Un événement est une observation ou une décision survenant dans le système objet au temps t, du commencement ou de la fin d'une occurrence d'une association".

FALKENBERG [FAL 75] et BREUTMANN [BRE 79] intègrent également le temps dans la modélisation des données car ils considèrent que " toute association a un début et une fin considérés comme des événements sur l'axe des temps".

- Modèles intégrant les traitements.

Grâce à un langage associé au modèle de données le modèle DATA-SEMANTICS d'ABRIAL [ABR 74] permet la description des traitements à un niveau abstrait. Il satisfait ainsi aux critères de la seconde ligne de notre tableau.

DELOBEL [DEL 73] utilise les concepts du modèle de FORRESTER [FOR 62] pour compléter son modèle de données et représenter ainsi la dynamique du SI.

- Représentation des traitements et de la dynamique dans un schéma conceptuel exprimé à l'aide d'un modèle conceptuel de données.

Dans le modèle entité-association ou le modèle relationnel, pour effectuer une certaine représentation des traitements et de la dynamique on prend en compte des contraintes d'intégrité dans le schéma conceptuel.

Ces contraintes sont des conditions que doivent vérifier les données de la base pour assurer leur cohérence.

Ces contraintes sont dites statiques si elles portent sur l'état de la base, elles sont dynamiques si elles portent sur un changement d'état de la base. [FOU 82].

Dans les modèles basés sur la logique les contraintes dynamiques sont écrites sous forme de prédicats.

Cette représentation des traitements et de la dynamique présente plusieurs inconvénients [FOU 82] :

- concepts insuffisamment précis pour formaliser les phénomènes réels à décrire par les contraintes,

- juxtaposition des contraintes empêchant d'avoir une

vue d'ensemble de la dynamique du système,

- difficulté d'effectuer des contrôles pour assurer la complétude et la cohérence de la représentation effectuée.

En résumé on peut dire que les modèles orientés données ne satisfont pas au critère "formalisation" de la dynamique ni au critère "intégration des représentations des aspects statiques et dynamiques".

c) APPROCHES UTILISANT LES TYPES ABSTRAITS.

Les types abstraits par rapport au modèle de LANGEFORS et aux modèles conceptuels de données proposent de nouvelles structures de données plus représentatives de la réalité organisationnelle. Les techniques de spécification permettent de décrire l'image abstraite. L'outil de modélisation utilisé est le type abstrait ; il permet la " définition d'un ensemble d'objets par l'expression (axiomatique ou pré/post) des opérations servant à manipuler ces objets" [THI 85].

Les types abstraits sont utilisés pour exprimer un schéma conceptuel du SI. Ils paraissent bien adaptés à cette fonction puisqu'ils permettent une définition indépendante de toute consideration technique. Citons les travaux de EHRIG et WEBER [SAL 80]. En effet les données sont définies par une collection de types, les opérations par l'intermédiaire des opérations liées aux types, certains aspects de la dynamique tels que les conditions d'application des traitements par l'intermédiaire des préconditions et des post-conditions associées aux opérations.

Ces approches présentent l'inconvénient de ne pas spécifier complètement la dynamique au même niveau de formalisme que celui des données. Dans notre tableau ces approches satisfont les deux premières lignes (représentation intégrée de données et des traitements dans une orientation conception) et partiellement la quatrième (les données et la dynamique ne sont pas représentées de façon équivalente).

De nouveaux modèles intégrant les développements des types abstraits ont été proposés : citons SPES [QUE 84], orientés bases de données [BRO 82, VER 82] ou conception de SI [DUB 84], [WAS 82]. Citons la modélisation LASSIF [THI 85] qui s'appuie sur le modèle conceptuel REMORA en intégrant une "couche" type abstrait [ROL 82].

d) LES APPROCHES INTEGRANT LES MECANISMES D'AGREGATION ET DE GENERALISATION.

Ce sont des formes d'abstraction de données établies à partir des recherches effectuées par J.M. SMITH et D.C.P SMITH [SMI 77]. En plus de l'agrégation et de la généralisation, on parle d'autres formes d'abstraction de données telles que l'instanciation, la classification, l'association, la représentation et la partition.

- agrégation d'objets.

C'est une forme d'abstraction dans laquelle une relation entre divers objets est considérée comme une classe d'objets de niveau

d'agrégation plus élevée. En fait il s'agit d'une collection de parties.

Exemple : un employé est un agrégat à un niveau supérieur des objets : salaire, numéro employé, nom employé....

- <u>généralisation .</u>

C'est l'apparition d'un objet générique de niveau plus élevé qui permet de caractériser un ensemble de données (ou de traitements) par des propriétés communes à plusieurs classes de données (ou de traitements). Ce générique symbolisera la relation entre plusieurs classes d'objets tout en ne considérant pas leurs caractéristiques individuelles :

Exemple : "employé" est générique de secrétaire, d'ouvrier ménager qui sont tous des employés.

Les notions d'agrégation, de généralisation ont été par la suite complétées par des notions de classification et d'association. On peut citer les travaux de l'équipe TIGRE [ADI 85], VEL [84], de BIG [CHR 85], DEMOLOMBE [DEM 85] qui s'appuient sur ces divers concepts et offrent ainsi la possibilité d'utiliser de nouveaux méconismes d'abstraction.

I-3-2-1-3 <u>LES MODELES ORIENTES DYNAMIQUE</u>.

Ces modèles sont beaucoup plus récents que les modèles de données et les modèles orientés traitement.

Il est désormais admis par la communauté scientifique la nécessité de représenter dans le schéma conceptuel du SI aussi bien les données que la dynamique des traitements (rapport ISO [ISO 81]). La dynamique étant définie comme "la façon dont les éléments changent et les raisons pour lesquelles ils changent". Pour représenter cette dynamique plusieurs types d'approche sont proposés :

- approche orientée état : expression de la dynamique par une collection de contraintes d'intégrité. Cette approche correspond à la représentation des traitements et de la dynamique dans un schéma conceptuel exprimé à l'aide d'un modèle conceptuel de données que nous avons déjà vu dans les modèles conceptuels de données,

- approche orientée commande : il s'agit des modèles orientés traitement que nous avons déjà vus en I-3-2-1-1,

- approche orientée interaction : "modèles qui décrivent les influences réciproques entre l'organisation à modéliser et son environnement ainsi que les liens de causalité qui existent entre les changements d'état que cette organisation subit et les traitements à effectuer " [FOU 82]. Plusieurs modèles appartiennent à cette dernière approche : pour l'illustrer nous avons choisi arbitrairement deux propositions :

* <u>LA PROPOSITION DE BENCI,BODART,BOGAERT ET CABANES [BEN 76]</u>.

Elle est antérieure à toutes les autres approches, le schéma conceptuel est le résultat d'une organisation conceptuelle qui comprend :

. la structure conceptuelle basée sur le modèle entité-relation qui correspond au schéma conceptuel en base de données complété par des contraintes d'intégrité,

. les règles d'évolution décrivant comment les données évoluent au cours du temps ; elles définissent les actions à effectuer lorsque surviennent des inconvénients ainsi que les contraintes d'intégrité à vérifier pour l'application des règles. Les concepts utilisés sont l'événement, l'opération et la contrainte d'intégrité.

Un événement est défini comme "une décision qui déclenche l'exécution d'opérations sur les données".

Une opération désignera "une action exécutée par un acteur, un homme ou une machine, dans un lieu donné, à un instant donné, elle durera un certain laps de temps " [FOU 82].

Des paramètres associés aux évènements permettent d'effectuer la liaison entre modélisation des données et modélisation des traitements.

Ces auteurs ne fournissent pas un moyen de réprésenter le temps dans la modélisation.

La proposition de BENCI et AL satisfait l'ensemble des critères de la grille mais ne fournit pas une formalisation des concepts pour prendre en compte les traitements et la dynamique.

*LA PROPOSITION DE BRACCHI,FURTADO,PELLAGATTI [BRA 79].

Cette proposition est voisine de celle de BENCI et AL, elle définit le schéma conceptuel comme une association de trois sous-schémas :

. le sous-schéma de données défini par des entités, des relations, des attributs et des contraintes statiques,

. le sous-schéma fonctionnel défini par des contraintes dynamiques,

. le sous-schéma d'évolution défini par des événements et des règles de modification des données et du schéma fonctionnel.

Ces règles correspondent aux règles d'évolution de la proposition précédente, elles précisent les transactions qui sont en fait des combinaisons d'opérations.

La proposition de BRACCHI et AL répond à la plupart des critères d'un modèle conceptuel de SI à l'exception de la formalisation des concepts représentant la dynamique.

D'autres propositions ont été faites : citons le modèle conceptuel de MERISE [MER 79], le modèle de MAESTRO [DUF 80], le modèle de IDA [BOD 79], le modèle de REMORA [ROL 87] qui a été un précurseur dans le domaine de la dynamique.

Nous pouvons à présent remplir la grille d'évaluation des modèles dont nous avons évoqué les grandes lignes en parlant des moyens (voir I-3-2).

Caractéristiques / Composantes	prise en compte du temps	formalisation	orientation conception	donnée	traitement	commandes dynamiques	intégration des trois composantes
MODELE							
Modèle info-logique							
.langefors	<----------	----------	----------	----------	---------->		
.sundgreen	<----------	----------	----------	----------	---------->		
Modèle conceptuel de données .relationnel			<----------	----------	---------->		
DE.Dérivées			<----------	----------	---------->		
.Entité-Association							
. Modèle binaire	<----------	----------	----------	----------	---------->		
DONNEES							
. Modèle logique		<----------	----------	----------	---------->		
. Modèle sémantique		<----------	----------	----------	---------->		
. Modèle fonctionnel		<---->		<---->			
* Type abstrait	<----------	----------	----------	----------	---------->		<--------->
MODELE DE LE							
Modèle Méthode Analyse						<------>	
DE Modèle de données orienté trt. TRT					<----------	----------	---------->
MODELE DE LE							
Proposition de Benci	<------>		<----------	----------	----------	----------	---------->
OR. DYN							
Proposition de Bracchi	<------>		<----------	----------	----------	----------	---------->

<---------> Couverture totale

<---------> Couverture partielle

I-3-2-2 LES LANGAGES DE SPECIFICATION.

Il ont pour but de permettre une véritable spécification du SI à l'aide de la formalisation choisie.

LES LANGAGES EXISTANTS.

A l'heure actuelle on distingue en dehors des premiers langages de programmation : [THI 85]

- les langages de spécification généraux développés pour répondre à des domaines particuliers mais extensibles. On peut citer les langages SETL [KEN 75], ALICE [LAU 76], CIP [BAU 78], PROLOG [WAR 77], MEDEE [PAI 79,FIN 79].

- les langages relationnels en conception de bases de données. Ils sont destinés à la manipulation des relations et à leur définition. Ces langages utilisent soit l'algèbre relationnelle et le calcul des prédicats, c'est le cas du langage ALPHA défini par CODD [COD 70], soit le calcul relationnel comme le langage SQUARE [BOY 74], soit des aspects de ces trois théories comme le langage SEQUEL [CHA 74]. Ce type de langage bien que favorisant la manipulation des relations et l'expression des prédicats, ne permet pas l'expression des schémas de relations et de types de constituants,

- les langages utilisant les types abstraits : seuls les types abstraits permettent cette expression grâce au type RELATION : J.J CHABRIER a introduit le type n-uple labellé dans VEGA [CHA 83]. H.J SCHMITT [SCH 77] a spécifié le type RELATION dans le langage PASCAL-R, alors que A. WASSERMAN [WAS 82] l'a fait dans PLAIN, les deux langages PASCAL-R et PLAIN étant tous deux construits autour du noyau PASCAL [WIR 76]. On peut citer comme autres langages CLU [LIS 77], ATM [MIN 79] ainsi que le langage Z développé par ABRIAL [ABR 78, ABR 80]. De même les nouveaux langages de conception de SI développés dans CRIS-1 et CRIS-2 sont fortement appuyés sur les concepts de type abstrait. Citons SYSDUL [ASH 82], ACM/PCM [BRO 82], CIAM [GUS 82], SDLA [KNU 82], RIDL [VER 82], BASIS [WAS 82]. On peut également citer les langages de bases de données multimédia tels que le proposent TIGRE [ADI 85] et BIG [CHR 85],

- les langages graphiques : ils permettent la description sous forme graphique (flèches, cercles, rectangles ou autre convention de notation) de la statique et de la dynamique du SI, de spécifier le dialogue utilisateur-machine...

I-3-2-3 LES DEMARCHES .

Associée au modèle et au langage, la démarche permet une meilleure spécification en proposant au concepteur un canevas de conception. Pour aborder le problème de la conception, diverses approches existent [THI 85] :

- les démarches orientées "traitement" : il s'agit d'approches dans lesquelles la conception du SI est basée sur la définition d'une collection d'actions interdépendantes. Le SI se présente alors comme une boite noire considérée comme un processeur de traitement de l'information qui mémorise,

traite, met en forme et communique des données [ROL 87]. Dans CRIS-1 [IFI 82] on peut citer un exemple de ce type dans ISAC [LUN 82].

- les démarches orientées "données" : la résolution du problème de la conception passe par une structuration des données. Ces méthodes utilisent des formes d'abstraction reposant sur le schéma conceptuel des données ou plus directement sur les types abstraits : on peut citer dans CRIS-1 NIAM [VER 82] et ACM/PCM [BRO 80],

- certaines sont orientées dynamique :

. soit elles reposent sur la notion de condition d'état où le SI est vu comme un système générant des transitions d'un état à un autre : certaines de ces démarches utilisent les réseaux de PETRI pour représenter la dynamique [RIC 82,PET 77,LEO 81,BOD 79,ANT 81],

. soit elles sont orientées "événement" ou changement d'état dans le SI lié plus ou moins à des conditions de déclenchement de traitements associés. On peut citer [ANT 81,BEN 76,RID 78,BRE 79],

- certaines enfin essaient d'intégrer à la fois structuration des données, des traitements et de la dynamique des traitements sur les données : On peut citer REMORA [ROL 82], MERISE [LE 84], et IDA [BOD 84] qui se sont en plus orientées ces dernières années vers une modélisation par les types abstraits.

I-3-2-4 LES OUTILS.

Ce sont des supports fort utiles à la construction du schéma conceptuel, au contrôle de sa validité et à son évaluation. Ils peuvent selon leur qualité faciliter la transition d'un niveau de représentation à un autre et constituer des aides efficaces lors de la conception. Des outils peuvent être utilisés dans toutes les phases du cycle de vie du logiciel afin d'améliorer la qualité de la conception et de la réalisation.

On distingue plusieurs types d'outils :

- les outils d'aide à la spécification : ils aident le concepteur à construire le schéma conceptuel du SI, à gérer l'évolution de la structure du SI. Ils peuvent se présenter sous la forme d'outils de conception assistée par ordinateur : outils généralement conversationnels et directifs avec utilisation d'un clavier ou d'un écran graphique. Utilisé sous forme de mémoire, l'outil ne fait qu'une vérification syntaxique du schéma conceptuel et ne vérifie pas la cohérence de ce schéma,

- les outils de contrôle : ils servent à vérifier la cohérence du schéma conceptuel : contrôle des informations entrées, de la validité des contraintes d'intégrité, contrôle des spécifications fournies ....

LES ATELIERS DE GENIE LOGICIEL (A.G.L).

Un concept élaboré vers les années 1967 (à la conférence de l'OTAN en 1968 sur le GENIE LOGICIEL utilisation pour la première fois de l'expression "SOFTWARE ENGINEERING") [GEN 88] celui des

A.G.L fait aujourd'hui l'objet de nombreuses publications et réalisations.

On entend par A.G.L l'ensemble des outils intervenant de façon intégrée dans la conception et la réalisation d'un logiciel ceci durant son cycle de vie. L'atelier désigne en fait une organisation ou une structure d'accueil qui permet la prise en charge des tâches répétitives et fastidieuses, offrant ainsi la possibilité aux concepteurs de s'adonner aux travaux les plus créatifs.

OBJECTIFS :

- améliorer la productivité informatique,

- améliorer le développement qualitatif des logiciels,

- alléger la charge de la machine d'exploitation, à partir du concept de machine dédiée au développement,

- tendre vers une plus grande professionalisation en vue d'une industrialisation du logiciel.

CONCEPTS :

L'A.G.L est donc construit autour d'un certain nombre de concepts : le cycle de vie, la structure d'accueil, les méthodes, les outils, les langages de programmation.

- le cycle de vie comme nous l'avons déja vu commmence depuis la définition des besoins en passant par les différentes phases de conception jusqu'à la phase de maintenance,

- la structure d'accueil est chargée d'intégrer outils et méthodes et d'automatiser la conduite et le contrôle des projets. Elle comprend en général un poste de travail et des terminaux qui servent d'interface utilisateur,

- des outils de base intégrés à la structure d'accueil, indépen - dants autant que possible des méthodes. Ces outils sont associés aux outils propres aux utilisateurs ; ils interviendront durant tout ou parti du cycle de vie du logiciel. On distingue :

.des outils de spécification, d'aide à la conception,

.des outils d'aide à la production (éditeurs de textes de programme, de détection et de correction d'erreurs, d'optimisation des performances,..., outils de documentation, outils de maintenance et de gestion, outils de simulation et de prototypage...),

Ces outils pourront communiquer entre eux grâce à des réseaux donnant ainsi lieu à des stations de travail conviviales avec possibilité de fenêtrage, représentations graphiques...;

- des méthodes couvrant la totalité ou une partie du cycle de vie,

- des langages de programmation : langages classiques ou

ou langages dits de quatrième ou de cinquième génération.

Ces outils doivent constituer un tout intégré dans le cadre d'un environnement d'A.G.L.

Comme exemple de réalisation, l'équipe REMORA dans le cadre de l'A.G.L appelé L.G.S.I (Logiciel de Gestion de SI) a conçu deux prototypes éléments de cet atelier [THI 85] :

- le logiciel d'aide à la construction de la spécification d'un SI (LASSIF),

- le logiciel d'interprétation de la spécification du SI (LISSIF) qui hérite de la spécification obtenue à partir de LASSIF pour construire le SI résultant, le faire évoluer et l'utiliser.

Pour citer des exemples d'environnement de programmation comme élément d'un A.G.L on peut parler de MENTOR [DON 75] pour la manipulation d'informations structurées, CORNELL/SYNTHETISER [TEI 81] environnement interactif pour la mise au point des programmes, le projet MAIDAY [GUY 84] basé sur la programmation déductive, CONCEPTOR (pour la conception assistée) et DELTA (pour servir d'atelier flexible de développement d'applications portables)....

D'autres A.G.L ont vu le jour, on peut citer : PAC BASE de CGI (CORIG), MAESTRO de PHILPS DATA SYSTEM, LINC de BURROUGHS, MULTIPRO de CAP-SOGETI, PROTEE-5000 de SIS, SOFTPEN de IPI, CONCERTO de CNET, XTOP de GFI, VULCAIN de CERCI.

PERSPECTIVES :

Aujourd'hui les A.G.L s'intéressent aux systèmes experts en intégrant de nouveaux outils orientés systèmes experts, outils qui auront à gérer de nouveaux besoins tels que la manipulation d'informations symboliques, le traitement d'expertises, la manipulation de connaissances représentées par des systèmes de formalisations mixtes....

Malgré quelques réalisations remarquables (MAESTRO de TRT-TI, CONCERTO développé par le CNET, EMERAUDE de l'ADI, MULTIPRO de CAP-SOGETI, l'atelier de SEMA-METRA, NEXUS générateur de systèmes experts hybrides de MIND SOFT, ATX un atelier de génération de systèmes experts pour l'aide à la conception de la société APSIDE TECHNOLOGIES, NEMO outil de développement de systèmes experts temps réel de la société S2O....), force est de constater que l'atelier idéal (celui qui recouvre intégralement le cycle de vie) n'existe pas encore.

I-4 LES METHODES RETENUES.

L'analyse que nous venons d'effectuer montre les divers thèmes qui forment notre cadre pour une analyse comparée des diverses méthodes de conception de SI.

Rappelons que notre étude ne concerne que le niveau conceptuel ; en résumé pour chaque méthode retenue nous nous intéresserons :

- aux niveaux de description couverts. Nous examinerons le niveau d'abstraction atteint par la méthode ainsi que la manière dont se déroule le cycle de vie,

- aux moyens de conception de SI mis en oeuvre et qui sont constitués :

. des modèles pour construire la solution de conception de SI,

. des langages pour exprimer la solution à partir des concepts,

. des méthodes ou différentes manières d'aborder la conception,

. des outils pour la résolution automatique des problèmes (documentation, réalisation ...) soulevés au cours de la conception.

I-4-1 QUELQUES METHODES NON ETUDIEES.

Nous avons étudié des méthodes développées dans CRIS-1 [IFI 82], analysées et comparées lors de CRIS-2 [IFI 83] et deux autres méthodes présentées dans CRIS-3 [IFI 84]. Ceci d'abord pour des raisons de documentation et aussi parce que ces méthodes rentraient dans le cadre que nous avons défini. Nous n'avons pas par contre étudié certaines méthodes qui ne répondaient pas à nos préoccupations. On peut citer : [THI 85]

- D2S2 [MAC 82] méthode qui permet un développement coordonné d'applications à l'aide du modèle entité/association . C'est une méthode très orientée "utilisateur" et essentiellement manuelle,

- ISAC [LUN 82] méthode qui facilite la communication utilisateurs-analystes, elle est très graphique et n'utilise pas la modélisation des données ; elle est orientée "traitements" et est essentiellement manuelle,

- EDM [RZE 82] méthode d'analyse "traditionnelle" couvrant la totalité du cycle de vie du SI. De nombreux outils y ont été développés,

- IML [RIC 82] outil de description utilisable dans un processus de conception qu'on peut intégrer dans n'importe quelle méthode de conception de SI. Cette méthode utilise la représentation graphique des réseaux de PETRI [PET 80] pour modéliser les traitements,

- DAISEE [SOL 82] C'est une méthode qui propose modèles et

outils d'analyse, de conception et d'implémentation de projet. Elle est essentiellement manuelle ; elle utilise le modèle entité association,

- CIAM [GUS 82] méthode qui couvre l'analyse des besoins jusqu'à la conception des traitements. Elle est orientée données et préconise l'utilisation d'un modèle entité-association intégrant le temps. C'est une méthode assistée par un système d'aide à la conception automatisé.

Nous avons rejeté les méthodes ci-dessus parce qu'en général elles sont trop orientées "analyse des besoins" et sont d'un niveau d'abstraction peu élevé. Certaines ne proposent pas de modélisation de la dynamique et de la structure du SI ; certaines modélisent seulement les données, d'autres seulement les traitements.

I-4-2 LES METHODES ETUDIEES.

Nous avons limité notre choix à cinq méthodes (ACM/PCM, NIAM, REMORA, USE, DADES) que CRIS-1 a considérées comme faisant partie des des méthodes les plus réprésentatives de la conception des SI. Nous avons ajouté à ces cinq méthodes MERISE et IDA qui ont été l'objet des travaux de CRIS 3 .

Plusieurs raisons ont guidé notre choix :

1) les méthodes à étudier pour la plupart ont un niveau d'abstraction et de spécification suffisamment élevé ; elles ont accordé une grande importance à la modélisation abstraite du monde réel, les concepts qu'elles manipulent sont relativement élaborés,

2) elles sont suffisamment variées dans leur orientation de conception : certaines sont orientées données, d'autres orientées traitement, certaines intégrent données et traitements, certaines prennent en compte la dynamique. Ceci montre la diversité des approches et ne peut qu'enrichir notre étude,

3) elles ne sont pas "traditionnelles" dans la mesure où elles ne sont pas trop orientées analyse des besoins et elles intègrent certains aspects des nouveaux développements en matière de conception de SI,

4) pour des raisons de documentation nous avons limité notre étude à sept méthodes développées dans CRIS-1, CRIS-2 et CRIS-3 qui se sont toutes attachées à la définition du niveau conceptuel à des degrés divers, et qui rentrent dans le cadre de réflexion que nous avons fixé.

Pour résumer disons que ce cadre de compraraison s'articule autour de deux points :

1) étude de la situation de chaque méthode par rapport au cycle d'abstraction et aux trois niveaux qui le composent en mettant l'accent sur le niveau conceptuel et en voyant si la méthode couvre ou non tout ou partie du cycle de conception du SI,

2) étude des moyens de spécification proposés par la méthode au niveau du cycle de contrôle.

Dans les pages qui vont suivre nous allons étudier chacune des méthodes que nous avons choisies en nous servant du cadre que nous avons défini et des thèmes que nous avons déjà développés tout en émettant certaines critiques au fur et à mesure de nos investigations.

PARTIE II

ETUDE DE DIVERSES METHODES DE CONCEPTION DE SI.

II-1 INTRODUCTION GENERALE.

Pour analyser les méthodes à travers le cadre que nous venons d'évoquer et qui s'articule autour de deux grands axes :

- l'étude des diférents niveaux de description et des moyens utilisés surtout au niveau conceptuel,

- l'étude des outils disponibles au niveau du cycle de contrôle.

Nous nous servirons d'exemples tirés de l'IFIP. Dans l'ordre nous étudierons les méthodes suivantes :

- USE,
- ACM/PCM,
- DADES,
- NIAM,
- REMORA,
- IDA,
- MERISE .

II-2 ANALYSE DETAILEE DES METHODES.

II-2-1 LA METHODE USE.

A / PRESENTATION.

Le projet USE (USER SOFTWARE ENGINEERING) a été entamé en 1975 à l'université de SAN FRANCISCO aux USA par l'équipe de WASSERMAN [WAS 78, WAS 81] avec pour objectif initial de développer une méthode de spécification, de conception et d'implémentation de systèmes informatiques interactifs. Cette méthode s'appuie sur des outils d'aide à la conception, un langage (PLAIN) orienté manipulation et développé autour d'un noyau PASCAL [WIR 76]. Elle met l'accent sur les aspects développement méthodique du logiciel et participation des utilisateurs. Dès les premières étapes du processus de developpement des logiciels, l'aspect externe est enrichi par la possibilité de créer ou de modifier des prototypes permettant aux utilisateurs d'évaluer réellement le système et de le corriger.

Les récents développements avec les travaux de N.LEVESON [LEV 80] enrichissent la méthode USE en introduisant les types abstraits comme nouvelle étape de spécification en amont des étapes de conception/réalisation effectuées avec USE et PLAIN.

Cette méthode est à l'heure actuelle en cours d'exploitation et d'évolution. Elle est commercialisée en FRANCE par la société I.G.L dans l'atelier STP.

OBJECTIFS.

Après une étude approfondie de l'état de l'art en matière de conception de SI interactifs, USE s'est fixé les objectifs suivants :

- gérer les évolutions du système, produire un logiciel portable, structuré et documenté,

- produire un prototype du système à mettre en oeuvre comme outil d'aide à l'analyse, à la spécification et à la validation du logiciel élaboré,

- gérer de façon appropriée le processus de conception et de développement en aidant à la compréhension du problème par les développeurs, les utilisateurs et toutes les parties concernées pendant la phase d'analyse et de spécification.

B / NIVEAUX D'ABSTRACTION.

USE couvre les trois cycles définis par BODART et AL à savoir les cycles d'abstraction, de contrôle, de décision. Cette méthode peut être située à un "haut niveau d'abstraction" vu son caractère complet et les concepts qui la sous-tendent.

C / LES CONCEPTS.

Les concepts sous-jacents relatifs à la modélisation de la

structure ou à celle de la dynamique ne sont pas exprimés de façon explicite dans le document élaboré par CRIS-1. Il en est de même du modèle utilisé. Cependant les spécifications en PLAIN montrent que USE préconise le modèle relationnel pour les données et les diagrammes de transition pour la dynamique. Elle a choisi de spécifier la base de données comme un ensemble de relations normalisées.

USE utilise les notions d'abstraction de données par les types abstraits pour définir lors de l'étape de spécification les objets abstraits et les opérations qui leur sont associées.

i) Objets abstraits : Il s'agit en fait d'entités au sens traditionnel du terme (exemple une "communication") sur lesquelles on définit formellement au sens des types abstraits un certain nombre d'opérations. La démarche de définition appelée BASIS (BEHAVORIAL APPROACH TO THE SPECIFICATION OF INFORMATION SYSTEMS) pour chaque opération définie formellement sur un objet abstrait inclut une spécification des divers éléments qui contribuent à la définition complète de son sens. Ces éléments sont :

- l'image abstraite ou l'objet réellement concerné auquel s'ajoutent d'autres éléments attachés à cet objet,

- l'invariant, ou condition à remplir par l'objet abstrait dans le cadre défini par le réel perçu,

- les contraintes d'entrée et de sortie exprimées sous forme de pré et post-conditions. Tout objet sera ensuite associé à une relation en PLAIN.

ii) Pré-conditions, post-conditions et invariants.

Ils permettent d'introduire des contraintes d'intégrité sur la base de données. On distingue deux types d'invariants sémantiques :

- les contraintes de domaine (plage de valeurs),

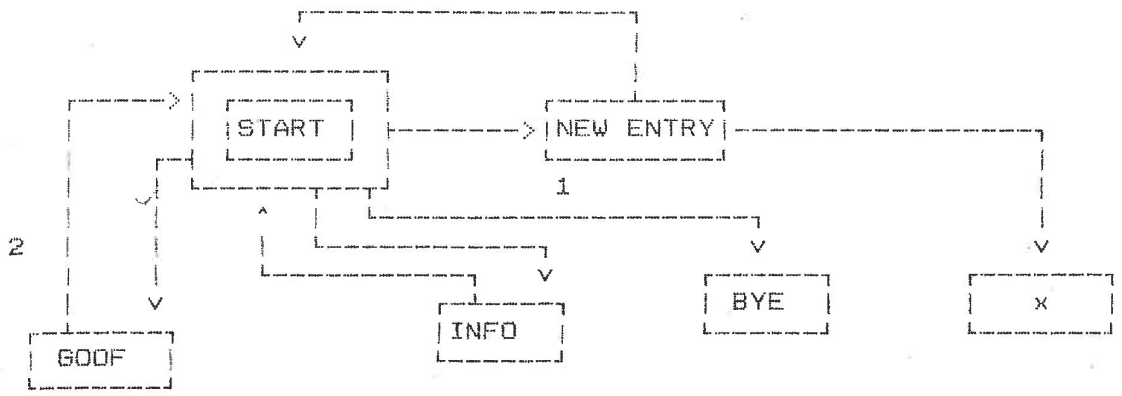
- les contraintes inter-objets qui sont les contraintes de relation entre divers objets (exemple : un papier ne peut être affecté à plus d'une session).

Les pré-conditions visent à éliminer les séquences illégales d'opérations et des techniques de vérification peuvent être utilisées pour montrer le caractère vrai des post-conditions.

Les pré-conditions, post-conditions et invariants ainsi definis seront ultérieurement implémentées grâce à des types de définition dans le langage d'implémentation PLAIN.

iii) Transition : les opérations sont des actions qui portent sur sur les données. Pour représenter la dynamique et associer les actions à des entrées spécifiques, chaque opération devra être liée à tout ou partie d'une action dans un diagramme de transition. Celui-ci est un réseau de noeuds et de chemins directs. Chaque chemin peut contenir un message (une chaine de caractères) ou le nom d'un autre diagramme. S'il est vide, il sera traversé comme une case normale si les chemins menant au noeud donné ne sont pas sollicités. Ces diagrammes sont utilisés pour modéliser le dialogue utilisateur/machine.

exemple : diagramme de transition de commandes initiales du Système



---> indication du chemin sollicité.

MESSAGES

START ="SVP TAPEZ une commande"
BYE ="BYE BYE"
INFO =" LES COMMANDES VALIDES SONT..."
GOOF =" COMMANDE ILLEGALE"

ACTIONS

1 Fermer la base de données

2 Mise à jour erreur utilisateur

COMMENTAIRE :

Le message associé à START est affiché. Si celui-ci est "QUIT" le chemin qui va de START à BYE sera traversé et l'action 1 associée sera effectuée et le message associé au noeud BYE sera affiché. De même si le message est "HELP", alors on va en INFO puis retour en START. Si le message est "entrée", on va traiter le diagramme NEWENTRY qui retourne les codes 0 ou 1 qui déterminent alors le chemin suivi et l'action à réaliser après ; en cas d'erreur on va en GOOF et on effectue l'action 2

Le diagramme de transition est un bon outil d'aide à la conception et il est facile à assimiler par les utilisateurs. Il constitue par conséquent un bon moyen de validation.

iv) Structure de la spécification complète.

Pour USE la spécification des SI interactifs se compose des trois éléments suivants :

1) Un ensemble de diagrammes de transition représentant le dialogue utilisateur-machine,

2) une base de données qui est un ensemble de relations normalisées déduites des objets abstraits,

3) les opérations portant sur les objets abstraits avec sa description formelle et informelle.
Pour chaque opération liée au diagramme son nom d'action, sa des-

cription formelle et informelle.

D / LE LANGAGE.

On distingue plusieurs langages :

i) Un langage graphique adapté au dialogue utilisateur-programme qui sert d'entrée à l'interprète de diagramme de transition de (TDI) (voir exemple ci-dessus). Avec ce langage on pourra traduire les noeuds, traiter les erreurs. Une fois que l'on a effectué l'encodage, TDI produit un programme exécutable simulant l'interface spécifié.

Exemple d'encodage par TDI du diagramme de transition

de l'exemple précédent :

```
DIAGRAM  figure  1   ENTRY START EXIT X

NODE START
   CS "please type a command"

NODE BYE
    "BYE BYE"

NODE INFO
    R$-2, " VALID COMMANDS ARE ..."

NODE GOOF
    R$,RV, "ILLEGAL COMMAND",SV

ARC   START
    ON "ENTRY" TO <NEWENTRY>
    ON "QUIT" DO 1 TO BYE
    ON "HELP" TO INFO
   ELSE DO 2 TO GOOF

ARC BYE SKIP TO X

ARC INFO SKIP TO START

ARC GOOF SKIP TO START

ARC < NEWENTRY >

  WHEN 0 TO START
  WHEN 1 TO X
```

ii) Un langage de spécification des objets abstraits en terme de pré et post-conditions, les traitements étant exprimés à l'aide d'un langage déclaratif basé sur le calcul de prédicat du premier ordre.

a) Exemple 1 : Soit l'objet abstrait papier à définir :

```
object  paper

     abstract image
        title :
         authors : set of author
          STATUS : PAPERSTATUS
```

```
        PAPERNO : 1 .. 300

    abstract invariant

    operations
       receive PAPER(PAPERTIME:TEXT,SUBMITTERS:SET OF AUTHORS)
               RETURNP:PAPER

       POST
       TIME = PAPERTIME
       AUTHORS = SUBMITTERS
       STATUS  = RECEIVED
       PAPERNO = CARDINALITY (PAPER)
    REVIEW PAPER (P:PAPER)
       PREP.STATUS = RECEIVED
       POSTP.STATUS = INREVIEW
    ACCEPT PAPER (P: PAPER)
       PREP.STATUS = INREVIEW
       POSTP. STATUS = ACCEPTED
    ASSIGN PAPER TO SESSION (P: PAPER,S:SESSION)
       PREP.STATUS = ACCEPTED  $S:PAPER COUNT < MAXPAPERS
       POST P.STATUS = INSESSION
    REJECT PAPER (P : PAPER)
       PREP.STATUS = INREVIEW
       POSTP.STATUS=REJECTED
    ACCEPTED ? (P:PAPER) RETURNSb:BOOLEAN
       POSTB = (P.STATUS=ACCEPTED p.STATUS=INSESSION)
    CHANGE PAPER TITLE (NEWTITLE : TEXT,P:PAPER)
       POSTP.TITLE=NEWTITLE
    CHANGE AUTHORSHIP (NEWAUTHORS:SET OF AUTHORS,P:PAPER)
       POSTP.authors=NEWAUTHORS
```

b) Exemple 2 : spécification formelle et informelle :

cas : définition de l'affectation d'un papier à un referee (il s'agit d'une opération).

```
    OPERATION SELECT REFS

    NAME ASSIGN REFEREE (P : PAPER)

    INFORMAL

    (définition informelle c'est à dire narrative de l'opération).

    FORMAL

    précondition ----->  PRE P. COUNTERS >= 0 & P. COUNTERS < MAXREFS

                         &V. NUMBER - ASSIGNED < MAXPAPERS

    postcondition ----> POST IS REFEREE (r,p) &P.

                         COUNTREFS=P.COUNT REFS+1

                         &DATE SENT (r,p) = [TODAY'S DATE] &- (V.NAME

                         IN P.AUTHORS)-
```

Commentaire : Ici l'expression de l'opération est faite de façon (narrative) et formelle ; la spécification formelle étant basée sur des axiomes et des conditions de validation.

iii) Un langage d'expression des relations normalisées de la base de données associée au SGBD TROLL. On exprimera ici les contraintes de domaine et les contraintes de relation entre divers objets, certains de ces domaines étant définis avec leurs relations.

Exemple :

```
domain PAPERRANGE : INTEGER (1..100);
domain SESSIONRANGE : INTEGER(1..30);
-
-
domain PAPERSTATUS: SCALAR (RECEIVED,INREVIEW,
                            ACCEPTED,INSESSION,REJECTED);

RELATION ACCEPTED PAPERS [KEY PAPERNO] OF
        PAPERNO :PAPERRANGE;
        TITLE :STRING;
        SESSION NUM : SESSIONRANGE;
 END;
 -
 -

 RELATION PC LIST [ KEY NAME ] OF
    NAME : PERSON
  PAPERCOUNT : INTEGER (0..10);[NO PC member HANDLES MORE
                                THAN 10 PAPERS]
 END;
```

iv) Un langage d'implémentation PLAIN : langage proche du PASCAL relationnel et qui est utilisé dans l'environnement logiciel.

Exemple:

Soit l'opération déjà traitée "ASSIGN-referee"
affectation d'un papier à un referee

```
CALL ASSIGN-REFEREE (NAME,PAPER-NO,TODAY)
```

La notation des paramètres dépend de leur position dans la notation TROLL :$0 sert de paramètre formel du nom.
$1 sert de paramètre formel du numéro du papier
$2 sert de paramètre formel pour TODAY
"REVIEWING" est la relation où on insère un nuple.

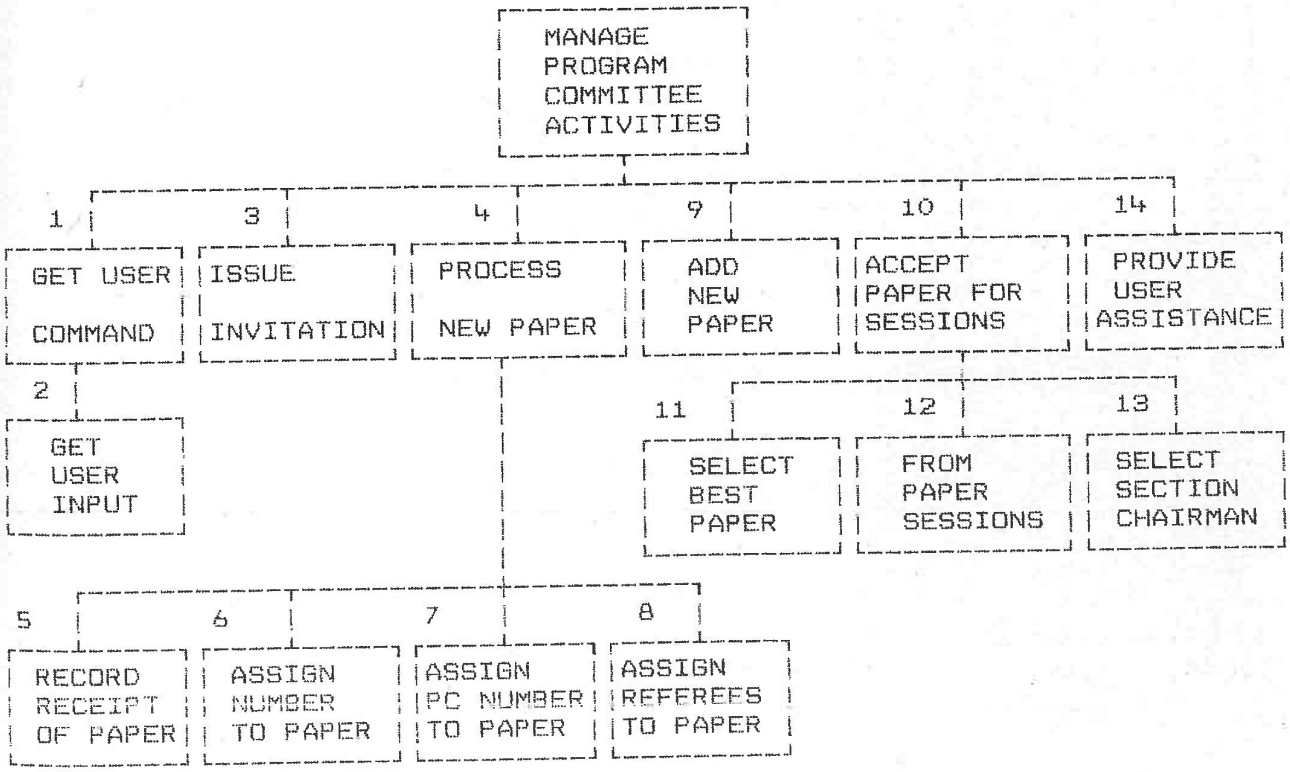
```
ASSIGN REFEREE
OPEN CONFERENCE;
IMPORT REFEREE - LIST;
IMPORT REVIEWING;

Preliminary version ... OMIT CHECKS ON NUMBER OF PAPERS
  ASSIGN AND ON SELF-REFEREEING

IF EXISTS [REFEREE-LIST[&O] THEN
  begin PRINT 'UNKNOWN REFEREE NAME'; END
```

```
ELSE  VALID REFEREE NAME

begin
  REFEREE LIST[&O]. NUMBER ASSIGNED:=

  REFEREE-LIST[&O] NUMBER-ASSIGNED + 1
 INSERT REVIEWING [$O],$1,$2];
END;
```

v) <u>Un langage de conception de programme</u> destiné à faciliter la compréhension du programme :

Après la phase de spécification viennent les phases de conception architecturale et de conception détaillée. La conception architecturale s'intéresse à la conception globale du traitement et à la conception des données.

<u>Exemple de structure du SI concernant le comité programme</u> :



La conception du programme relatif au comité de programme se présente sous la forme d'un arbre avec six modules comme branches dont chacune peut à son tour se décomposer en d'autres modules. Ainsi le module 4 donnera lieu à quatre sous-modules.

Le SI est structuré en modules, les entrées et les sorties seront définies avec les interfaces, les fonctions aussi seront spécifiées de façon appropriée.

La conception détaillée consiste en une étude de chaque module. Bien qu'utilisant la démarche structurée USE estime que celle-ci est peu appropriée pour les opérations portant sur les bases de données. C'est l'une des raisons qui a poussé cette méthode à se servir d'un langage de conception de programme dont le but est d'exprimer la logique du module de programme sous une forme narrative afin que celui-ci soit compris par les autres concepteurs.

Exemple: langage USE de conception de programme concernant le module "ASSIGN REFEREE"

```
MODULE ASSIGN REFEREE
INPUT
   PAPER-NO: PAPERRANGE
   PAPERS, AUTHOR LIST, REFEREE LIST, REVIEWING: RELATION;
OUTPUT
   PAPERS , REFEREE LIST, REVIEWING: RELATION;
  (ALL THREE RELATIONS MODIFIED BY THIS MODULE)

CALLS
CALLED BY
  NEW PAPER
LOCAL DATA
  INPUT : STRING; [USER INPUT OF NAME(S)]
  TEMP : MARKING ON AUTHOR LIST

FUNCTION
    description narrative de la fonction accomplie par ce
    module

 ALGORITHM
    WRITE 'REFEREE NAME':
    READ INPUT;

            explication du traitement à effectuer sous forme
            narrative.
    IF
    -
    ELSE
    -
 EXCEPTIONS
 END MODULE
```

E / LES OUTILS.

Tout au long du déroulement du cycle de vie du logiciel USE propose de nombreux outils :

- TDI (Transition Diagram Interpreter) : il est destiné à créer les diagrammes de transition, il permet aussi une construction et une modification rapides des prototypes utilisateurs-programmes. TDI effectue des contrôles (par exemple s'assurer qu'il y'a un chemin à partir de chaque noeud), mesure les performances des utilisateurs lors du test du système partiel (facilité d'utilisation, taux d'erreur...). On peut également utiliser le graphisme avec TDI et gérer le curseur.

- TROLL (SGBD relationnel) : cet outil propose un interface de

type algèbre relationnelle pour un petit système de gestion de bases de données relationnelles. Il sert lors de l'exécution de programmes de PLAIN et lors de la création de systèmes partiels avec TDI.

- RAPID (RAPID PROTOTYPES OF INTERACTIVE DIALOGUES) : cet outil combine TDI et TROLL pour la construction rapide de systèmes partiels. On peut ainsi associer la manipulation de données et/ou l'utilisation de routines écrites dans un langage de haut niveau pour permettre le dialogue utilisateur-programme.

- USE CONTROL SYSTEM : outil destiné à la construction modulaire du logiciel et qui fournit un cadre de développement standard. UNIX a été choisi comme support du développement du logiciel.

- Tous ces outils associés forment l'environnement logiciel de la méthode appelée << Unified Support Environnement >> (USE).

F / ETAPES.

Le cycle de conception de USE se déroule selon les étapes suivantes :

1) - Analyse des besoins : connaissance du problème posé, modélisation des traitements grâce à la méthode SSA [GANE 79], création d'une base de données relationnelles, construction d'un modèle conceptuel de la BDD, procédures de validation avec utilisateurs.

2) - Conception du dialogue utilisateur-programme.

3) - Création d'une maquette du dialogue avec possibilité d'effectuer des mises à jour.

4) - Spécification informelle du SI grâce à la simple narration.

5) - Conception préliminaire de la base de données relationnelles.

6) - Création d'un système partiel fournissant dans la mesure du possible toutes les fonctions du SI.

7) - Spécification formelle des opérations du SI.

8) - Conception de l'architecture globale du SI et des modules.

9) - Implémentation en PLAIN.

10) - Divers tests et vérification de la conformité du logiciel.

USE se sert de techniques existantes lors de la réalisation de certaines étapes, pour d'autres elle a élaboré des techniques spécifiques.

G / CONCLUSION.

USE se présente comme une méthode complète de développement (depuis l'analyse des besoins jusqu'aux tests et vérification de la conformité du logiciel) pour la conception/réalisation de SI.

i) Ses points forts sont :

. L'importance et le soin accordés à la spécification : en effet en plus des conditions attendues d'une spécification que nous avons déjà évoquées dans la première partie (minimalité, fidélité, cohérence, complétude, souplesse et indépendance vis à vis des usagers) USE s'attache à assurer les qualités suivantes de la spécification : testabilité (possibilité de quantifier de façon précise la spécification en vue de la tester), et possibilité de traduire la spécification,

. couverture du cycle de conception : On l'a vu USE est une méthode complète de comception/réalisation,

. modélisation à l'aide des types abstraits,

. l'intervention de l'utilisateur à toutes les étapes du cycle de vie ; importance accordée au traitement interactif,

. nombreux outils pour la simulation et la construction de prototypes.

ii) Points faibles :

- manque d'informations sur le cycle de management,

- caractère non explicite du modèle conceptuel,

- la phase d'enchainement avec BASIS n'est pas systématiquement liée à la phase de réalisation avec USE et PLAIN ; au lieu d'être complémentaires c'est à dire enchainées, ces deux phases paraissent juxtaposées,

- un environnement logiciel coûteux pour le développement de gros SI,

- USE parait en fait surtout utilisable dans le cas de SI transactionnels de petite ou moyenne taille.

II-2-2 LA METHODE ACM/PCM (ACTIVE AND PASSIVE COMPONENT MODELLING).

A / PRESENTATION.

La méthode ACM/PCM a été développée ces six dernière années à l'Université de MARYLAND aux USA par l'équipe de MICHAEL BRODIE.

Cette méthode est basée sur les travaux effectués en base de données et sur les langages de programmation ; elle reprend en particulier les travaux de SMITH et SMITH [SMI 77]. La conception de la structure (données) et la conception de la dynamique (transition) sont faites en parallèle, le tout débouche sur un modèle abstrait. Celui-ci est alors transformé en un programme plus détaillé. Cette méthode est actuellement en cours d'exploitation et de développement dans plusieurs entreprises aux U.S.A.

OBJECTIFS : Proposer des moyens qui utilisent la modélisation en vue de développer des SI complexes de grande ou de moyenne envergure, tournés vers les transitions sur les bases de données interactives dont on devra gérer la complexité et dont il faudra également définir et assurer un haut degré sémantique.

B / NIVEAU D'ABSTRACTION.

ACM/PCM s'appuie solidement sur le principe de l'abstraction. Aussi bien le traitement des aspects statiques que celui des aspects dynamiques ainsi que la gestion de l'évolution des SI s'effectuent de manière explicite et selon le principe de l'abstraction. Cette méthode couvre le cycle d'abstraction et en partie le cycle de contrôle : On distingue trois niveaux d'abstraction identiques à ceux définis par ANSI/SPARC :

- le niveau de transaction (prise en considération des vues particulières des utilisateurs),
- le niveau conceptuel (débouchant sur un schéma conceptuel),
- le niveau de la base de données (ou niveau interne).

De par ses objectifs et ses concepts sous-jacents, ACM/PCM peut être située à un "haut niveau" d'abstraction.

C / CONCEPTS.

C-1 Modélisation de la structure.

Pour la conception et la spécification des propriétés structurelles des applications de bases de données, ACM/PCM propose une extension du modèle SHM [SMI 77] appelée SHM+ (modèle hiérarchique sémantique étendu). SHM+ repose sur le concept structurel d' "objet" et sur quatre formes d'abstraction relatives à des objets. Ces quatre formes sont celles proposées dans [SMI 77] (l'agrégation et la généralisation) ainsi que l'association et la classification.

i) Classification.

C'est une forme d'abstraction dans laquelle une collection d'objets est considérée comme une classe d'objets de haut niveau. On entend par classe d'objet tous les objets de la collection qui partagent les mêmes caractéristiques. Un objet est donc un élément d'une classe d'objet s'il possède les propriétés définies dans la classe. La classification induit la notion d'instanciation (réalisation) d'un objet dans une classe d'objet. On introduit ainsi la relation "occurrence". Cette notion de classification correspond à la notion de classe, de type, de population qu'on rencontre généralement dans la conception de base de données.

Exemple :

Classe d'objet "employé" avec les propriétés nom-employé, numéro-employé, salaire peut avoir comme instance de l'objet : John (nom), 402 (numéro), 20000 (salaire).

ii) Agrégation.

C'est une forme d'abstraction dans laquelle une relation entre les objets composants est considérée comme un objet agrégé de niveau plus élevé. Elle permet d'exprimer les propriétés des objets d'une classe.

Exemple : un "employé" est un objet agrégé de numéro-employé, nom-employé, salaire.

Cette forme d'abstraction introduit la relation "partie de".

iii) Généralisation.

C'est une forme d'abstraction dans laquelle une relation entre les objets d'une catégorie est considérée comme un objet générique de niveau plus élevé. Cette forme permet d'exprimer des relations hiérarchiques entre les objets.

exemple : le générique "employé" peut être une généralisation des catégories "secrétaire" et "manager". Cette forme d'abstraction induit la spécialisation d'où la relation "est un".

iv) association.

C'est une forme d'abstraction dans laquelle une relation d'objets membres est considérée comme un ensemble objet de niveau plus élevé. Cette forme permet de définir des sous-ensembles dans une classe d'objets.

Exemple : l'ensemble "syndicat" est une association des employés membres.

Cette relation induit la relation "membre de".

v) Modélisation effective de la structure.

Pour modéliser la structure on utilise de façon répétée les techniques de composition/décomposition ce qui aboutit à des hiérarchies d'association et d'agrégation ; les techniques de

généralisation/spécialisation fournissent des hiérarchies de types.

C-2 Modélisation de la dynamique .

Avec SHM+ la modélisation de la dynamique est faite en même temps que celle de la structure. En ce qui concerne la conception et la spécification des propriétés dynamiques des bases de données, SHM+ fournit les primitives d'opérations portant sur les objets et trois abstractions de contrôle avec lesquelles on compose des opérations orientées applications.

i) Opérations primitives : chaque opération sur la base de données est soit une opération de modification ou d'interrogation qui ne porte que sur une seule occurrence d'une classe d'objet. Les opérations permises sont :

- en modification : si on prend l'objet "employé" on peut avoir par exemple : INSERT (insérer un employé), DELETE (supprimer un employé), UPDATE (mettre à jour le salaire d'un employé).

- en interrogation : FIND (trouver un objet dans la base de données), CREATE (créer un objet en utilisant une procédure définie), et REQUEST (chercher un objet de façon interactive).

ii) Abstractions de contrôle : elles sont utilisées pour relier des opérations en vue de former des opérations de haut niveau. Les trois formes d'abstraction de contrôle (séquence, choix, répétition) qui sont des concepts de la dynamique, correspondent aux trois formes d'abstraction de la structure.

a) l'agrégation correspond à la séquence : une opération sur un agrégat est composée d'une séquence d'opérations, une sur chaque composant.

Exemple : créer un employé revient à valoriser son numéro, son nom ...

b) la généralisation correspond à un choix : on a deux types de structure de contrôle : IF THEN ELSE et CASE.

Une opération sur un générique est composée d'une opération appliquée à la catégorie d'un générique.

Exemple : créer un employé de type secrétaire (opération sélective).

c) l'association correspond à une répétition : on a deux types de structure de contrôle : DO WHILE et FOR EACH.

Une opération sur un ensemble est appliquée sur chaque membre de l'ensemble.

Exemple : constituer un "syndicat" et éditer les appels aux cotisations.

Ces trois abstractions de contrôle peuvent être utilisées pour définir toutes les fonctions primitives récursives. Elles sont suffisantes pour le traitement de la plupart des applications.

iii) Modélisation des transactions.

La modélisation de la dynamique au niveau conceptuel comprend l'identification, la conception et la spécification des actions ceci pour chaque objet. On dispose de deux formes d'abstractions procédurales destinées à la modélisation des transactions sur la base de données en plus de celles déjà vues.

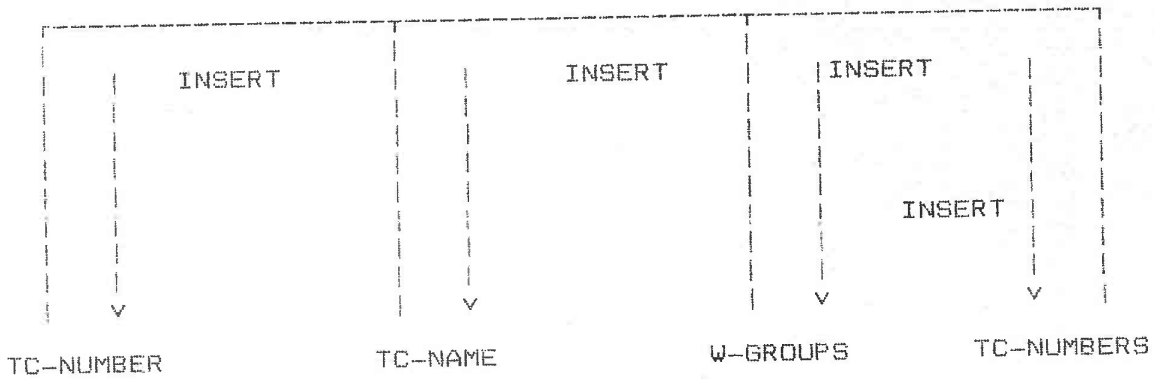
a) action : c'est une opération orientée application conçue pour un objet afin de s'assurer que toutes les propriétés de l'objet sont satisfaites. Avant d'effectuer l'opération sur la base de données certaines pré-conditions doivent être réalisées et des actions sur les objets peuvent alors être nécessaires.

Après la réalisation des actions une post-condition doit être vérifiée et l'opération sur la base de données est réalisée. La conception des pré et des post-conditions doit tenir compte des traitements d'exception.

La dynamique d'un objet est entièrement définie par ses actions qui seules peuvent modifier un objet. Cette forme d'abstraction permet de faire la différence entre l'objet considéré et les autres objets atteints uniquement par des actions : le domaine d'une action comprend l'objet directement concerné et tous les objets en relation immédiate avec lui par le biais de l'agrégation, la généralisation ou l'association.

Exemple : soit le schéma d'action "insérer" le comité technique.

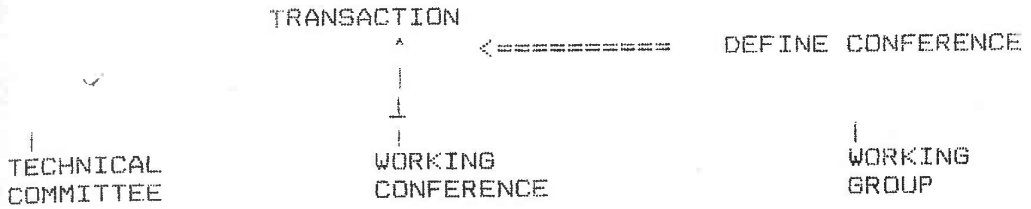
Conception du schéma d'action "insérer" : pour exécuter l'opération "insérer" le comité technique (TC), il faut insérer les constituants clés (tc-number et tc-name qui sont les objets de base) et les composants essentiels (tc-members et w-groups) qui sont les objets associés. Schématiquement :



b) Transaction : Les transactions sont les seuls moyens par lesquels les utilisateurs finals peuvent agir sur la base de données. Une transaction est une opération orientée application qui peut porter sur un ou plusieurs objets. La portée d'une transaction comprend tous les objets atteints durant la transaction. Ici aussi comme pour les actions, on a des contraintes relatives aux pré-conditions, aux post-conditions, au traitement des

exceptions qui spécifient les actions. Celles-ci induisent des opérations sur la base de données.

Exemple : schéma de transaction de définition du congrès : insérer un objet "conférence" avec les comités techniques et les groupes de travail impliqués dans l'organisation du congrès.



D) Les langages

Il existe deux sortes de langage :

i) un langage graphique complet qui permet la modélisation de la structure et de la dynamique. Ce langage est utilisé pour la réprésentation des schémas dynamiques. Ceux-ci intègrent les propriétés statiques et dynamiques en une seule représentation : la flèche (----->) lie l'objet sur lequel porte l'opération à l'objet dont est issue l'opération. Une flèche double (=====>) est utilisée pour distinguer les actions ou les transactions des opérations. La relation entre opérations est exprimée grâce aux différentes formes d'abstractions. (voir schéma page précédente). Ce langage graphique assure un lien de communication aisé avec les utilisateurs.

ii) le langage de spécification BETA qui permet la description des schémas d'objets et des transactions. Cette spécification doit répondre à un certain nombre de conditions :

. être abstraite, formelle, facile à utiliser et à modifier,

. le langage BETA est un langage de haut niveau répondant aux caractéristiques ci-dessus énumérées de la spécification,

. il est basé sur les techniques axiomatiques et prédicatives (non procédurales),

. il tient compte de la spécification pré/post des opérations sur les types abstraits.

exemple :

1) représentation graphique

ATTENDEE	PERSON	REFEREE
TC MEMBER		WG MEMBER
PC MEMBER		OC MEMBER
PC CHAIR PERSON		CFP INVITEE
CONFERENCE INVITEE		OC CHAIR PERSON
NATIONAL REPRESENTATIVE		SESSION CHAIR PERSON

2) description d'une personne en BETA

```
PERSON  =    OBJECT
        AGGREGATE OF
        PERSON - NUMBER;
        PERSON - NAME;
        ADRESS - ESSENTIAL;
        KEY PERSON - NAME;
        PRIMARY KEY PERSON - NUMBER;

     GENERIC OF
        national - représentatives;
        cfp-invitée : NONEXCLUSIVE
        conference-invitée :NON EXCLUSIVE
        tc-member :NON EXCLUSIVE
        wc-member :NON EXCLUSIVE
        referée :NON EXCLUSIVE
        oc-member :NON EXCLUSIVE
        pc-member :NON EXCLUSIVE
        pc-chair-person :NON EXCLUSIVE
        oc-chair-person :NON EXCLUSIVE
        session-chair-person :NON EXCLUSIVE
        attendee : NON EXCLUSIVE
END OBJECT;
```

Commentaire : pour représenter le schéma objet "PERSON" encore appelé générique nous avons les différentes catégories (attendee...session chairperson) concernées, une personne pouvant être représentée par plusieurs catégories. Une personne est un agrégat de son adresse, du numéro de la personne, de son nom, de la clé primaire numéro de personne.

3) définition d'une contrainte d'intégrité :

```
ALLp IN PERSON
(SOME a IN authorship (a.person number=p.person number)
(SOME ia IN INTENDED.author (p is ia) OR
(SOME cfpi:IN cfp. invitée (P IS cfpi) OR
(SOME r IN referée (p,s r)
(SOME tcm IN tc-member (P IS tcm) OR
(SOME wsm IN ws-member (P IS wsm) OR
(SOME ocm IN oc-member(P IS ocm) OR
(SOME occ IN oc-CHAIR PERSON(P IS occ) OR
(SOME pcc IN pc-CHAIR PERSON(P IS pcc) OR
(SOME SC IN SESSION-CHAIR PERSON (P IS SC))
```

Commentaire : dans les différentes catégories (REFEREE, TC MEMBER...) du schéma objet "PERSON", un individu peut jouer plusieurs rôles différents, la contrainte étant que le nom et l'adresse ne changent pas.

4) Définition d'une action : insérer un papier

```
ACTION  insert-paper (pn,title)
IN(a:authorship,title :paper-title,d:registration date)
Pn :(paper - number)
OUT (p:paper)
LOCAL (SP :submitted - papers)
PRE-CONDITION:
```

```
no-paper (pn ,title)?
submitted - papers-exist (pn)?
POST-CONDITION :authors-exist (pn)?
OB-OPERATION :INSERT paper (title,pn,d);
```

Commentaire : l'action "insérer un papier " montre que le numéro de papier et le titre sont reçus comme des paramètres tandis que les autres informations d'entrée (auteur, titre du papier, date d'enregistrement) sont obtenus grâce à un dialogue avec l'utilisateur.

Comme données locales à l'action "insérer un papier", on a les papiers déjà soumis.

Comme pré-condition il ne doit pas exister de papier dans la base avec le même titre et le même numéro ; comme post-condition il y'a au moins un objet auteur dans la base pour le papier inséré.

F / OUTILS.

Le papier présenté dans CRIS-1 ne parle pas de cet aspect. Dans CRIS-2, M-BRODIE signale qu'aucun logiciel automatisé n'est actuellement opérationnel.

E / ETAPES.

Dans la méthode ACM/PCM le cycle de conception se décompose en plusieurs étapes dont chacune traite un aspect de la conception des SI. Les étapes sont au nombre de six :

1 - formulation des besoins et analyse de ceux-ci. La connaissance du "monde réel" de l'application est informellement décrite et analysée,

2 - conception logique et spécification. Un modèle abstrait, précis de l'application est spécifié pour donner un modèle de données sémantique,

3 - conception de l'implémentation :

un modèle d'implémentation (schéma et programmes) est défini en conformité avec le système d'implémentation utilisé.

4 - implémentation :

le modèle d'implémentation est encodé, testé pour produire un ensemble exécutable.

5 - exécution, gestion et maintenance du logiciel,

6 - évolution, adaptation et modification :

Prendre en compte les besoins nouveaux et faire évoluer le logiciel.

Chacune de ces étapes est validée pour la compléter et la

consolider.

G / CONCLUSION.

Pour résumer : dans ACM/PCM la modélisation de la dynamique est faite en deux étapes :

1) les propriétés dynamiques au niveau global sont conçues, les actions et les transactions sont identifiées et reliées aux objets,

2) ces actions et transactions sont affinées et spécifiées en utilisant un langage approprié.

C'est donc une méthode très centrée sur l'activité de spécification.

Point forts :

- la conception est très intégrée procedure, un équilibre judicieux est réalisé en mettant également l'accent sur les objets,

- la modélisation intégre aussi bien les aspects statiques que les aspects dynamiques (l'expression des traitements est bien faite, ce qui n'est pas le cas des commandes dynamiques). Elle est précise, détaillée et basée sur les types abstraits,

- la modélisation s'effectue de façon modulaire,

- le langage graphique est très développé,

- le langage de spécification est compréhensible et complet.

Points faibles :

- le défaut d'environnement logiciel,

- le modèle abstrait encombré de beaucoup de considération d'implémentation,

- les concepts d'événement ou de trigger ne sont exprimés que par l'intermédiaire des graphes,

- une spécification obtenue encore bien éloignée de l'implémentation du SI.

LA METHODE DADES

A/ PRESENTATION.

La méthode DADES (DATA ORIENTED DESIGN) a été développée à la Faculté d'Informatique de l'Université Polytechnique de BARCELONE après quatre années de recherche par l'équipe d'ANTONI OLIVE . Elle met surtout l'accent sur les données qu'un SI collecte, stocke ou produit ; c'est une approche par les données beaucoup plus qu'une approche par les traitements. Les travaux de recherche sur lesquels se sont appuyés les concepteurs de DADES sont ceux de LANGEFORS [LANG 73] relatifs au concept de " précédence entre divers groupes d'information", ceux de VERRIJ [VERRI 1975] relatifs aux algèbres de l'information.

Cette méthode peut être utilisée en partie ou en totalité par d'autres méthodes. Elle propose un modèle dérivé du modèle relationnel intégrant les aspects statiques et dynamiques. La version presentée dans CRIS-1 en est encore au niveau de la recherche, il lui faut certaines adaptations et certains outils qui lui font aujourd'hui défaut.

OBJECTIFS.

Créer une méthode qui permette d'avoir :

- un langage formel pour la spécification des besoins du SI,

- une méthode pour valider formellement la consistance logique des specifications,

- une méthode pour vérifier formellement la logique des décisions de conception.

Les procédures de validation formelle s'appliquent aussi bien au traitement des aspects statiques qu'à celui des aspects dynamiques.

B/ NIVEAUX D'ABSTRACTION.

DADES traite du cycle d'abstraction et du cycle de contrôle. Ses propositions concernent plutôt le niveau conceptuel.

C/ CONCEPTS.

La lecture du document élaboré par CRIS-1 ne montre pas de façon explicite les concepts sous-jacents à la modélisation de la structure ou à celle de la dynamique.

C-1 MODELISATION DE LA STRUCTURE.

La modélisation de la structure est basée sur la description des classes d'éléments manipulés dans l'univers du discours, des

relations qu'entretiennent ces classes entre elles et des contraintes d'intégrité. Le résultat de cette modélisation est appelé schéma conceptuel. DADES préconise le modèle relationnel pour l'élaboration de ce schéma.

C-2 MODELISATION DE LA DYNAMIQUE.

Elle s'effectue grâce à la prise en compte des entrées, des sorties, du temps et surtout grâce à la dérivation.

C-2-1 La dérivation.

Elle exprime en fait la dynamique. Un schéma relationnel est considéré comme un schéma de base si ses tuples ne peuvent pas être dérivés des tuples d'autres relations de ce schéma. Dans le cas contraire ce schéma est dérivé.

On distingue deux classes de schémas relationnels dérivés :

a) celles qui sont dérivées par le biais de l'algèbre relationnelle : il doit y avoir une loi de dérivation pour chaque schéma relationnel inclus dans le schéma conceptuel. Cette loi de dérivation est simplement l'expression algébrique relationnelle correspondante,

b) celles qui ne sont pas dérivables par le biais de de l'algèbre relationnelle et qui sont le résultat d'opérations utilisateurs telles que les opérations arithméthiques, les modèles statistiques...Dans ce cas la loi de dérivation est plus complexe car il faudra spécifier les entrées, les sorties et la relation de production.

b-1 la relation de production.

Soit R le schéma de relation dérivé.

Soit K=K(R) et DRi la loi de dérivation de R ; on appellera relation de production PRi de DRi notée PRi = PRi(DRi) une relation sur K telle qu'il existe une correspondance entre chaque tuple de PRi et chaque tuple de R : en fait chaque tuple de PRi produit un tuple de R ; on peut également dire qu'il y a une occurrence de DRi pour chaque tuple de PRi.

b-2 relation de précédence.

Pour DADES l'analyse de dérivabilité pose le problème de la précédence c'est à dire le fait de savoir s'il existe une relation de précédence entre divers groupes d'information.

b-2-1 Le concept de précédence .

Soit A défini comme un groupe d'information avec ses composants (1... i) et B défini par (j... n), on dira qu'il y a précédence notée P[A,B] entre A et B s'il existe un traitement dans lequel les entrées sont les composants (j... n) de B et les sorties les composants (1... i) de A.

Grâce au concept de précédence pour savoir si un groupe d'information peut être dérivé à partir d'un autre, il suffit de

connaître ses groupes d'information précédents.

Il est possible à partir de lois de précédence données d'obtenir par déduction des lois de précédence additionnelles . Pour les groupes d'information A,B,C,D, on peut obtenir les lois de déduction suivantes :

- loi 1 : réflexivité : si A<B alors P[A,B],

- loi 2 : augmentation : si A<B et P[C,D] alors P[AUC,BUD],

- loi 3 : transitivité : si P[A,B],et P[B,C],alors P[A,C],

- loi 4 : pseudotransitivité : si P[A,B] et P[C,AUD] alors P[C,BUD].

- loi 5 : union : si P[A,B] et P[C,B] alors P[AUC,B],

- loi 6 : décomposition : si P[AUB,C] alors P[A,C] et P[B,C].

Les lois de dérivation étant des traitements élémentaires, on peut les regrouper pour obtenir d'autres traitements ceci grâce à des opérateurs de regroupement. On définit un opérateur de regroupement en indiquant ses opérandes, les conditions qu'il doit satisfaire et le traitement obtenu comme résultat.

Citons quelques uns de ces opérateurs :

- le regroupement horizontal P=HG(P1...Pn) : il regroupe plusieurs traitements en un traitement P dont les entrées et les sorties sont l'union des entrées et des sorties de P1..Pn,

- le regroupement vertical P=VG(P1,P2,B) : il permet de regrouper les traitements P1 et P2 en un traitement P qui élimine la lecture et la production de B, B regroupant l'information d'entrée de P1 et l'information de sortie de P2,

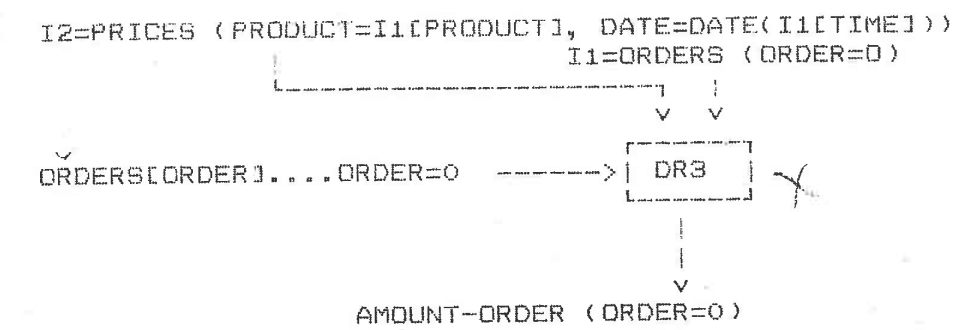
- la projection PJ=(P1,X) : cet opérateur agrége toutes les occurrences de P1 ayant la même valeur que les attributs de X, X étant un ensemble d'attributs tels que X<A (PR(P1)),

- la sélection P=S(P1,R) : cet opérateur permet la sélection de quelques occurrences de traitement P1. On aura en fait des occurrences de P qui constitueront des sous-ensembles des occurrences de P1.

D/ LANGAGES.

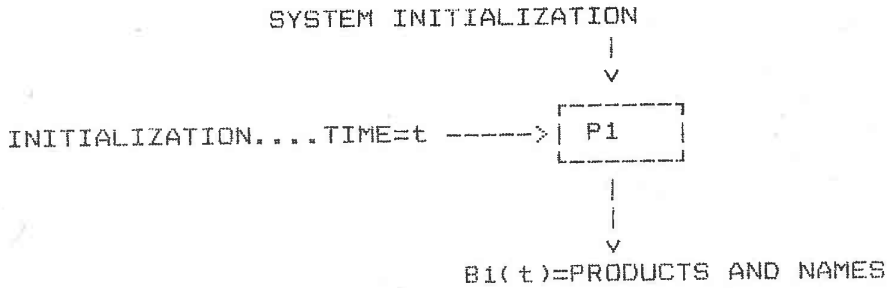
i) Un langage graphique de représentation des lois de dérivation, des modules constitutifs de l'architecture générale de traitement et pour la définition des entrées-sorties du SI.

Exemple 1:



Commentaire : pour des produits commandés à une date d, connaissant le prix on peut déduire le montant de la commande.

Exemple 2/.

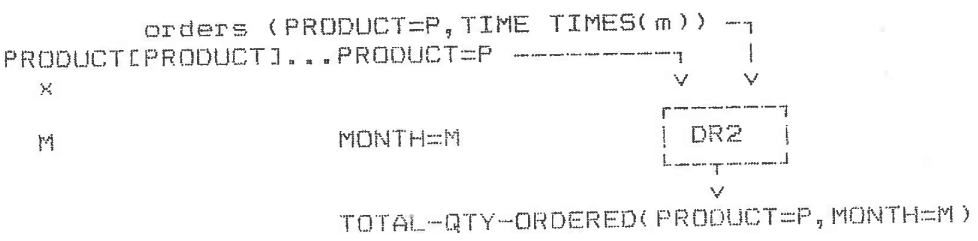


Commentaire : pour le module de traitement P1, l'initialisation du système à un temps t produit comme résultats B1(t).

ii) utilisation de la narration et du langage formel inspiré de YOUNG et KENT [YOUNG 1958] pour la spécification du schéma conceptuel.

Exemple : soit la loi de dérivation (DR2) ci-dessous à décrire :

```
   DR2=TOTAL QUANTITY ORDERED PER MONTH
PROD-REL:FOR EVERY PRODUCT AND MONTH;
         PR=PRODUCTS[PRODUCT] x MONTH;
PARAMETERS :PRODUCT=P,MONTH=M;
OUTPUT:TOTAL QUANTITY ORDERED
    O1=TOTAL-QTY-ORDERED(PRODUCT=P,MONTH=M);
```



E/ OUTILS.

DADES est une méthode essentiellement manuelle. Dans CRIS-1 aucun outil spécifique ne nous est proposé.

F/ ETAPES.

DADES ne couvre pas tout le cycle de conception du SI. Elle ne trait ni de l'analyse des besoins, ni de la conception détaillée ni de l'implémentation. Elle fait cependant appel à d'autres méthodes pour combler quelques unes de ses lacunes.

F-1 PHASE DE SPECIFICATION PRELIMINAIRE.

Etape 1 : définition des besoins : DADES utilise à ce propos la méthode SSA (STRUCTURED SYSTEM ANALYSIS) [GANE 79],

étape 2 : élaboration du schéma conceptuel abstrait basé sur le modèle relationnel,

étape 3 : définition des conventions : en accord avec les utilisateurs on attribue un code spécifique aux classes d'objets du réel perçu.

F-2 PHASE DE SPECIFICATION PROPREMENT DITE.

Etape 1 : élaboration plus fine du schéma conceptuel :

- en définissant les domaines des attributs et leur type,

- en analysant les attributs temps dont on détermine les ensembles de valeur, les points spécifiques et les intervalles de validité afin de dimensionner les informations et de prendre en compte les aspects dynamiques du système,

- en décomposant les schémas relationnels en schémas de relation auxquels on ajoute le temps et les valeurs des attributs.

Etape 2 : définition plus fine des besoins d'entrée et de sortie : chaque entrée ou sortie esr définie comme une expression relationnelle avec précision des moments d'arrivée des entrées et de production des résultats, précision du lieu et estimation de la fréquence,

Etape 3 : définir les lois de dérivation .
Classement des schémas de relation soit comme schémas de base soit comme schémas dérivés. Pour chaque relation dérivée, on doit avoir une loi de dérivation,

Etape 4 : validation des spécifications.

Pour valider les spécifications déja effectuées, DADES fournit une méthode de validation de la logique des spécifications dont la principale condition est que les sorties anticipées doivent être dérivables à partir des entrées. Cette validation s'effectue grâce à l'analyse de précédence et à l'approche de regroupement de

traitement.

Les erreurs rencontrées doivent être corrigées, ce qui peut entrainer une modification du schéma conceptuel ou de quelques unes des lois de dérivation, un ajout ou une suppression de certaines entrées ou sorties.

F-3/ PHASE DE CONCEPTION ARCHITECTURALE.

DADES propose une approche "bottom-up" pour la conception de l'architecture du SI.

Etape 1 : conception logique de la base de données.

La base de données obtenue regroupe des informations de base et des informations dérivées, elle tient compte aussi de la dimension temporelle de ces informations. La structure logique obtenue est spécifiée dans un langage de définition des données.

Etape 2 : conception de la structure du SI.

Conception la plus optimale possible avec regroupement de traitement, multiples itérations pour assurer la consistance logique de l'architecture.

G/ CONCLUSION.

En résumé DADES se présente comme une méthode orientée analyse des besoins avec un accent particulier mis sur les données au niveau des spécifications.

Ses points forts sont :

- les lois de dérivation qui permettent une certaine prise en compte de la dynamique du SI,

- les règles de validation des spécifications.

Ses points faibles sont :

- une absence d'outils : méthode trop manuelle,

- d'être d'une certaine façon un outil à incorporer à d'autres méthodes ; de ne pas être une véritable methode de conception de SI car elle ne traite ni de la conception détaillée ni de l'implémentation,

- elle ne couvre pas tout le cycle de conception du SI,

- le processus de déclenchement de la dérivation est mal explicité.

II-2-4 LA METHODE NIAM.

A/ PRESENTATION.

La méthode NIAM (NIJSSEN'S INFORMATION ANALYSIS METHOD) a été développée dans un but commercial chez CONTROL DATA aux PAYS-BAS à partir des travaux de NIJSSEN, connu plus particulièrement pour avoir introduit la notion d'univers du discours [NIJ 77].

NIAM est une méthode très orientée "analyse de l'information" où la conception des diagrammes de flux d'information joue un grand rôle.

NIAM propose en plus des concepts propres à la modélisation des données, un langage graphique, et intègre des notions abstraites. Elle est à l'heure actuelle en cours d'exploitation et d'évolution.

Elle est commercialisée sous le nom de IA par CONTROL DATA; elle est diffusée dans plusieurs A.G.L (par exemple GRAPHTALK de RANK XEROX).

OBJECTIFS :

Lors de l'étape d'analyse de l'information, première étape du développement d'un SI NIAM, propose des concepts pour décrire le modèle de fonction. Des langages et des outils sont également proposés.

B/ NIVEAU D'ABSTRACTION.

NIAM est une méthode utilisant un cycle d'abstraction spécifique et fait référence de façon explicite au cycle de décision et au cycle de contrôle de la vie du SI. NIAM ne se propose pas de traiter toutes les étapes de développement d'un SI, mais il traite de l'étape d'analyse de l'information "étape inévitable à prendre en considération lors du développement du SI", ceci indépendamment de toute contrainte technique de réalisation. Etant spécifiquement applicable au cycle d'abstraction en tant que méthode d'analyse de l'information, NIAM se situe de par ses concepts et ses objectifs à un haut niveau d'abstraction.

C/ LES CONCEPTS.

C-1 CADRE DE DEFINITION ET DESCRIPTION DE LA STRUCTURE.

Le cadre de définition des concepts est celui défini par G.M.NIJSSEN, cadre qui est à l'origine des concepts introduits par le rapport préliminaire du groupe ISO TC97/SC5/WG3 [ISO 81] intitulé "Concepts and Terminology for the Conceptual Schema", rapport qui préconise un cadre pour l'élaboration du schéma conceptuel.

i) Eléments de base.

Ce cadre se compose de trois éléments principaux :

- le système objet : c'est la part de la réalité observable pour laquelle on veut collecter de l'information et retrouver l'information éventuellement dérivée,

- le système d'information : ou outil qui délivre l'information nécessaire au fonctionnement du système objet,

- l'environnement, somme d'information pour le SI et le système objet.

Ce cadre introduit les concepts propres à la modélisation des aspects statiques du SI c'est à dire de sa structure. Ces trois éléments sont à l'origine du système d'abstraction et de la base d'information, ensemble ils contribuent à l'élaboration du schéma conceptuel.

ii) Système d'abstraction : modèle mental du système objet qui reflète les situations et comportements intervenant dans le système objet ; il consiste en des classes d'objets et d'activités d'une part, en des règles d'autre part.

iii) Base d'information : c'est la partie du SI qui reflète l'état du système objet. Les flux d'information y prennent leur origine et leur terme, la base d'information agit comme un moyen de stockage des messages.

Le concept de base d'information est complété à partir de la notion de modèle de phrase. La base d'information consiste en un ensemble de phrases élémentaires qui sont un cas particulier des phrases en langue naturelle telles que :

"PIETERS vit à AMERSFOOT".

Une telle phrase présente beaucoup de lacunes, elle n'a de sens que dans un contexte déterminé. Ce contexte est précisé par l'introduction de deux nouveaux concepts :

iv/ objets lexicaux : une phrase ne montre pas les objets concernés mais les identifie par des noms "PIETERS", "AMERSFOOT". Les objets lexicaux sont des chaines de caractères qui se réfèrent à un objet.

Les objets lexicaux sont regroupés en types d'objets lexicaux ou "LOT" (LEXICAL OBJECT TYPES).

Exemple : PIETERS est une occurrence du LOT surnom,
AMERSFOOT est une occurrence du LOT nom de ville.

Les LOT sont en fait les types propriétés habituels en conception de bases de données.

v/ objets non lexicaux : Ce sont des choses réelles ou abstraites dans le système objet qui sont regroupées en types d'objets non lexicaux ou NOLOT. Ce sont en fait les types entités habituellement rencontrés en conception de bases de données.

Exemple : si on prend le NOLOT PERSONNE, on peut avoir comme occurrence un lecteur donné d'un papier, ou l'auteur de ce papier.

Les occurrences de NOLOT sont par définition imprécises.

La phrase "PIETERS vit à AMERSFOOT " peut être maintenant transformée en ce modèle de phrase :

"La personne dont le nom est PIETERS vit dans la ville dont le nom est AMERSFOOT".

Ou en un autre modèle de phrase :

"La ville dont le nom est AMERSFOOT est la résidence de la personne dont le nom est PIETERS".

Les deux phrases indiquent la même association entre deux objets non lexicaux "personne" et "ville". Chacune des phrases est complétée par des prédicats "vit dans", "est la résidence de" que NIAM appelle des rôles.

On constate que chaque phrase est riche des différents types d'objets en association entre eux et des prédicats qui complètent cette phrase.

vi/ <u>Association :</u> cette structure fait apparaître :

- une association entre des objets non lexicaux. Une telle association sera appelée une idée ou une réalisation d'une idée type.

Exemple : "une personne vit en ville " traduit un type d'idée.

- une association entre un objet non lexical et un objet lexical. On appelle une telle association un "pont" ou une réalisation d'un "type pont".

Exemple : "une personne est référencée par son nom".
"Une ville est référencée par son nom".

NIAM décompose les types de phrases en associations binaires c'est à dire en "idées types" binaires et en "ponts type". Cette approcne est habituellement appelée pseudo-binaire.

vii/ <u>sous-typage.</u>

Le sous-typage dépend des propriétés partagées ou non (idée type et pont) par le type objet. Ce concept est directement dérivé de la notion de "généricité" de [SMI 77].

Exemple : soit le NOLOT "personne", il couvre les sous-types "membres du comité de programme " et "représentants nationaux".

Le SI va enregistrer des informations concernant ces deux sous - types.

NIAM applique cette opération de typage qui implique que chaque occurrence d'un (sous) type objet peut se réaliser dans les types idées ou ponts relatifs au (sous) type objet.

Pour compléter la description du système d'abstraction NIAM fait intervenir les contraintes d'intégrité. Celles-ci peuvent être décrites sous une forme graphique ou dans un langage de manipulation procédural.

NIAM distingue plusieurs types de contraintes :

a) contrainte de sous-ensemble : elle signifie qu'une population d'un ou de deux rôles (c'est à dire d'un type idée ou pont) est un sous-ensemble d'une population d'un ou de deux rôles (eux-aussi d'un type idée ou d'un type pont) qui provient des mêmes types objets,

Exemple : le présentateur d'un papier doit être l'un de ses auteurs;

b) contrainte d'égalité : elle signifie qu'une population ayant un rôle particulier est égale à une population d'un autre rôle pour le même type-objet.

Exemple : pour une conférence les dates (début et fin) doivent être enregistrées ou aucune ne doit être enregistrée;

c) contrainte d'exclusivité : elle signifie qu'une combinaison d'occurrences (toujours provenant de type idée et/ou type pont différents) identifie de façon unique un certain objet lexical.

Exemple : une contribution est uniquement identifiée par la personne délivrant la contribution et la conférence pour laquelle la contribution est effectuée;

d) contrainte disjointe : elle signifie que les populations respectives de deux sous-types s'excluent mutuellement. Les deux sous-types devraient appartenir à la même famille d'objet non lexical,

Exemple : les papiers acceptés et les papiers rejetés appartiennent tous à la famille des papiers présentés;

e) contrainte de rôle total : elle signifie qu'un objet appartenant à un type objet doit obligatoirement jouer un certain rôle

Exemple : une conférence a lieu obligatoirement durant une session. Ne sont pas enregistrées dans la base d'information les conférences pour lesquelles la session n'est pas spécifiée en premier;

f) contrainte de clé identifiante : elle est dédiée uniquement à un type idée ou à un type pont. Elle limite la population des rôles d'un type idée ou d'un type pont.

Exemple : le numéro de session identifie la session; tout congrès comprend au moins une session.

L'ensemble de ces contraintes constitue un outil très puissant d'expression.

C-2 MODELISATION DE LA DYNAMIQUE.

Les aspects dynamiques sont décrits à partir des concepts de base que sont les fonctions, les flux d'information qui vont constituer un modèle de fonction qui autorisera une décomposition fonctionnelle avec des sous-fonctions.

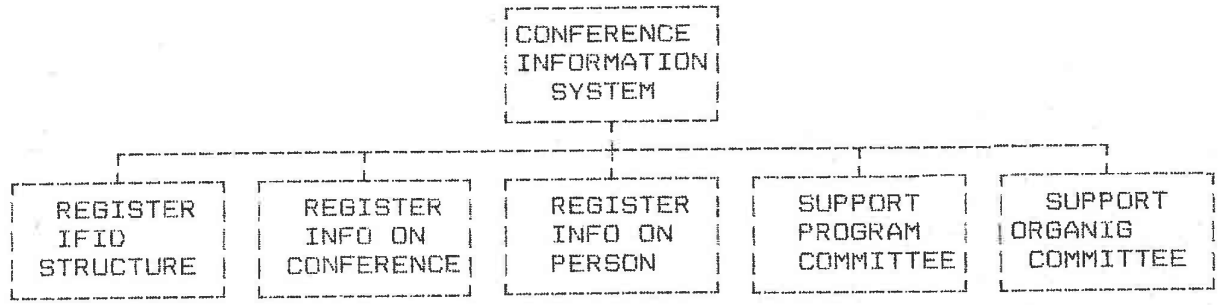
a) flux d'information : il véhicule un message représentant une communication entre deux partenaires. Tout flux d'information a une origine et une destination,

b) fonction : elle représente la capacité de transformer les flux d'information. Cette transformation est souvent caractérisée par un verbe.

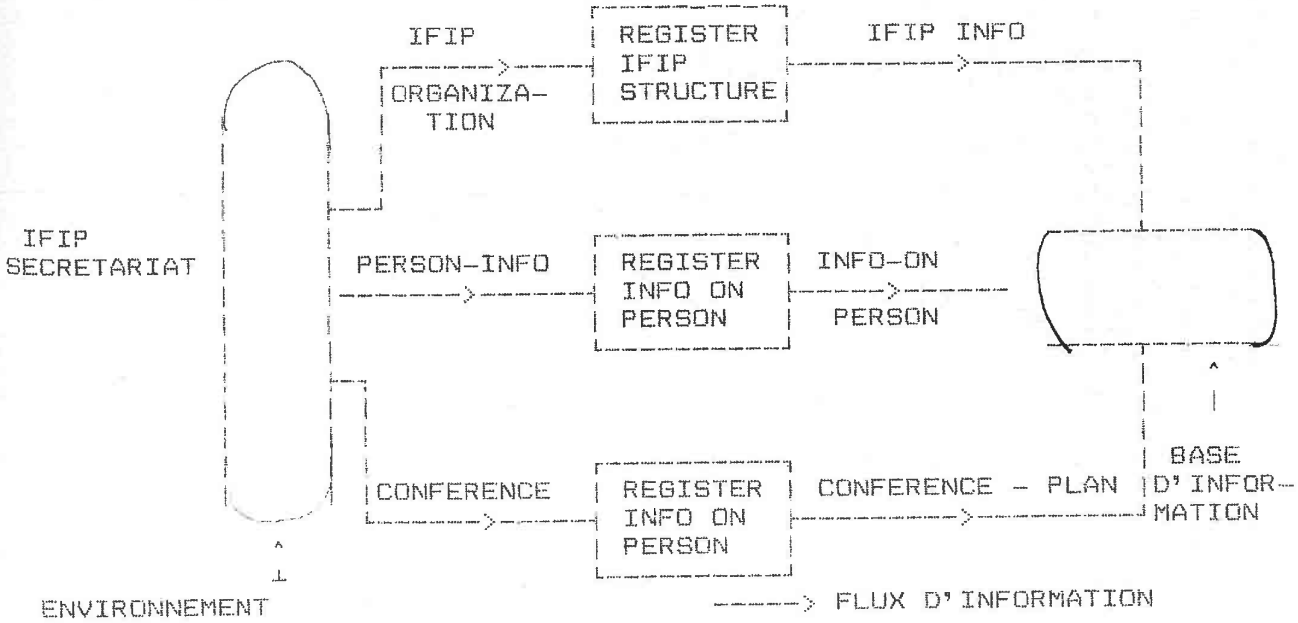
Une fonction n'est formalisable que lorsqu'on en connait le mécanisme de transformation. Seules les fonctions formalisables sont automatisables.

Une fonction complexe sera décomposée en sous-fonctions qui pourront être décomposées à leur tour jusqu'à ce que l'on puisse décrire le résultat de façon très détaillée et définir ses flux d'information.

Grâce à cette décomposition on établit un diagramme des flux d'information qui met en relation les fonctions, l'environnement et la base d'information.



Commentaire : une fois la décomposition effectuée, les flux d'information inventoriés, on peut établir un diagramme des flux d'information qui met en relation les fonctions, l'environnement et la base d'information.



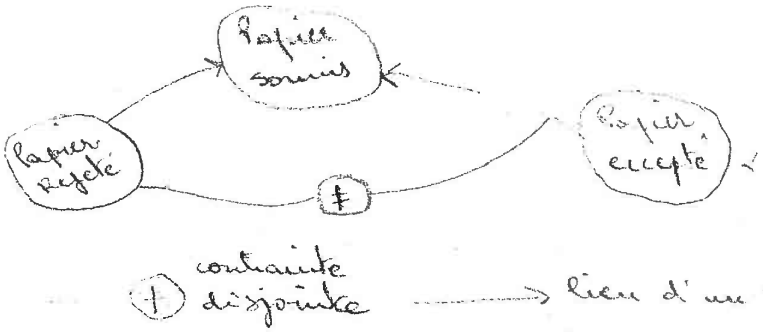
D/ LE LANGAGE.

NIAM propose deux sortes de langage :

1) Un langage graphique permettant une spécification des aspects

statiques et dynamiques du SI. Vu la simplicité du graphisme l'utilisateur peut participer à l'élaboration de la grammaire conceptuelle (ou schéma conceptuel) qui comprend les types de phrase, les objets lexicaux ou non, les sous-types et les contraintes,

Exemple :



Commentaire : un papier est soit rejeté ou accepté, dans les deux cas ce papier appartient à la famille des papiers présentés.

2) un langage formel RIDL (REFERENTIAL IDEA LANGUAGE) qui permet de spécifier de façon formelle le système d'abstraction. Ce langage permet :

- la spécification des IFD (diagrammes de flux d'information) et des ISD (diagrammes de structure d'information),

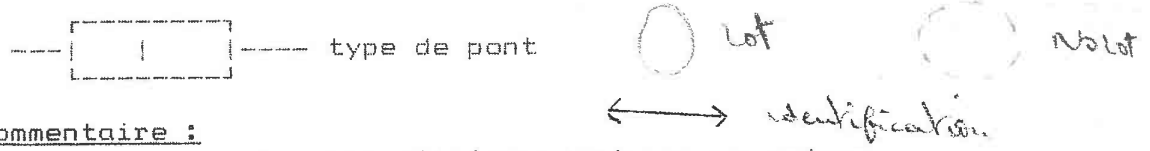
- la spécification des contraintes d'intégrité (CI) qui ne peuvent être représentées par une notation graphique. Ces types de contraintes sont appelées contraintes "procédurales",

- la spécification des transformations des flux d'information appelés "procedures".

Le formalisme de RIDL est basé sur des mots-clés prédéfinis pour les prédicats et procédural pour la description des fonctions.

Exemples :

- soit le pont suivant :



Commentaire :
une personne peut avoir plusieurs prénoms, un prénom peut être partagé par plusieurs personnes.

DESCRIPTION DU PONT PRECEDENT :

```
add bridge-type PERSON IDENTIFICATION
    roles (PERSON bearing and SURNAME of);
```

```
* description de types entites (NOLOT):
 ADD NOLOT PERSON,PAPER,CONFERENCE

* description de propriétés :
 ADD LOT PERSON-NR,PAPER-NR,SURNAME,TITLE

*description des contraintes :
 soit la contrainte disjointe vue précédemment a décrire :

add constraint ACCEPTED-OR-REJECTED
     condition
       ACCEPTED-PAPER IS disjoint of REJECTED-PAPER
     holds

*définition d'une procédure :
 procedure list-invitees
   in conf:CONFERENCE-NR
     begin
       invitees:=PERSON receiving INVITATION for CONFERENCE
                 identified-by CONF
       sort INVITEES or SURNAME of PERSON;
       for INVITEES do
           list (PERSON-NR of,INITIALS of,SURNAME of)
       end for
     end;
```

E/ LES OUTILS PROPOSES.

NIAM CONTROL/DATA propose de nombreux outils de contrôle de la cohérence et de documentation.

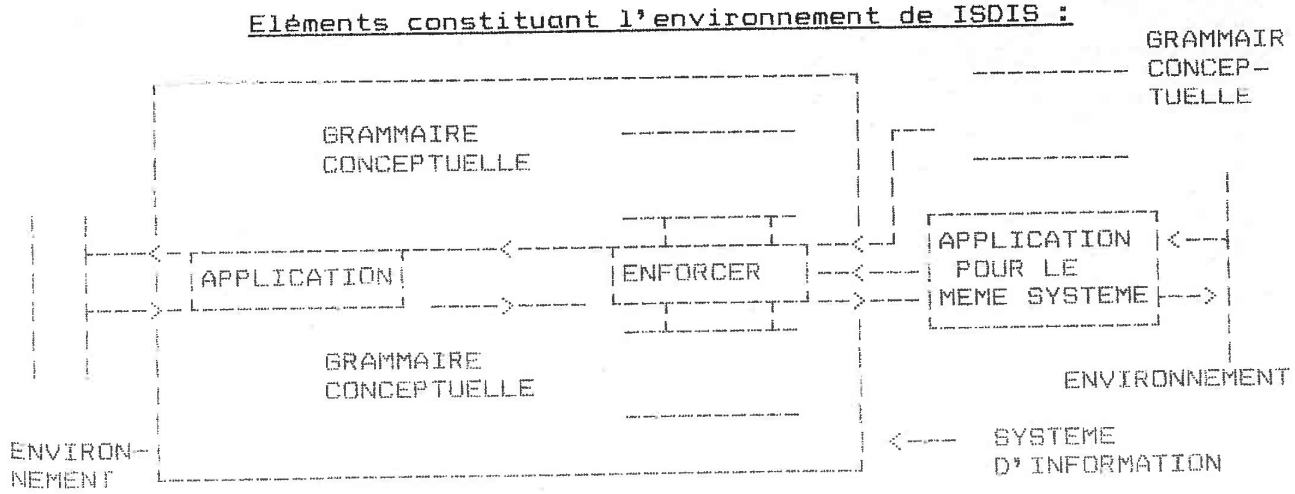
Le plus important de ces outils est ISDIS méta-système destiné à l'analyse de l'information. Il est constitué d'un dictionnaire d'information et d'un environnement d'outils.

- il stocke et met à jour la grammaire conceptuelle en utilisant les concepts de NIAM déjà vus,

- il montre les implications et les conséquences de la grammaire conceptuelle spécifiée,

- il compile la grammaire conceptuelle pour la rendre exploitable par l'outil appelé "ENFORCER".

Eléments constituant l'environnement de ISDIS :



L'ENFORCER tel que le montre le schéma ci-dessus s'appuie sur la grammaire conceptuelle et sur le SI. Il effectue les mises à jour du SI et les productions d'information qui en résultent. Il n'offre pas la possibilité de simuler ou de faire fonctionner la dynamique du SI à cause du manque de complétude de la description dynamique.

De plus NIAM est devenue maintenant un standard dans plusieurs A.G.L quoiqu'à un degré moindre par rapport à MERISE.

F/ ETAPES.

1) Analyse du système objet : afin d'en connaitre les entrées et les sorties et d'étudier l'opportunité de l'amélioration du SI, et de savoir en cas d'améliorations où et comment celles-ci se feront.

2) Analyse de l'information : elle consiste à :

- déterminer les fonctions du SI,

- décomposer ces fonctions : cette décomposition consiste en :

  . la construction des IFD,

  . l'analyse des flux d'information débouchant sur des ISD,

  . l'expression de tous les ISD grâce à une grammaire conceptuelle,

  . la formalisation des contraintes et des performances des différentes fonctions.

3) Implémentation, installation et exploitation.

L'examen de la grammaire conceptuelle permet de décider de la partie à implémenter (qui peut ne pas se faire par le biais de l'informatique), et de la partie à automatiser.

G/ CONCLUSION.

NIAM n'est pas une méthode de conception complète de SI mais peut être utilisé dans le cadre d'une autre méthode.

Ses points forts sont :

- une description très puissante de la structure du SI grâce à l'analyse de l'information,

- le langage utilisé pour décrire le système, ses nuances et ses caractéristiques,

- la notation graphique dans laquelle l'analyse de l'information est présentée. Cette notation, outre le fait qu'elle est utile au concepteur, constitue un très bon moyen de communication avec l'utilisateur.

Ses points faibles sont :

- manque de complétude de la description de la dynamique : pas de possibilité de simulation ni de validation,

- la complexité du schéma conceptuel obtenu dont la compréhension n'est pas évidente par des utilisateurs.

II-2-5 LA METHODE REMORA.

A/ PRESENTATION.

Cette méthode (du nom d'un poisson le "remora" considéré dans la légende comme le poisson "pilote" du requin) a été développée au CRIN de NANCY depuis 1974 , à PARIS depuis 1978 par l'équipe de C.ROLLAND. Comme son nom l'indique, sa vocation est de "piloter" le processus de conception/réalisation d'un SI c'est à dire de proposer en particulier un environnement logiciel d'aide à la conception de SI. Cette méthode est en cours de développement et d'exploitation dans l'industrie.

OBJECTIFS : Proposer :

- une conception intégrée de la structure des faits et du comportement du SI,

- une conception en deux étapes correspondant à deux niveaux d'abstraction et de spécification du SI : une conceptuelle et une logique,

- une aide à la conception (concepts, langages formels ou graphiques, règles méthodiques, outils à chaque étape),

- une conception orientée vers la production de SI actifs.

B/ NIVEAU D'ABSTRACTION.

REMORA ne couvre pas l'intégralité du cycle de conception du SI. Il identifie deux niveaux d'abstraction appelés respectivement le niveau conceptuel et le niveau logique. Cette méthode couvre principalement les cycles d'abstraction et de contrôle. Au niveau conceptuel REMORA s'intéresse à la représentation sémantique de la réalité organisationnelle, alors que le niveau logique traite de la définition d'une solution technique en tenant compte du schéma conceptuel, des caractéristiques techniques de l'environnement, et des impératifs organisationnels de traitement. A chacun de ces deux niveaux sont pris en compte à la fois les aspects statiques et dynamiques du SI dans l'optique d'une conception intégrée.

O. THIERY [THI 85] a proposé le langage LASSIF (Langage de Spécification de Système d'Information) pour exprimer les propriétés statiques et dynamiques ceci en faisant appel aux types abstraits.

On peut en analysant les concepts sous-jacents à REMORA placer cette méthode à un "haut niveau" de formalisme d'abstraction.

C/ CONCEPTS .

Le modèle conceptuel de REMORA repose sur une description causale de la dynamique d'un système.

Dans une perspective dynamique, trois catégories de phénomènes sont représentées, elles sont décrites selon leurs propriétés comme

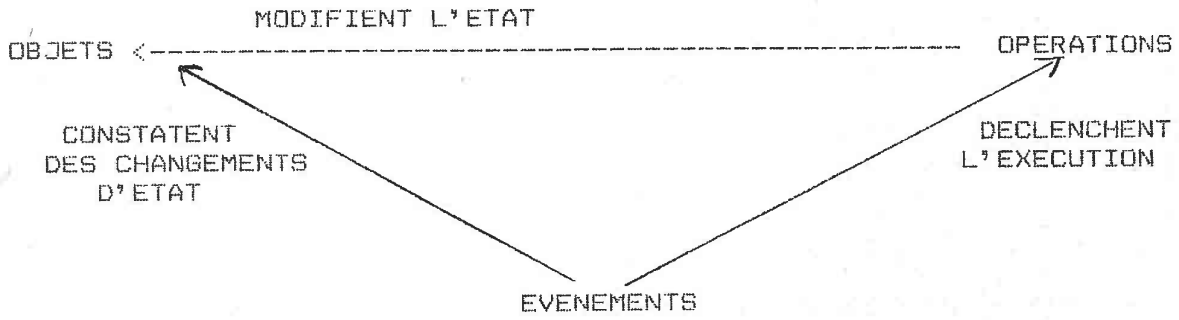
objets, événements et opérations.

Les objets définissent l'état du système. Ce sont des composants durables concrets ou abstraits que l'on peut particulariser [ROL 82]. Les changements d'état sont la conséquence de l'exécution d'actions appelées opérations déclenchées par des événements qui sont des stimilis internes ou externes qui peuvent survenir à n'importe quel moment.

Pour représenter complètement la dynamique, REMORA utilise trois catégories d'association entre les trois catégories de phénomène que nous venons d'évoquer : modifier (opération,objet), constater (objet, événement) et déclencher (événement, opération).

La description causale de la dynamique du système se traduit ainsi : les événements provoquent l'exécution d'opérations qui entrainent des changements d'état des objets ce qui peut aussi faire apparaitre d'autres événements.

Les relations inter-temporelles entre événements doivent être éga - lement décrites.



Les trois catégories de phénomènes (objets, événements, opérations) sont structurés en classes. Dans une classe, tous les phénomènes qui définissent cette classe sont décrits par la même collection de propriétés et ils jouent le même rôle dans l'organisation.

Ainsi , à des fins de représentation on fera correspondre à chaque classe un élément-type.

Exemples :

à la classe des objets employés, on fait correspondre le "t-objet" employé,

à la classe des associations d'affectations, on fait correspondre la "t-association" affectation,

à la classe des opérations de calcul du salaire, on fait correspondre la "t-opération" calcul de salaire,

REMORA propose un modèle normatif qui intègre le temps et permet la représentation des trois catégories de phénomènes évoquées précédemment. La réalité organisationnelle ainsi décrite reflète les relations qui s'établissent entre ces trois classes. La correspondan - ce entre la réalité et la représentation conceptuelle est telle que [ROL 87] :

1 - chaque classe de phénomènes est représentée par une ou plusieurs relations; il en est de même pour chaque classe d'associations,

2 - chaque propriété type d'une classe de phénomènes ou d'une classe d'associations est représentée par un attribut de relation,

3 - la catégorie de la classe de phénomènes ou de la classe d'associations est représentée par le type de relation correspondant (noté C-objet, C-opération, C-événement).

Les éléments de base du modèle conceptuel sont par conséquent les concepts de C-objet, C-opération, C-événement; ils permettent la repésentation normalisée des t-objets, t-opérations, t-événements, de leurs t-propriétés et t-associations.

En effectuant une correspondance entre l'analyse descriptive et la représentation conceptuelle :

- chaque t-objet ou chaque t-association d'objets sera représenté par un ou plusieurs C-objets,

- chaque t-opération et les t-associations où elle intervient seront représentées par une ou plusieurs C-opérations,

- chaque t-événement et les t-associations où il intervient seront repésentés par une ou plusieurs C-événements.

C-1/ LE CONCEPT DE C-OBJET.

Avant de définir le concept de C-objet, il nous faut définir auparavant la notion de dépendance fonctionnelle permanente entre deux attributs A et B. Cette dépendance est une dépendance élémentaire canonique où quel que soit a appartenant à A et b appartenant à B dépendant fonctionnelement de a, a et b ont la même durée de vie.

Dans une relation de type C-objet, le cas d'une relation de dépendance permanente entre chaque attribut et l'identifiant est une relation en 3FN (chaque attribut est en dépendance fonctionnelle permanente avec l'identifiant de la relation).

Exemple :

(1) NCOMPTE,DATASOLDE --> SOLDE
NCOMPTE : numéro de compte du client
DATASOLDE : date du dernier mouvement sur le compte
SOLDE : solde du compte

Commentaire : un compte n'a qu'un solde à un moment donné.

La relation fonctionnelle (1) est permanente.

- Définition :

"Soit AOB, l'ensemble des attributs représentant des t-propriétés de t-associations d'objets,

Soit F l'ensemble des dépendances fonctionnelles directes,

élémentaires, canoniques et permanentes existant entre les éléments de AOB,

Un sous-ensemble R de AOB est une relation de type C-objet si :

1. il existe I appartenant à R tel I soit clé de R,
2. quel que soit K appartenant à [R-I] I------> K existe dans F,
3. il n'existe pas L appartenant à [AOB-R] tel que I------> L existe dans F. " [ROL 87]

------> indique la dépendandance fonctionnelle.

1 et 2 correspondent à la définition de la troisième forme normale permanente,

3 correspond à une contrainte de maximalité des c-objets consistant à regrouper dans une même relation tous attributs en dépendance fonctionnelle permanente avec une même clé.

Exemple : soit le t-objet Produit défini par (numéro produit, nom du produit, couleur, poids, prix unitaire, numéro du catalogue où figure le produit sous un prix donné, date de changement de prix et de catalogue, quantité en stock, date de changement de valeur de la quantité en stock),

Ce t-objet pourra être représenté par les schémas de relation de type C-objets suivants :

PROD1 (numéro, nom, couleur, poids) qui décrit les propriétés constantes au cours du temps,

PROD2 (numéro, date de changement de prix, prix unitaire, numéro catalogue) qui décrit les propriétés variables (tarifs successifs),

PROD3 (numéro, date de changement de valeur, quantité) qui décrit l'évolution des stocks.

Un C-objet représente donc un aspect temporel de la classe puisqu'il identifie un ensemble de sous-schémas de classes.

On peut considérer un C-objet comme étant le plus grand ensemble de propriétés d'une classe d'objets ou d'associations ayant un comportement dynamique c'est à dire dont les propriétés sont créées, modifiées ou supprimées au même moment.

Il existe trois types de C-objets :

- C-objet permanent,
- C-objet variable,
- C-objet suppression.

Un C-objet représente un aspect atomique de l'univers d'application. La spécification d'un C-objet comporte [ROL 87] :

- le nom du C-objet ,
- le type du C-objet (permanent ou variable),
- le nom des attributs ,
- la liste des attributs avec pour chaque attribut :
  - . son nom,
  - . son domaine de valeurs,
- la clé du C-objet ,
- le nom de la classe à laquelle appartient le C-objet,
- une liste de contraintes d'intégrité associées.

Les différents aspects d'une classe sont représentés par la présence de plusieurs C-objets. On distingue deux types de schéma de relation C-objet :

- le premier type où les attributs représentant les propriétés permanentes de la classe d'objets ou d'association d'objets ne changent pas au cours du temps,

- le deuxième type où les attributs représentant les propriétés variables de la classe changent au cours du temps (voir exemple ci-dessus).

On appelle C-classe l'ensemble de ces C-objets.

C/2 LE CONCEPT DE C-OPERATION .

Le concept de C-opération est défini par rapport à celui de C-objet en respectant les deux contraintes de normalisation :

C-opération --> C-objet,

C-opération --> Texte - opération (TEXT-OP) .

Texte - opération désigne la règle associée à l'opération, les contraintes sont exprimées par des dépendances fonctionnelles :

- la première dépendance fonctionnelle entre une C-opération et un C-objet exprime qu'une C-opération modifie l'état d'un seul C-objet; il s'agit d'une contrainte structurelle,

"Soit OB l'ensemble des C-objets du SI,
soit OP l'ensemble des C-opérations du SI.
alors :

1. il existe une fonction MOD : OP ------> OB

2. quel que soit (opi,obi) : obi = MOD (opi) et opi appartenant à OP,
   il existe une fonction MODi : iopi ------> iobi
   iopi et iobi étant respectivement les clés de OPi et OBi.

1 sign 1 signifie que qu'une C-opération est liée à un seul C-o

2 signifie qu'une occurrence d'une C-opération ne modifie qu'une et une seule occurence d'un C-objet. "

- la deuxième dépendance fonctionnelle définit une règle de gestion de l'organisation qui doit obligatoirement être spécifiée dans la spécification de la C-opération; il s'agit d'une contrainte de complétude,

" Soit AOP l'ensemble des noms des attributs des C-opérations

soit T le sous-ensemble de AOP représentant des t-propriétés de type TEXTE.

soit IOP le sous-ensemble de AOP représentant les identifiants des C-opérations. Alors :

quel que soit iop appartenant à IOP, il existe un seul t appartenant à T, tel que iop ----→-- t.

Signification : tout C-opération a un énoncé de règle référencé par un attribut de type texte. "

Définition :

Etant donné que chaque C-objet représente un seul type de changement d'état élémentaire du futur SI, "il en résulte qu'une C-opération représente un type d'opération élémentaire du SI".

Une C-opération représente une action atomique (élémentaire et irréductible) de l'univers d'application. Elle représente un aspect temporel d'une classe d'opérations de la réalité organisationnelle; au cours du temps une C-opération modifiera un unique C-objet; une trace des exécutions successives pourra être conservée.

La représentation des C-opérations par des relations est effectuée à l'aide de relations permanentes en troisième forme normale.

Les relations C-opérations ont trois types de schémas :

- le premier type ROP où les attributs caractéristiques de la C-opération ne changent pas au cours du temps,

- le deuxième type RTEXT où l'attribut (REFTEXTE) désigne les valeurs des références de règles de gestion appliquées successivement par la C-opération,

- le troisième type REXEC où l'attribut IOB est l'identifiant du C-objet dont les occurrences sont modifiées par l'exécution des opérations.

Exemple : "Soit la C-opération - mise à jour de stock - correspondant à une t-opération définie par les t-propriétés :

MODSTOCK : nom de la C-opération,

DATOP : date de création de la C-opération dans

l'organisation,

TSTOCK : nom du texte,

DT : date de changement du texte de la C-opération,

DEXEC : date d'éxécution des opérations,

et par la t-association modifie avec le C-objet PROD3 (voir exemple sur C-objet) de clé IPROD3,

Cette opération peut être représentée par les 3 relations de type c-opérations de schémas suivants :

MODSTOCK1 (modstock, prod 3, datop)

MODSTOCK2 (modstock, dt, tstock),

MODSTOCK3 (modstock, dexec, iprod3). "

MODSTOCK1 est du type ROP, MODSTOCK2 du type RTEXT et MODSTOCK3 du type REXEC.

La spécification d'une C-opération comporte :

- le nom de la C-opération,

- le nom du C-objet auquel elle s'applique,

- la référence du texte associé et le texte lui-même,

- éventuellement les schémas de relation de type ROP, RTEXT et REXEC et leur définition.

L'ensemble des C-opérations forme la classe d'opérations.

C/3 LE CONCEPT DE C-EVENEMENT .

Il est défini par rapport aux concepts de C-objet et de C-opération en respectant les trois contraintes :

1) C-EV --> C-OB ,

2) C-EV --> (P-INIT,P-FIN) ,

3) C-EV --> C-OP .

P-INIT et P-FIN sont des désignations de deux prédicats définissant la pré-condition et la post-condition associées à un C-événement.

La première contrainte signifie qu'un C-événement constate le changement d'état particulier d'un seul et unique C-objet, et au moins une C-opération qu'il déclenche (il s'agit d'une contrainte structurelle),

"Soit OB l'ensemble des C-objets du SI,

soit OP l'ensemble des C-opérations du SI,

soit EV l'ensemble des C-événements du SI,

soit P(OP) l'ensemble des parties de OP.

1. il existe une fonction CONST : EV -->> OB.

2. quel que soit (ev,ob) : ob = CONST (ev).

   si iob est la clé de ob et iev la clé de ev, alors il existe une fonction CONST : iob --/-- iev.

3. il existe une fonction DECL : EV --/-- P(OP)

4. quel que soit (ev,op) : op appartient à DECL (ev) si iev est la clé de op, alors il existe une fonction DECL : iev --/-- P(iop).

1. signifie qu'un C-événement est lié à un seul C-objet,

2. signifie qu'une occurrence de C-événement constate le changement d'état d'une seule occurrence de C-objet,

3. signifie qu'un C-événement est lié à au moins une C-opération,

4. signifie qu'une occurrence de C-événement déclenche au moins une occurrence d'une même C-opération. Elle peut en déclencher plusieurs. "

La deuxième contrainte spécifie l'état antérieur du C-objet et l'état de ce même C-objet après le changement d'état,

La troisième contrainte indique qu'un C-événement déclenche l'éxécution d'opérations correspondant à une ou plusieurs C-opérations différentes (contrainte de complétude).

"Soient respectivement OB, EV, OP les ensembles de C-objets, C-événements et C-opérations d'un SI,

soient les fonctions CONST et DECL, définies précédemment, liant les éléments de OB, EV et OP,

soit B l'ensemble des fonctions booléennes de type ob --/-- [vrai,faux] où ob appartient à OB,

soit F l'ensemble des fonctions de type obi -->> P(obj) où obi et obj appartiennent à OB.

1. quel que soit (ev,ob) : ob = CONST (ev), il existe pred appartenant à B : (pred : ob ---- [vrai,faux]).

2. quel que soit (ev,op,ob) : op appartient à DECL (ev) et ob = CONST(ev),

- il existe cond appartenant à B :(cond : ob -->> [vrai,faux]),
- il existe fac appartenant à F : (fac : ob --/-- P(obj)) avec obj appartenant à OB.

Signification :

1. à tout événement est associé une fonction booléenne qui restitue la valeur "vrai" lorsque le changement d'état est un événement et "faux" dans le cas contraire.

2. Le déclenchement d'une C-opération par un C-événement peut être conditionnel (déclenchement selon que la condition est vraie ou non) ou itératif (exécution de l'opération spécifiée autant de fois que l'indique le facteur de déclenchement).

Pour résumer on peut dire qu'un "C-événement représente un "type de changement d'état particulier et élémentaire du SI qui provoque un type de transition d'état élémentaire par l'exécution d'un ensemble d'opérations élémentaires".

La représentation des C-événements par des relations est effectuée par des relations permanentes en 3FN.

La spécification d'un C-événement comporte :

- le nom du C-événement,

- le nom du C-objet constaté,

- la référence du prédicat associé et l'énoncé du prédicat,

- la liste des C-opérations déclenchées avec, pour chacune d'elles si nécessaire :

  . la référence de la condition de déclenchement et l'énoncé de la fonction booléenne associée,

  . la référence du facteur de déclenchement et l'énoncé de la fonction correspondante.

- la définition des domaines des attributs.

Un C-événement peut être décrit comme une relation qui permet la représentation d'un aspect temporel d'une classe d'événements de l'univers d'application et plusieurs relations permettent de décrire tous les aspects de la classe.

Exemple : "Soit le c-événement "rupture de stock" correspondant à un t-événement défini par :

- les t-propriétés :

RUPT : nom du C-événement
DRUPT : date d'introduction du C-événement dans le SI,
DARRUPT : date d'arrivée d'une rupture de stock,

- la t-association "constate" avec le C-objet PROD3 de clé IPROD3 (voir exemple sur C-objet) de t-propriétés :

PRUPT : prédicat définissant la condition de rupture de stock,
OPRUPT : date de modification de PRUPT,

- la t-association "déclenche" avec la c-opération REAPRO de clé IREAPRO (réapprovisionnement) de t-propriétés :

CONDREAPRO : condition de réapprovisionnement,
FACREAPRO : facteur de réapprovisionnement,
DCONDFAC : date de modification du facteur et de la condition,
DREAPRO : date de déclenchement d'un réapprovisionnement.

Ce C-événement peut être représenté par 5 relations de type C-événement de schémas suivants :

RUPT1 (rupt, prod3, réapro, drupt)
RUPT2 (rupt, drupt, prupt),
RUPT3 (rupt, réapro, dcondfac, condreapro, facreapro),
RUPT4 (rupt, darrupt, iprod3),
RUPT5 (irupt, ireapro, drapro)
irupt est synonyme de rupt, darrupt ".

RUPT1 décrit les propriétés permanentes,
RUPT2 donne les références des prédicats,
RUPT3 définit les références des conditions et des facteurs de déclenchement de l'opération de réapprovisionnement utilisés au cours du temps,
RUPT4 définit l'ensemble des ruptures de stock survenues au cours du temps,
RUPT5 définit pour chaque rupture de stock l'action de réapprovisionnement entreprise.

RUPT2, RUPT3, RUPT4, RUPT5 gardent une trace des actions effectuées ou des événements qui ont eu lieu.

LASSIF [THI 85] a introduit pour compléter le modèle REMORA quatre types qui permettent la spécification du SI ; ces quatre types correspondant aux trois classes et aux trois types d'association sont appelés constructeurs ; ce sont :

- CLASSE-OB (expression d'une classe d'objets du monde réel),

- COPERATION (expression d'une classe d'opérations),

- CEVENT (expression d'une classe d'événements et de ses conséquences),

- SI (expression de la composition d'un SI particulier).

Cette formalisation par rapport à la modélisation dans REMORA permet une simplification de la présentation du modèle, complète l'expression de ce dernier et favorise une spécification intégrée modèle/outil.

D - LES LANGAGES .

REMORA propose deux types de langages :

- un langage graphique permettant la description de la statique et de la dynamique d'un SI.

Conventions de notation :

représente un C-objet

-------> représente une C-opération

représente un C-événement

--------->> représente un déclenchement itératif d'une C-opération

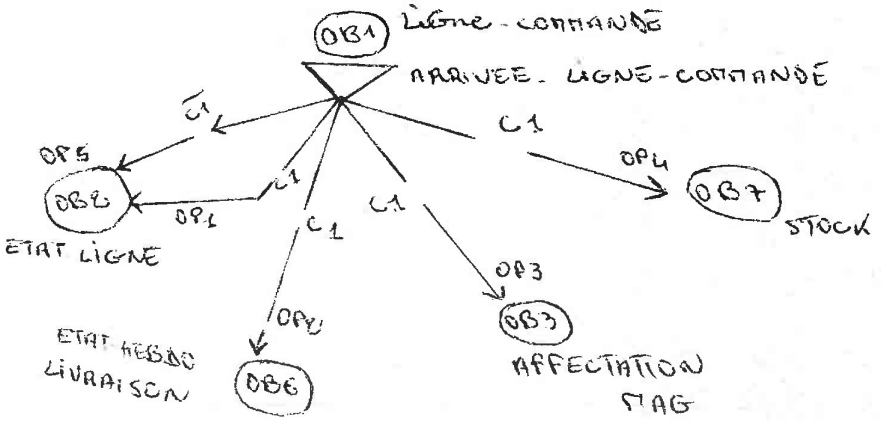
---------> représente un déclenchement conditionnel d'une C-opération.

représente la relation "constater" entre un C-objet et un C-événement

représente la relation "déclencher" entre un C-événement et des C-opérations.

représente la relation "modifier" entre une C-opération et un C-objet.

Exemple :



Commentaire : l'arrivée d'une ligne de commande est un événement de type EV1 qui déclenche :

. s'il existe un stock de l'article commandé suffisant (C1)

- l'affectation des quantités disponibles en stock à la ligne de commande (OP3),
- la mise à jour correspondante de stock (OP4),
- la préparation d'un état récapitulatif des lignes à livrer dans la même semaine (OP2),

. si le stock de l'article commandé est insuffisant (C1), la ligne de commande est mise à l'état "attente de fabrication" (OP5).

- un langage déclaratif à norme SQL [ROL 87] permettant d'exprimer la manipulation des données et qui est associé à un langage d'expression des traitements type PASCAL (Remo-langue).

E / LES OUTILS.

REMORA comprend :

- un système de conception assistée par ordinateur destiné à la conception du SI et à sa maintenance. Un automate appelé "pilote" le contrôle, en plus de cette dernière fonction il coordonne les interventions des hommes et des outils qui sont amenés à prendre un certain nombre de décisions et à exécuter des actions appropriées.

Ce système assume quatre fonctions principales :

* la fonction de contrôle de la consistance et de la fidélité des divers modules constitutifs du schéma conceptuel,

* la fonction d'intégration des divers modules,

* la fonction de documentation du concepteur et de l'automate,

* la fonction de simulation.

- un système de gestion de SI (SGSI) [THI 85] dont un outil particulier est le processeur d'événements. Son rôle est de reconnaitre les divers états du SI pendant la phase de conception et d'implémentation et de déterminer ainsi les actions à exécuter pour le bon déroulement du processus de conception.

L'ensemble des outils du SGSI permet de contrôler automatiquement la dynamique du SI. Jusqu'à présent ces outils sont restés à l'état de prototypes.

F/ ETAPES :

REMORA comprend deux grandes étapes : l'étape conceptuelle et l'étape logique.

1) **l'étape conceptuelle** comprend quatre sous-étapes interconnectées :

- l'analyse et la description du domaine d'application : phase d'analyse des besoins,

- la représentation formelle par un schéma conceptuel,

- la spécification de ce schéma dans ses détails,

- la validation : contrôles formels (cohérence, non redondance, complétude de la spécification), contrôles informels par utilisateurs .

2) **l'étape logique :** prise en compte des paramètres techniques et organisationnels pour la définition de la structure de la base de données et du découpage des transactions.

Au niveau logique REMORA présente un modèle logique standard indépendant de tout SGBD. Ce modèle permet de déterminer la structure logique optimale des données quel que soit le domaine d'application. La démarche au niveau logique aussi bien pour la structure des données que pour les transactions est de dériver la solution logique de la solution conceptuelle obtenue par l'application des règles

qui tiennent compte aussi bien des besoins des utilisateurs que des impératifs de fonctionnement de l'environnement technique et d'exploitation. Il s'agit là de règles formelles ou de règles validées par l'expérience des concepteurs.

Le résultat de cette étape se traduit comme pour l'étape conceptuelle par un schéma appelé schéma logique qui se décompose en deux sous-schémas complémentaires intimement liés :

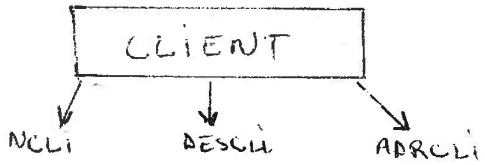
- le sous-schéma logique des données,

- le sous-schéma transactionnel.

2-1 . Le SOUS-SCHEMA LOGIQUE DES DONNEES.

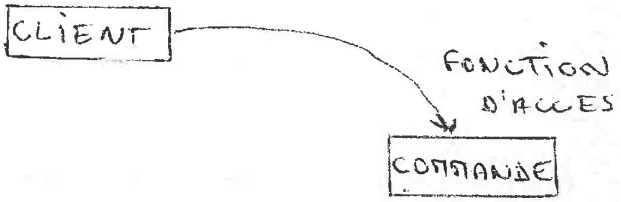
Il peut consister en :

- un regroupement des propriétés définissant un objet représenté ainsi :



Commmentaire : on définit ainsi un fichier logique CLIENT dont chaque article donne accès au numéro client (NCLI), à sa désignation (DESCLI) et à son adresse (ADRCLI).

- un choix d'une fonction d'accès qui permet par exemple pour un client donné d'accéder aux données correspondant aux diverses commandes passées par un client ce que l'on représente par :



On peut voir le schéma logique des données comme un graphe avec des noeuds représentant les fichiers logiques et des arcs indiquant les chemins d'accès entre ces fichiers logiques.

Le sous-schéma logique des données tient donc compte :

- du sous-schéma conceptuel statique,

- des besoins exprimés sous forme de fonction d'accès.

Pour assurer l'indépendance de la modélisation du schéma logique par rapport aux divers SGBD, REMORA procède de façon méthodique :

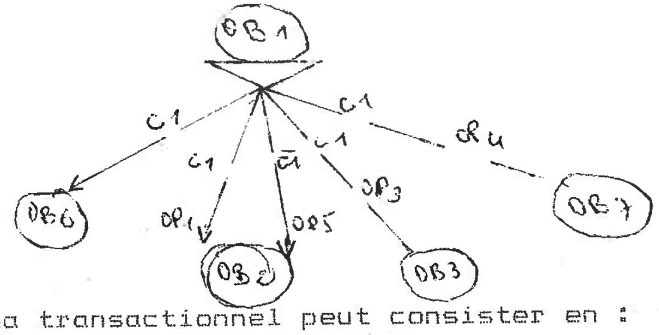
- 1) construction d'un schéma logique standard dont le modèle n'est supporté par aucun SGBD,

- 2) construction d'un schéma logique spécifique (spécifique car il tient compte des contraintes de structuration du modèle logique attachées au SGBD utilisé) à partir du modèle logique

standard.

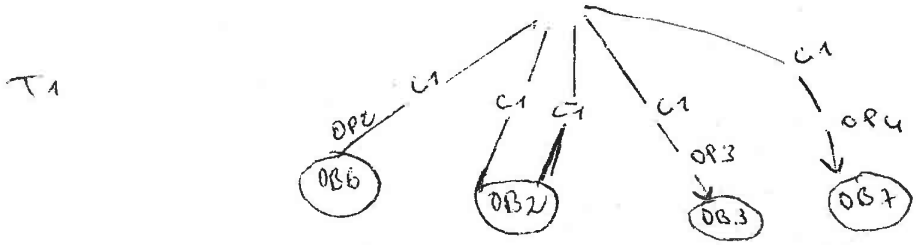
2-2 LE SOUS-SCHEMA TRANSACTIONNEL :

Etant donné un C-événement et les C-opérations qu'il déclenche

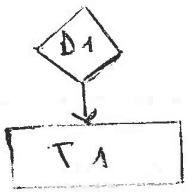


définir un sous-schéma transactionnel peut consister en :

- un regroupement dans une même transaction logique de plusieurs opérations munies de leurs conditions de déclenchement



- un choix de déclenchement manuel de la transaction Ti : un déclenchement Di (c'est à dire les préconditions à vérifier avant le déclenchement) est associé à la transaction Ti :



la transaction T1 est déclenchée chaque fois qu'il y'a une nouvelle ligne de commande.

- la prise en compte du déclenchement simultané de n transactions : exemple : n transactions Ti seront déclenchées à l'arrivée de n lignes de commandes.

Le sous-schéma transactionnel peut se définir comme un graphe de synchronisation des transactions où les noeuds sont les transactions exécutables à un instant donné et où les déclencheurs sont reliés aux transactions grâce à des arcs.

Pour l'élaboration du sous-schéma transactionnel le concepteur doit tenir compte :

- des besoins de renseignement des utilisateurs : besoins manifestés sous forme d'accès,

- de l'importance de trouver une solution optimale du point de vue accès aux données et sécurité des traitements,

- des diverses possibilités et contraintes des SGBD disponibles.

REMORA propose de procéder en deux étapes pour la production du sous-schéma transactionnel :

1) l'interprétation d'un sous-schéma conceptuel dynamique (prise en compte des transactions élémentaires et des conditions de leur déclenchement); on définit ainsi un schéma logique transactionnel qualifié de minimal,

2) la dérivation du sous-schéma logique définitif : à partir de l'interprétation précédente tenir compte des contraintes d'environnement technique et des contraintes utilisateurs pour élaborer la solution définitive.

6/ CONCLUSION .

REMORA est une méthode très orientée conception et spécification.

Ses points forts sont :

- une conceptualisation élaborée des aspects statiques et dynamiques du SI (ensemble cohérent de concepts),

- ses propositions de passage du niveau conceptuel au niveau logique,

- ses outils originaux en particulier le processeur d'événements et les procédures de contrôle qu'il permet.

Ses points faibles sont :

- des outils à l'état de prototypes,

- une absence de couverture du cycle de conception (gestion et contrôle du cycle de vie en particulier limités au processus de conception),

- un formalisme d'abstraction au niveau conceptuel difficile à comprendre car conduisant à de nombreuses relations. La couche Types Abstraits [THI 85] a permis de clarifier le formalisme (bien que n'introduisant aucun nouveau concept) car le mécanisme de hiérarchie de types permet des regroupements "en fait" de relations.

II-2-6 IDA : INTERACTIVE DESIGN APPROACH.

A / PRESENTATION .

La méthode IDA a été développée à l'Institut d'Informatique des Facultés Universitaires de NAMUR (Belgique) par F. BODART et Y. PIGNEUR en coopération avec le projet ISDOS développé depuis 1969 à l'Université de MICHIGAN sous la direction du professeur TEICHROEW.

Cette méthode s'appuie sur une structuration en parallèle des traitements et des données, elle met en oeuvre des modèles, une démarche et des outils logiciels qui supportent cette démarche. Dans son état actuel la méthode propose des modèles et des outils pour la maîtrise des étapes d'étude de l'opportunité et d'analyse conceptuelle du SI. Elle est commercialisée par la société METSI et complétée par l'atelier DELTA.

OBJECTIFS :

Proposer des moyens qui utilisent la modélisation en vue de développer des SI complexes, fiables, maintenables, adaptatifs de grande ou de moyenne envergure tournés vers les transactions sur les bases de données interactives. L'objetif assigné à IDA est de contribuer à la meilleure réalisation des objectifs assignables à un schéma conceptuel (complétude, cohérence, conformité aux besoins...).

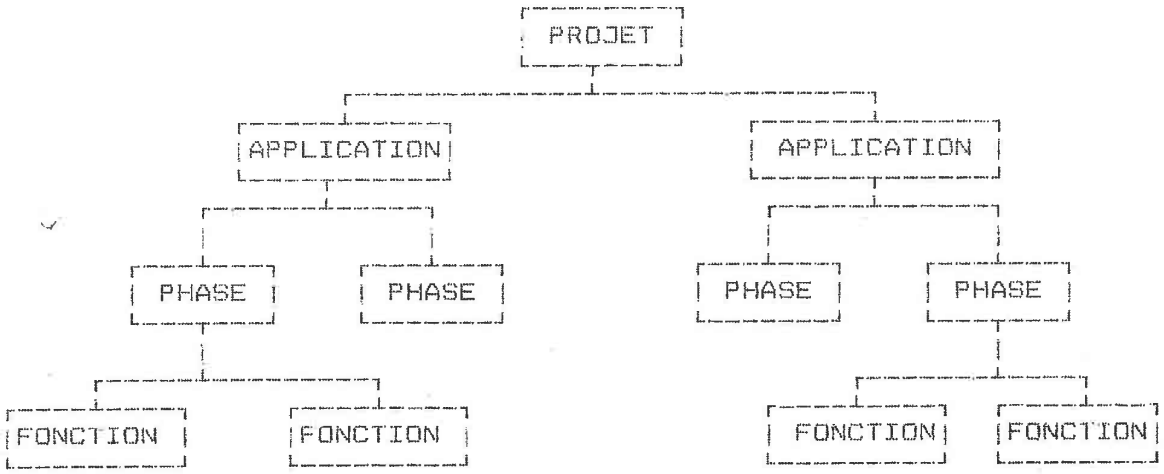
B / NIVEAU D'ABSTRACTION.

IDA couvre les trois cycles définis par BODART et AL à savoir les cycles d'abstraction, de contrôle, de décision. IDA s'appuie sur le principe de l'abstraction pour traiter les aspects statiques et dynamiques du SI, et la gestion de l'évolution de ce dernier. IDA propose une architecture à 3 niveaux de représentation (conceptuel, logique, physique). La méthode IDA peut donc être située à "un haut niveau d'abstraction".

C / CONCEPTS.

C-1 / Structuration statique des traitements.

IDA propose un ensemble de "concepts" permettant aux concepteurs de décomposer un projet en traitements plus élémentaires en procédant par raffinements successifs. Il s'agit plus précisement d'une proposition de modèle qui devrait fournir des critères d'identification de ces traitements en fonction d'une nomenclature standardisée. Les repères de cette nomenclature sont représentés ci-dessous.



i) Projet : C'est la partie du SI qui fait l'objet d'une analyse. Il n'y a à proprement pas de critères d'identification d'un projet qui est en fait lié aux objectifs opérationnels et de gestion d'un domaine déterminé de l'organisation.

Exemple : gestion de la documentation dans une Université.

ii) application : c'est un traitement quasi automatique par rapport aux autres applications d'un projet. C'est la dimension minimale d'un projet informatique. Comme critères d'identification on peut citer la faible interaction avec d'autres applications, l'intense circulation à l'intérieur de cette application.

iii) Phase : c'est un traitement (manuel ou automatisable) possédant une unité spatio-temporelle d'exécution. Ce qui implique que la phase soit exécutée dans une cellule d'activités ou centre homogène d'activité dans le temps et dans l'espace.

Exemple : dans la gestion des prêts on peut distinguer la phase de saisie des prêts, la phase de saisie des retours des livres, les différentes phases d'édition...

iv) fonction : elle correspond au niveau élémentaire des traitements, elle résulte de la décomposition d'une phase en sous-problèmes et spécifie les règles de traitement (procédures) relatives à la production de messages et aux différentes actions.

Exemple : dans la phase saisie des prêts on peut distinguer la fonction identification de l'emprunteur, la fonction enregistremenr des livres empruntés....

C-2/ Modèle de la dynamique des traitements.

L'objectif de ce modèle est de simuler le comportement du SI. Pour cela des concepts sont élaborés pour représenter l'enchainement des traitements et les conditions de leur déclenchement. Ce modèle de la dynamique repose sur deux concepts de base : l'évènement et le processus.

C-2-1 : concepts de base .

i) processus : c'est l'exécution d'une procédure de traitement de l'information dont la progression peut être représentée, à des points dans le temps, par son état. Trois états différents caractérisent le cycle de vie d'un processus :

a) état déclenché : le processus existe, il est en attente d'exécution par manque de ressources,

b) état actif : le processus est en cours d'exécution,

c) état terminé : le processus atteint son état final et cesse d'exister.

Exemple : la réception d'un bon de commande déclenche la création du processus "ENREGISTREMENT-COMMANDE". Tant qu'un employé n'est pas libre pour lancer le processus, ce dernier est en attente, dès que l'employé le lance il devient actif, dès que le processus est exécuté il cesse d'exister.

ii) événement : il correspond à un changement d'état du SI localisé dans le temps et dans l'espace. Un évènement est dit externe s'il provient de l'environnement externe du SI, il est dit interne s'il correspond à un changement d'état interne du SI.

Exemple d'événement externe : la reception d'un bon de commande.

C-2-2 Structures dynamiques élémentaires.

Le modèle de la dynamique s'appuie sur les concepts de base que nous venons de voir et sur des mécanismes particuliers.

C-2-2-1 Mécanismes du modèle :

- synchronisation : elle consiste à coordonner des événements afin de préparer le déclenchement d'un processus. La condition de synchronisation est la coordination nécessaire à réaliser pour le déclenchement des actions dynamiques,

- condition et multiplicité de déclenchement : la condition stipule le caractère optionnel du déclenchement du processus alors que la multiplicité de déclenchements stipule le nombre de processus d'un même type à déclencher.

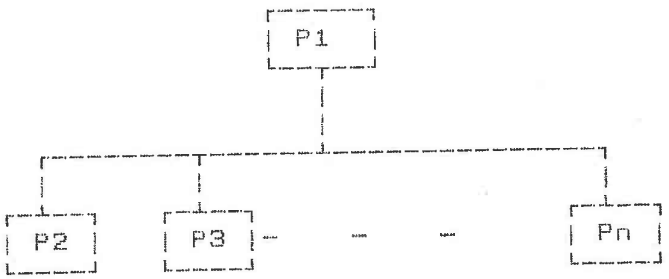
C-2-2-2 Structures dynamiques élémentaires.

IDA définit six structures élémentaires correspondant à divers enchainements de traitement exprimant la dynamique du SI.

a) enchainement séquentiel : il a lieu quand la fin d'un processus P1 provoque le déclenchement d'un processus P2.

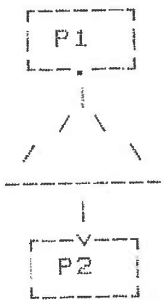
Exemple :

b) Enchainement éclaté : il a lieu lorsque la fin d'un processus P1 provoque le déclenchement simultanée de P2, P3,... Pn.

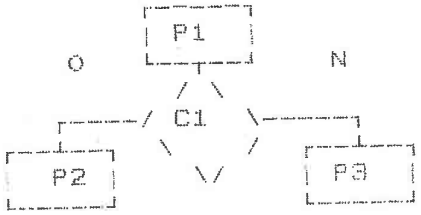


c) Enchainement multiple : il a lieu lorsque la fin d'un processus P1 provoque le déclenchement simultané des exécutions du traitement P2.

Exemple : la fin d'un processus "enregistrement d'une livraison fournisseur déclenche plusieurs processus de mise à jour relatives aux divers produits réapprovisionnés.

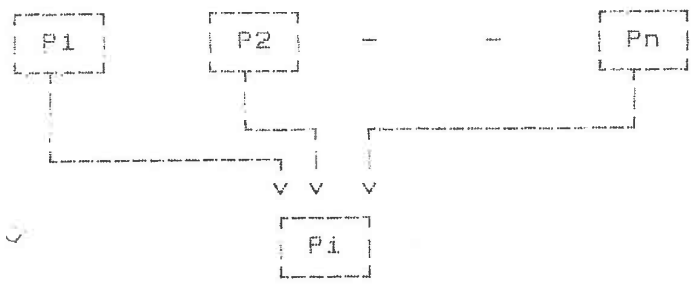


d) Enchainement conditionnel : il s'agit de conditions à remplir à la fin d'un processus P1 pour déclencher tel ou tel processus.

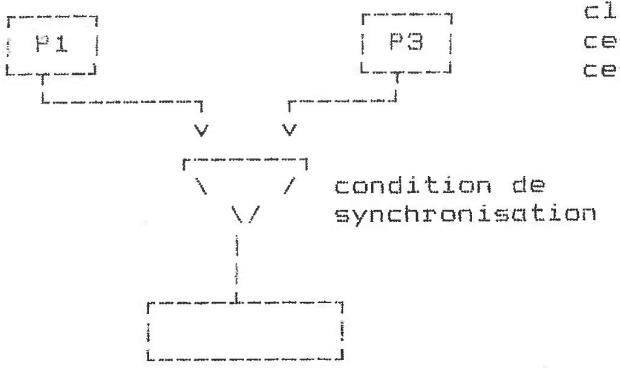


e) Enchainement convergent : il a lieu lorsque le déclenchement d'un processus Pi est provoqué par la fin d'un des

processus P1 ou P2 ou ... Pn.



f) Enchainement synchronisé : un processus P3 ne peut être déclenché qu'à la fin d'un processus P1 et à la fin d'un processus P3.



C-3 Modèle de structuration des informations.

Ce modèle sert à définir la sémantique des informations appartenant à la base des informations du SI. Il met en evidence les données et les relations qui les lient. IDA utilise le modèle Entité-Association dont les concepts fondamentaux sont les suivants :

C-3-1 concepts de base.

i) l'entité : elle est une chose concrète ou abstraite appartenant au réel perçu à propos de laquelle on veut enregistrer des informations. Une entité a une existence autonome à l'intérieur d'un groupe.

Exemple : chose concrète : un cours
chose abstraite : une personne morale.

Dans le processus de description du réel perçu on s'intéressera à la classe ou type d'entité qui est la classe de toutes les entités possibles du réel perçu qui vérifient la définition constitutive du type.

Exemple : Pour l'entité client représentant un individu on a le type-entité client représentant toutes les catégories de clients.

ii) l'association et ses composantes.

a) l'association : elle exprime le lien entre deux ou plusieurs entités (non nécessairement distinctes) où chacune assume

un rôle donné.

Exemple : la passation de commande (association) mettra en correspondance la commande X (entité) et le client Y (entité).

b) Le type d'association étant alors la classe de toutes les associations possibles du réel perçu qui vérifient la définition constitutive du type.

Exemple : le type association " passation de commande" mettra en correspondance l'entité-type client et le type entité COMMANDE.

c) On appellera degré d'un type d'association le nombre de types d'entités (non nécessairement distinctes) mis en correspondance. Dans l'exemple ci-dessus le degré est 2.

d) On appellera connectivité : le nombre minimum et maximum d'occurrences de l'association auxquels doivent participer à tout moment les occurrences des divers types d'entités.

Dans notre exemple, on aurait (0-N) pour client : un client au minimum peut ne pas passer de commande (0), au maximum pourrait passer plusieurs commandes (N).

On aurait (1-1) pour commande qui signifierait que toute commande passée doit avoir été passée par un et un seul client.

iii) <u>l'attribut.</u>

C'est la caractéristique ou qualité d'une entité ou d'une association.

Exemple : pour l'entité personne, on a les attributs caractéristiques : nom personne, prénom, adresse ....

On parlera d'identifiant d'une entité l'attribut ou le groupe d'attributs qui permettent d'identifier de façon particulière et exclusive une entité.

Exemple : le numéro client qui permet d'identifier toute entité CLIENT.

On parlera d'identifiant d'une association la concaténation des identifiants des entités sur lesquelles elle est définie.

Exemple : pour l'association ligne de commande reliant les entités COMMANDE et PRODUIT, l'identifiant de l'association sera constitué du numéro commande (identifiant de COMMANDE) et du numéro produit (identifiant de PRODUIT).

Pour garantir la cohérence et la validité de la base de données le concepteur doit prendre en compte un certain nombre de contraintes qui complètent la description obtenue en termes d'entités, d'attributs, d'associations et de connectivités. Une contrainte d'intégrité étant définie comme une propriété que doivent satisfaire les informations appartenant à la mémoire du SI. IDA distingue :

- la contrainte d'intégrité statique qui est une propriété qui doit être vérifiée à tout moment c'est à dire indépendamment des changements d'état de la base de données,

Exemple : pour une personne mariée la contrainte d'intégrité statique est représentée par date de mariage > date de naissance.

- la contrainte d'intégrité dynamique : elle définit les séquences possibles des changements d'état de la base de données.

Exemple : une personne mariée peut devenir divorcée ou veuve et se marier de nouveau.

- dépendance fonctionnelle : ce concept exprime la relation existant entre la valeur d'un attribut d'un type-entité et la valeur d'un ou de plusieurs autres attributs de ce type-entité.

Il peut également s'appliquer aux valeurs des attributs d'un type association.

Exemple : soit le type-entité PRODUIT identifié par le numéro produit, on dira qu'il y' a dépendance fonctionnelle de numéro produit vers chaque attribut non identifiant de PRODUIT si à chaque valeur de numéro produit, il correspond au plus une occurrence de chaque attribut non identifiant qui peut être le libellé, le prix unitaire.....

C-4 Modèle de la statique des traitements .

Ce modèle vise à fournir une solution pour la résolution d'un projet une fois la structuration des informations et la dynamique des traitements effectués. Cette solution une fois affinée aboutit à un algorithme.

Elle se présentera sous la forme d'une boite noire qui spécifie les objectifs à réaliser, les performances souhaitées, les informations en entrée et en sortie, ainsi que les traitements à effectuer sur les informations. Ces dernières peuvent être :

- un message entrée : émis par l'environnement à destination du SI,

- un message externe : résultant du traitement effectué dans le SI à destination de l'environnement,

- un message interne : résultat du traitement interne à destination d'une autre boite noire.

C-5 Modèle des ressources.

Son objectif est d'étudier le caractère réalisable d'une solution conceptuelle en fonction des disponibilités en ressources de l'organisation (ressources humaines, matérielles en particulier). IDA distingue les ressources réutilisables (ou processeurs) des ressources consommables :

a) **processeur :** c'est une ressource disponible pour un processus qui une fois le processus achevé reste disponible pour un autre processus.

Exemple : un terminal.

Un processeur est caractérisé par des propriétés telles que l'unité de mesure, (exemple : la mémoire centrale sera mesurée en kilo-octets), la capacité disponible, le calendrier de disponibilité, le nombre de points d'entrée (nombre de possibilités de partage de la ressource), la prise unitaire par unité de temps.

b) ressources consommables : ce sont des ressources non réutilisables une fois consommées.

Exemple : le papier imprimante.

Une ressource consommable est caractérisée par les mêmes propriétés qu'un processeur.

Des relations d'utilisation de ces ressources exprimant le taux de réquisition ou durée totale d'exécution d'un processeur seront établies sous forme de nombres ou sous forme d'une distribution de probabilités.

D / LES LANGAGES.

On distingue trois sortes de langages graphiques chez IDA :

1) un langage graphique qui permet la modélisation de la structure et de la dynamique, et qui assure la possibilité de communication aisée avec les divers utilisateurs,

2) un langage de spécification DSL (DYNAMIC SPECIFICATION LANGUAGE) qui permet de décrire un SI à partir des modèles que nous avons vus plus haut. Quelques remarques :

. le langage DSL est basé sur le modèle entité-association,

. il est non procédural ; il permet une description du SI en ordre quelconque et de manière progressive.

Notons que le langage DSL appartient à la famille des langages PSL (PROBLEM STATEMENT LANGUAGE) d'ISDOS ; il utilise les concepts d'objet à la place d'entité, de relation à la place d'association, de propriété à la place d'attribut.

3) un langage interactif d'interrogation de la base de données (QUERY SYSTEM). Il y'a en plus PROTO qui est un langage de prototypage.

Exemples :

1) représentation graphique :

- pour la structuration des informations les lettres majuscules représentent le nom d'un type entité, les lettres minuscules les noms des occurrences d'un type donné. Les types entité sont représentés par des rectangles, le type association par des hexagones. Sur le trait qui relie un type entité et un type association on inscrit le rôle joué par le type entité dans le type association. En dessous du trait on indique les connectivités minimum et maximum.

a) Exemple : commande de produit.

Une commande porte sur un produit au minimum et n produits au maximum. Un produit peut ne pas être commandé, il peut être commandé n fois.

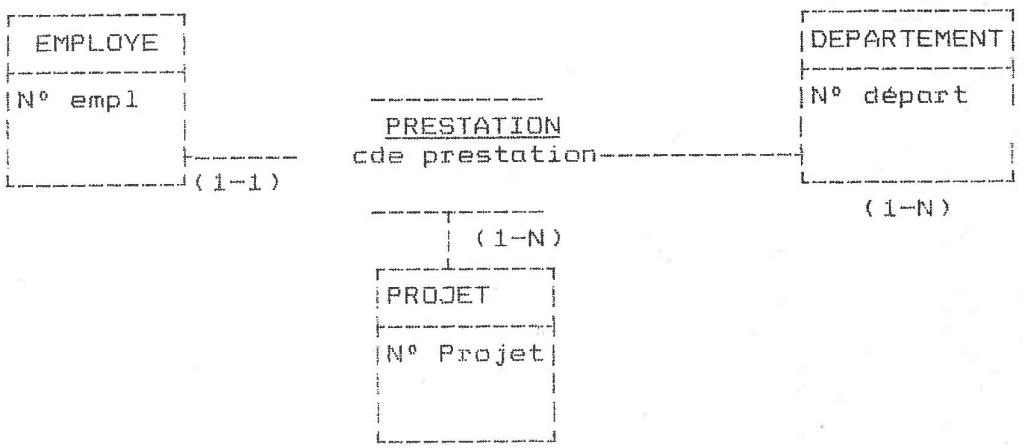
b) Exemple portant sur les dépendances fonctionnelles :

PRODUIT
Numéro produit ----------> libellé produit

Ce qui veut dire dans le type entité PRODUIT qu'il existe une dépendance fonctionnelle de numéro produit vers libellé produit.

Notons qu'à la place d'un type entité on peut avoir un type association.

Dans la structuration des informations les dépendances fonctionnelles sont représentées par des arcs.



Le type association PRESTATION lie les trois types entités EMPLOYE, DEPARTEMENT, PROJET pour indiquer la prestation effectuée par un employé dans un département pour la réalisation d'un projet.

Les dépendances fonctionnelles portent sur les identifiants.

PRESTATION
Numéro employé (N° empl) ----------> numéro département (N° Départ)

PRESTATION
Numéro département ----------> numéro projet (N° projet)

et par transitivité on a la dépendance

PRESTATION

Numéro employé ----------> numéro projet

2) Spécification en DSL.

- pour spécifier l'exemple vu plus haut sur la commande de produit :

```
DEFINE  SET                 struct-donnée-comm;
        DESCRIPTION;

        spécifications des T.E et des T.A composant la structure
        de données "struct-donnée-comm";
        COLLECTION OF       Commande,
                            Produit,
                            Ligne-commande;
        CARDINALITY IS      10.000 FOR Commande,
                            500 FOR Produit
                            DURING calendrier-1983;

DEFINE  ENTITY              Commande;
        DESCRIPTION;
        On appelle commande un ordre de commande,reconnu
        valable,passé par un client auprès de la firme;
        SYNONYMS ARE        Comm-Client,comm-cli;
        CONSISTS OF         Nr-commande,
                            Date-Commande;
        IDENTIFIED BY       Nr-Commande;

DEFINE  ENTITY              Produit;
        DESCRIPTION;
        On appelle Produit une référence figurant au catalogue
        diffusé par la firme;
        CONSISTS OF         Nr-Produit
                            Libellé-Produit
        IDENTIFIED BY       Nr-Produit

DEFINE  RELATION            Ligne-Commande;
        DESCRIPTION;
        Une ligne-commande représente pour une commande acceptée
        la quantité commandée d'un produit figurant au catalogue;
        RELATES             Commande Produit;
        ROLE NAME IS        "Concerne" For Commande,
                            "Commandé par" FOR Produit;
        CONNECTINITY IS     sp-1-n  FOR Commande,
                            sp-0-n  FOR Produit,
        CONSIST OF          Qte-commandée;
```

- soit la structuration statique suivante :

```
DESCRIPTION :

DEFINE  PROCESS     trait-commandes-clients;

        SYNONYM IS trait-commcli;

        ATTRIBUTE  niveau "application";
```

```
DEFINE      PROCESS         enreg-commande-client;
            SYNONYM IS      enreg-comm-cli;
            ATTRIBUTE       niveau "phase";
            PART OF         trait-comm-cli;

DEFINE      PROCESSES       identif-client;
                            enreg-client;
                            modif-adresse-client;
                            enreg-commande;
                            verif-ligne-commande;
                            enr-ligne-commande;
                            rejet-ligne-commande;
            ATTRIBUTE       niveau "fonction";
            PART OF         enreg-commande-client;
```

3) Utilisation du QUERY LANGUAGE.

LET S2 = ENTITY AND NOT ? IDENTIFIED BY !

Pour demander quelles sont les entités qui ne possèdent pas d'identifiant.

E/ OUTILS :

On distingue :

- l'analyseur DSA (DYNAMIC SPECIFICATION ANALYZER) qui permet grâce à ses programmes de mise à jour de stocker les spécifications rédigées à l'aide du langage DSL,

- les outils de documentation (DOCGEN) qui permettent d'extraire de la base de données des informations de synthèse, de créer ou de gérer des dossiers,

- les outils de vérification de la cohérence et de la complétude des spécifications effectuées,

- le système de prototypage (DSL-PROTO) qui permet une réalisation à échelle réduite du SI dont on a défini les spécifications et déterminé les ressources disponibles,

- l'outil DSL-SIM sert à évaluer les performances du SI, il en permet une approche du fonctionnement avant le fonctionnement proprement dit,

- le poste de travail IDA qui autorise une interface conviviale avec le système central. Ce poste comprend à un niveau décentralisé un éditeur de langage DSL, un éditeur des commandes des rapports de DSL-SIM et DSL-PROTO, un gestionnaire d'écran associé à DSL-PROTO et un poste de travail graphique.

L'ensemble des outils ci-dessus énumérés forme le système logiciel IDA.

Des études sont en cours pour réaliser un module d'aide à l'analyse organique appelé DSL-ORGA en fait commercialisé avec

l'atelier DELTA par METSI pour assurer les étapes ultérieures à la spécification.

F/ ETAPES.

Le cycle de conception IDA se déroule selon les étapes suivantes :

1) L'analyse d'opportunité : Pour justifier ou non de l'intérêt d'une automatisation. Elle comprend plusieurs aspects une fois que la décision d'analyse a été prise :

a) identification du projet : elle consiste à déterminer sa nature, ses fonctions, ses déficiences,

b) définition de projet cadre : elle permet la spécification des besoins sous forme d'objectifs à atteindre (objectifs informationnels, objectifs d'efficacité),

c) étude critique du SI existant : elle se fait généralement au moyen d'un diagramme de flux des informations,

d) élaboration des diverses solutions : proposer des solutions en fonction des critiques et de la dimension du projet de cadre,

e) choix d'une solution : évaluation des diverses solutions sur la base des critères d'efficacité et de choix d'une solution,

f) mise à jour du projet cadre : une fois choisie une solution on indiquera les résultats attendus, le planning des tâches à effectuer ultérieurement, les points de contrôle dans la réalisation du projet.

2) L'analyse conceptuelle :

C'est la phase de conception du projet, elle se subdivise en trois grandes parties :

a) élaboration et validation du sous-schéma conceptuel correspondant à chaque phase du projet ; à ce niveau on construira :

- les sous-schémas relatifs aux informations avec élimination des redondances, les éventuelles désagrégations et décomposition de type-entités ;

- les sous-schémas relatifs aux traitements avec spécification de la dynamique et de la statique d'une phase ;

b) établissement du schéma conceptuel par consolidation des sous-schémas. Cela consiste à assembler les sous-schémas des différentes phases en ce qui concerne les informations et les traitements et ce avec élimination des redondances.

c) quantifications : tous les éléments de la solution conceptuelle seront quantifiés à savoir :

- les données (estimation des volumes en fonction des différents type-entités et type-associations),

- les traitements (nombre d'exécutions par unité de temps),

- les messages (nombre de caractères véhiculés par un message, détermination du volume du trafic),

- les opérations sur la base de données (fréquence d'exécution des opérations élémentaires sur les bases de données),

- les ressources à mettre en oeuvre.

6/ CONCLUSION .

IDA se présente comme une méthode complète de conception de SI.

Ses points forts sont :

- une approche très modulaire du SI,

- une modélisation intégrée des aspects statiques et des aspects dynamiques,

- un langage de spécification complet, facilement compréhensible,

- une assistance de nombreux outils au cours de la conception.

Ses points faibles sont :

- l'absence d'informations sur les phases de réalisation et sur le cycle de management,

- un trop grand nombre de modèles dont on ne voit pas très bien le lien entre eux.

II-2-7 LA METHODE MERISE .

A/ PRESENTATION.

Dès 1976 le Ministère de l'Industrie en FRANCE avait mesuré l'importance des méthodes et senti la nécessité de doter le secteur public et privé d'une méthode moderne de conception de SI. En 1977 plusieurs sociétés de services ainsi que le C.E.T.E d'AIX EN PROVENCE furent sélectionnées. C'est dans ce contexte qu'est née la méthode MERISE du nom du fruit d'un arbre le MERISIER, méthode qui fut présentée dans un livre par H.TARDIEU, A.ROCHFELD, et R.COLETTI [TAR 83].

MERISE est une méthode de conception et de développement de SI orientée bases de données et systèmes transactionnels qui effectue une modélisation parallèle des données et des traitements, met en oeuvre des modèles et des outils. Elle est commercialisée par de nombreuses sociétés de service en particulier SEMA/METRA et MEGA.

OBJECTIFS :

Fournir une philosophie, une démarche, des modèles et des formalismes destinés à la conception et à la réalisation de SI de toute nature.

Proposer des moyens qui permettent une conception de SI fiables et évolutifs orientés bases de données et systèmes interactifs, inclure des procédures de validation y compris une participation active des utilisateurs.

B/ NIVEAU D'ABSTRACTION.

MERISE couvre les trois cycles définis par BODART et AL à savoir le cycle d'abstraction, le cycle de contrôle et le cycle de décision. MERISE se base sur le principe de l'abstraction pour traiter des aspects statiques et dynamiques du SI et de son évolution. Le cycle d'abstraction utilise les trois niveaux de représentation (conceptuel, logique, physique).

MERISE peut donc être située à un haut niveau d'abstraction.

C/ CONCEPTS.

C-1/ Description statique du SI et le modèle conceptuel des données (M.C.D).

la description statique consiste à représenter les données grâce à un modèle appelé modèle conceptuel des données (M.C.D) qui définit la sémantique des informations appartenant à la base des informations du SI. Les données sont liées entres elles par des relations. MERISE utilise le modèle entité-association. Les concepts sur lesquels s'appuie le M.C.D sont :

a) l'individu (appelé aussi objet ou entité) : c'est une chose abstraite ou concrète ayant une existence autonome appartenant au réel perçu et ayant un sens pour ce dernier. Les individus sont regroupés en classes ou types caractérisés par le fait qu'ils partagent les mêmes propriétés. L'individu est décrit par une liste de propriétés. Tout individu est distinguable d'un autre individu grâce à une propriété particulière appelée identifiant.

Exemples : objet concret : un produit donné de la classe PRODUIT
objet abstrait : une personne morale.

b) la relation (appelée aussi association) : c'est le lien unissant deux ou plusieurs individus (non nécessairement distincts). Une relation n'a pas d'existence propre, elle n'existe qu'en fonction des individus. Une relation-type est une classe d'association d'individus appartenant à une même liste d'individus-types,

Exemple : la relation unissant la classe "PARENT" à la classe "ENFANT" qui est "ELEVER".

Certaines caractéristiques apportent plus de précision au concept de relation :

- la dimension d'une relation-type est définie par le nombre d'individus-type (non nécessairement distincts) mis en correspondance,

Exemple : dimension de l'exemple précédent = 2.

- la cardinalité : elle définit le nombre minimum et maximum où une occurrence de l'individu-type peut et doit intervenir dans les occurrences de la relation ,

Exemple : Un parent élève au moins un enfant et au maximum N enfants (cardinalité 1-N); un enfant a au moins une relation "ELEVER" avec un parent et au plus deux (cardinalité 1-2). Sont exclus les orphelins.

b) la propriété : c'est le plus petit élément d'information manipulé par l'organisation et ayant un sens en lui-même. Une propriété caractérise un individu ou une relation.

Exemple : le poids d'un produit.

Pour garantir l'intégrité sémantique de la base d'informations le concepteur ne doit pas seulement se contenter des cardinalités maximum, il doit tenir compte des contraintes d'intégrité fonctionnelles (CIF) ou encore dépendances fonctionnelles qui sont les relations existant entre les valeurs des propriétés des individus-types ou des relations-types.

Notons que le M.C.D obtenu (la construction de la base de données se place dans le cadre des SGBD de type CODASYL) peut être converti dans un schéma de bases de données relationnelles.

C-2/ <u>Modélisation de la dynamique des traitements ou le modèle conceptuel des traitements (M.C.T).</u>

La modélisation consiste à simuler le comportement du SI dont le contenu est transformé par des sollicitations extérieures à l'orga-

nisation et par des faits internes. Les traitements représentent les aspects dynamiques du SI pour la modélisation desquels MERISE utilise les concepts suivants :

a) événement : c'est un fait dont l'apparition est de nature à déclencher l'exécution de traitements au sens large.
Un événement est dit externe quand il provient de l'extérieur de l'organisation, interne quand il est le produit (résultat) de la réponse codifiée de l'organisation à un événement. Chaque événement appartient à une classe ou type,

Exemple : arrivée d'une commande ; cet événement entraine le traitement de la commande (identification par exemple) ce qui donne comme résultat une commande identifiée ou une commande inconnue.

b) opération : c'est un ensemble structuré de règles de gestion dont l'activation est provoquée par l'apparition d'un événement. Chaque opération appartient à une classe ou type,

Exemple : l'opération d'identification de la commande de l'exemple précédent.

c) synchronisation : c'est la précondition (sous forme booléenne) qui reflète l'absence ou la présence d'événements pour le démarrage de l'opération. Chaque synchronisation appartient à une classe ou type,

Exemple : le résultat (émission commande) et l'événement (émission commande) déclenchent l'opération "déblocage commande inconnue".

d) processus : c'est une suite d'opérations engendrées par un événement externe ou plusieurs événements ; il est défini comme permettant d'agréger des ensembles synchronisés d'opérations.

Exemple : le processus "réception de marchandises" est à l'origine de plusieurs opérations.

Pour construire le M.C.T MERISE utilise un graphe des flux qui permet d'identifier les acteurs jouant un rôle dans le réel perçu et les flux d'information échangés, ce qui met en évidence les événements et les résultats.

C-3 Prise en compte des aspects organisationnels :

En fait chez MERISE le niveau organisationnel correspond au niveau logique puisqu'on y évoque l'organisation logique des procédures de traitement et les lieux d'exécution de ces traitements.

C-3-1 Modèle organisationnel des traitements (M.O.T).

Le M.O.T complète le M.C.T en réintroduisant les éléments d'organisation dont on avait fait abstraction au niveau conceptuel. C'est un niveau de description relatif aux choix d'organisation des traitements (entre l'homme et la machine, temps réel ou batch, centralisation ou décentralisation).

Les concepts utilisés tournent autour du concept de "procédure".

Une procédure est un ensemble de phases exécutées consécutivement par un poste ; une phase étant une séquence de tâches exécutées consécutivement ou sein d'un poste de travail.

Exemple : une procédure de saisie-contrôle peut comprendre une phase

de saisie suivie par une phase de contrôle, chacune de ces phases pouvant se décomposer en plusieurs tâches ; tout ce travail s'effectuant au poste de réception des commandes par exemple.

C-3-2 Modèle logique des données .

De même que pour les traitements on procédera au niveau organisationnel pour les données à une intégration au MCD des choix d'organisation effectués en matière de gestion de données.

Le MCD validé sera alors transcrit dans un formalisme dépendant du choix organisationnel (fichiers ou bases de données) un choix qui est dicté par la disponibilité d'un logiciel donné.

C-4 Modèle opérationnel des traitements .

Le niveau opérationnel des traitements dépend des caractéristiques du logiciel et du matériel utilisé ; l'objectif étant de définir comment les traitements définis au niveau conceptuel vont être mis en oeuvre.

A ce niveau vont intervenir les méthodes de programmation, les concepts de conversation (équivaut à la procédure en temps réel du niveau organisationnel), de tâche temps réel ....

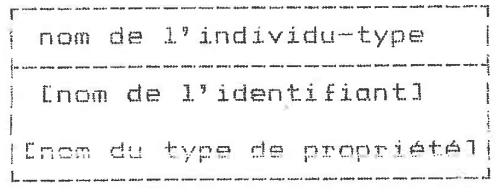
D/ LES LANGAGES.

On distingue deux types de langage :

- un langage graphique qui permet de représenter les divers modèles relatifs à la structure et à la dynamique du SI.

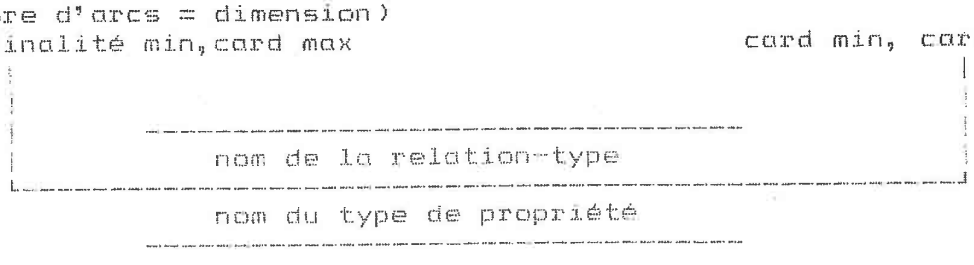
* représentation du M.C.D :

les individus-types par des rectangles ,

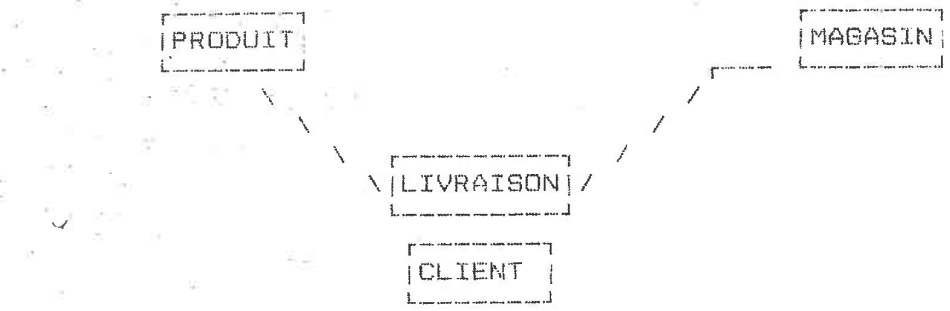


la partie inférieure du symbole est optionnelle

les relations-types par des ovales ,



Exemple :



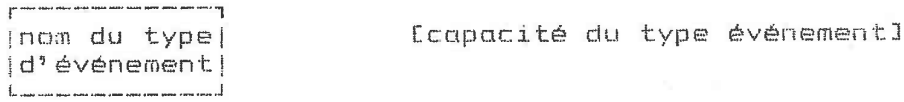
PRODUIT X CLIENT ------ LIVRAISON

Commentaire : Un produit peut être ou non impliqué dans une relation-type LIVRAISON, il peut être impliqué dans plusieurs livraisons. Un magasin peut être ou non impliqué dans une livraison s'il l'est, il peut être impliqué dans plusieurs livraisons. Un client est nécessairement impliqué dans au moins une livraison, mais il peut être impliqué dans plusieurs livraisons.

La dépendance fonctionnelle PRODUIT X CLIENT ----> LIVRAISON portant sur la relation-type LIVRAISON exprime que pour un client donné et un produit donné, un seul magasin peut livrer ce produit à ce client.

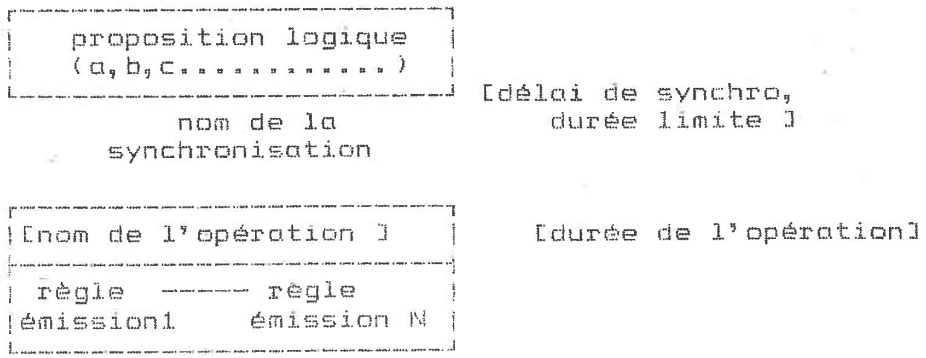
* représentation du M.C.T :

types d'événements par un cercle comportant le nom du type d'événement et un chiffre indiquant la capacité du type d'événement,



les types de synchronisation et d'opération de la façon suivante :

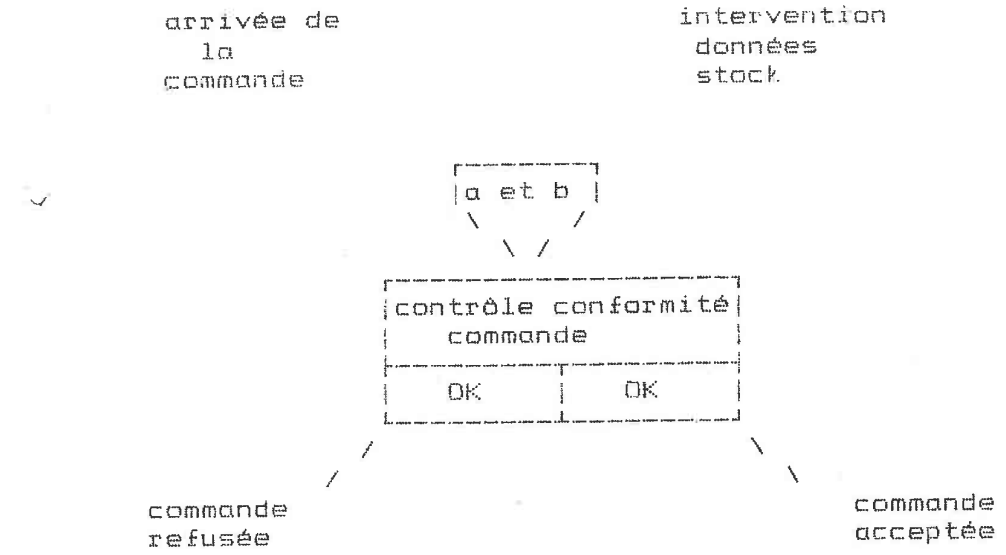
nom du type d'événement contributif	nom du type d'événement contributif	nom du type d'événement contributif	[capacité du type d'événe.]



E1 [événement interne]

En [événement interne]

Exemple :



Commentaire : les événements "ARRIVEE DE LA COMMANDE" et "INTERVENTION DES DONNEES STOCK" entrainent l'opération "CONTROLE CONFORMITE" de la commande émise. Le résultat d'une telle opération pouvant être le rejet ou l'acceptation de cette commande.

* représentation du M.O.T :

On reprend la représentation du M.C.T à laquelle on ajoute :

- une colonne pour indiquer le temps (périodicité) ,
- une colonne pour indiquer la nature du traitement ,
- une colonne pour indiquer le poste de travail concerné.

temps	enchainement des procédures	nature procédure	poste de travail
	voir M.C.T		

Exemple :

jour	catalogue de la centrale d'achat ↓ PF1 encodage ↓ catalogue encodé	manuelle	atelier de saisie du siège

- un langage de spécification (à l'état théorique) pour décrire les données et les traitements.

* description des données :

Exemple :

```
INDIVIDU-TYPE PERSONNE
IDENTIFIANT numéro insee
DESCRIPTION date de naissance,lieu de naissance,prénoms
RELATION-TYPE COURS
DIMENSION 3
COLLECTION  PROFESSEUR,SALLE,CLASSE
CARDINALITE PROFESSEUR 0,n
            CLASSE     0,n
            SALLE      0,n
DESCRIPTION nombre d'élèves présents
```

* description des traitements :

Exemple :

```
EVENEMENT ext 1 EXTERNE
  DECISION D'ORGANISER UNE NOUVELLE CONFERENCE c PENDANT
  MEETING TC (NTC)
-
-
-
SYNCHRONISATION  DEFINITION DU COMITE
SI               ext 1

OPERATION        01
                 - génération d'un numéro de conférence NCONF
                 -
                 -
                 -
SYNCHRONISATION  REJET LETTRE D'INTENTION

SI               (INT 24 OU INT 25) ET EXT 5

AVEC             date. EXT 5 = date de fin de
                 réception des lettres d'intention
-
-
-
```

E/ OUTILS :

MERISE a été utilisée comme méthode de conception et de développement de SI dans plusieurs sociétés de services qui ont pour la plupart développé des outils de soutien à la méthode.

Notons que les outils proposés sont très parcellaires et que les diverses propositions des sociétés de services sont disparates sans aucune coordination entre elles.

Prenons l'exemple de la plate-forme de développement de la société de services SEMA-METRA.

Cette entreprise propose un ensemble intégré d'outils de Génie logiciel :

- couvrant l'intégralité du cycle de vie avec enrichissement progressif des définitions construites à chaque étape,

- permettant la réalisation de ces définitions à chaque niveau du cycle d'abstraction.

L'ensemble d'outils intégrés est organisé autour d'une base d'informations dont les composants sont les suivants :

- DIAMANT III : dictionnaire généralisé des données et des traitements du SI qui comprend les fonctions d'administration, de définition des modèles, d'aide à la conception ainsi qu'un outil de consultation multi-critères por exploiter le contenu de la base,

- METRADOC : pour élaborer la documentation textuelle et graphique associée au projet,

- SCREEN PAINTER : permet de réaliser les dessins des masques d'écran de l'application ainsi que la définition des propriétés associées aux zones définies. Il intègre une procédure d'archivage/restitution des informations saisies dans la base commune gérée par DIAMANT III,

- MAQUETTAGE : en phase de conception détaillée cet outil permet de définir les règles d'enchainement d'écran relatives à un dialogue écran. On trouve également au niveau de cet outil les écrans de simulation permettant à l'utilisateur d'effectuer à l'avance certains traitements,

Au niveau de la réalisation la méthode utilisée HOS (HIGHER ORDER SOFTWARE) est supportée par l'outil USE.IT (USER SPECIFICATION AND EVALUATION INTEGRATED TOOL) qui comprend :

* un éditeur qui permet la définition graphique des procédures de contrôle ainsi que la spécification des variables entrantes et sortantes à chaque niveau noeud de l'arborescence définie,

* un analyseur qui permet d'effectuer les contrôles intrinsèques sur les données saisies et des contrôles de cohérence sur les définitions saisies,

* un générateur qui permet l'assemblage des composants (structures, primitives, types de données), le contrôle de leur implémentation physique, ceci afin de générer du code source dans le langage choisi et dans l'environnement technique retenus.

F/ <u>ETAPES :</u>

MERISE propose les étapes suivantes :

- <u>ETAPE 1 :</u> schéma directeur qui consiste à découper le SI en domaines ou sous-ensembles indépendants et à déterminer leur articulation.

- <u>ETAPE 2 :</u> étude préalable sur chaque domaine identifié lors de l'étape précédente. Pour évaluer le futur SI un sous-ensemble représentatif peut être choisi.

Cette étape comprend une phase de recueil (qui permet une formalisation du niveau conceptuel des données et des traitements, de cerner les dysfonctionnements du SI) et une phase de conception (en fonction de la phase précédente définir les orientations nouvelles et les solutions au niveau conceptuel, organisationnel et opérationnel),

- Etape 3 : étude détaillée dont l'objectif est de compléter dans le détail les choix effectués à l'issue de l'étude préalable et de les valider.

- Etape 4 : étude technique dont l'objectif est la prise en compte de l'environnement technique pour passer des spécifications détaillées à la définition d'une solution en termes de traitements (conception du logiciel, conception de la structure physique de la ou des bases de données).

- Etape 5 : production de logiciels : divers codages et tests unitaires ou d'enchainement.

- Etape 6 : mise en oeuvre ou utilisation effective des nouvelles fonctions automatisées.

- Etape 7 : maintenance pour les éventuelles corrections et adaptations assurant ainsi l'évolution du SI.

6/ CONCLUSION .

MERISE est une méthode complète de conception et de développement de SI.

Ses points forts sont :

- une couverture complète du cycle de conception,

- une assistance du concepteur par de nombreux outils pendant le déroulement du cycle de vie (outils parcellaires proposés par les S.S.I.I sans une reélle coordination entre eux ce qui peut créer des problèmes de compatibilté entre ces outils avec comme résultat final de ne pas répondre aux attentes de l'utilisateur),

- une modélisation en parallèle des aspects statiques et dynamiques du SI, avec un rapprochement ultérieur des deux modélisations,

- un management satisfaisant de la conception et du développement du SI,

Ses points faibles sont :

- une pseudo-symétrie données-traitements, les traitements étant plus favorisés,

- l'absence d'une réelle méthode pour la modélisation conceptuelle des données,

- sa relative complexité pour la compréhension de l'utilisateur non averti,

- l'absence de langage d'implémentation et une certaine lourdeur.

TROISIEME PARTIE

BILAN

ET

PERSPECTIVES

3-1 INTRODUCTION.

Au terme de cette étude à la fois descriptive et analytique de ces sept methodes de conception de SI il devient possible de tenter une comparaison en mettant en évidence leurs points communs et leurs divergences.

Rappelons que notre étude porte sur un certain nombre de thèmes rélatifs aux divers concepts liés aux trois niveaux de description et aux paramètres associés aux différents cycles de vie du logiciel :

- au niveau du cycle d'abstraction l'accent est mis sur l'étude du niveau conceptuel; on s'intéresse au fait de savoir si la méthode couvre ou non tout le cycle de vie du SI et quels sont les moyens (modèles de spécification, langages de spécification) utilisés pour répondre aux qualités du schéma conceptuel,

- au niveau du cycle de contrôle une étude des divers outils de spécification sera effectuée.

Les résultats obtenus au cours de cette étude comparative nous permettront entre autres de déterminer quelles doivent être les caractéristiques d'une bonne méthode de conception de SI et avancer ainsi un certain nombre d'idées de base qui serviront dans le cadre de l'amélioration de méthodes de conception existantes ou dans le cadre de l'élaboration d'une nouvelle méthode de conception de SI. Ceci nous permettra de compléter la réflexion faite dans la première partie de notre mémoire.

3-2 COMPARAISON DES SEPT METHODES.

Dans un premier temps nous apprécierons chaque méthode selon les éléments définis du cadre de comparaison à savoir les objectifs, le cycle de conception, le niveau d'abstraction, les concepts, les langages et les outils caractéristiques de la méthode.

Dans un deuxième temps un résumé des points forts et des points faibles de chaque méthode sera effectué.

3-2-1 POSITION DES METHODES VIS A VIS DE CERTAINS CRITERES :

A / PRESENTATION : ORIGINE , OBJECTIFS, CYCLE DE VIE.

Une remarque s'impose : la plupart des méthodes retenues sont d'origine universitaire : seules NIAM et MERISE voient la participation d'organismes extra-universitaires, les recherches en elles-mêmes étant d'origine para-universitaire. Cette voie (recherches universitaires en association étroite avec les milieux professionnels) garantit souvent la qualité et la richesse des études; par un phénomène de feed-back il se produit une validation ou non des études théoriques universitaires (ce qui est très positif pour l'avancement de la recherche) tout en conservant une base de réflexion formelle qui nous semble indispensable.

Toutes ces méthodes mettent l'accent sur l'aspect "conception" du SI, DADES étant un peu en retrait car elle se présente plus comme un outil qu'une méthode de conception de SI. Par contre la couverture du cycle de conception est plus ou moins complète selon ces méthodes :

- USE : couvre les étapes de spécification, conception, implémentation,

- ACM/PCM : accorde une importance particulière à l'étape de conception,

- DADES : met en évidence l'aspect spécification des besoins et l'aspect conception,

- NIAM : accorde une grande importance à l'étape de conception,

- REMORA : accorde une grande importance à l'aspect conception et à l'aspect logique,

- IDA : est une méthode de conception de SI relativement complète (le passage au niveau logique n'est pas en tout cas explicite),

- MERISE : est une méthode complète de conception de SI.

B / NIVEAU D'ABSTRACTION.

Toutes les méthodes étudiées peuvent être situées à un haut niveau d'abstraction avec quelques réserves pour DADES qui apparait plutôt comme une méthode orientée analyse des besoins.

Si on se réfère aux trois cycles définis par BODART ET AL on voit que :

- USE : couvre les trois cycles (abstraction, contrôle, décision),

- ACM/PCM : couvre le cycle d'abstraction et en partie le cycle de contrôle,

- DADES : ses propositions concernent plutôt le niveau conceptuel,

- NIAM : couvre le cycle d'abstraction, fait une certaine référence au cycle de contrôle et au cycle de décision,

- REMORA : couvre le cycle d'abstraction et de contrôle (en fait ce contrôle reste limité au processus de conception),

- IDA : couvre en grande partie les trois cycles,

- MERISE : couvre complètement les trois cycles.

On voit après ce bref survol que toutes les méthodes traitent de l'étape de conception du SI ; elles lui accordent en fait la plus grande importance.

C / CONCEPTS.

Toutes les méthodes préconisent une modélisation aussi bien des aspects statiques que des aspects dynamiques du SI avec une réserve pour DADES où les concepts utilisés sont soit peu clairs ou pas du tout explicites (en particulier au niveau de la structure).

Si la modélisation statique est claire dans l'ensemble (voir réserves ci-dessus), la modélisation de la dynamique connaît quelques

variantes : chez REMORA, MERISE, IDA, elle est puissante et explicite, assez explicite dans USE, ACM/PCM, peu explicite dans NIAM.

Toutes les méthodes se sont attachées à définir les concepts utilisés.

D / LES LANGAGES.

Toutes les méthodes proposent un langage graphique : il est très puissant chez NIAM et ACM/PCM, et élaboré chez REMORA, IDA, MERISE. Toutes aussi proposent un langage de spécification de la modélisation conceptuelle obtenue :

- le langage est soit procédural (NIAM, REMORA, USE, ACM/PCM (langage BETA sur les opérations)) ou déclaratif,

- USE élargit la gamme des langages en proposant un langage d'implémentation et un langage de conception de programme.

E / OUTILS.

La nature des outils proposés est très diverse : outils d'élaboration et de contrôle du schéma conceptuel, de simulation, de de prototypage, de documentation....

Toutes les méthodes à l'exception de ACM/PCM,DADES (qui ne propo - sent rien dans les documents étudiés), proposent des outils aux différentes étapes de conception. USE, IDA, et MERISE proposent un environnement logiciel relativement complet bien que chez MERISE on assiste à un manque de coordination dans la proposition des outils ; NIAM se focalise sur l'aspect statique du SI, alors que REMORA propose des prototypes qui traitent surtout du niveau conceptuel.

F / ETAPES.

L'étape de conception est couverture par toutes les méthodes. Certaines méthodes (USE, IDA, MERISE) couvrent tout le cycle de vie, REMORA ne traite que des étapes conceptuelle et logique.

3-2-2 RESUME DES POINTS FORTS ET DES POINTS FAIBLES :

USE : Points forts : importance accordée à la spécification,
couverture du cycle de conception,
modélisation à l'aide des types abstraits,
prise en compte de l'utilisateur,

Points faibles : cycle de management inexistant,
caractère non explicite du modèle conceptuel.

ACM/PCM : Points forts : conception orientée procedure,
modélisation modulaire tenant compte
des aspects statiques et des aspects
dynamiques,
langages (graphique et de spécification),

Points faibles : défaut d'environnement logiciel,
modèle abstrait encombré de
considérations d'implémentation,
spécification obtenue éloignée de
l'aspect implémentation du SI.

DADES : Points forts : dérivation et validation des spécifications,

Points faibles : méthode incomplète (pas de couverture du cycle de conception, méthode intégrable à d'autres méthodes),
manque d'outils.

NIAM : Points forts : l'analyse de l'information offrant une grande puissance de description de la structure du SI,
langage utilisé et notation graphique

Points faibles : description incomplète de la dynamique,
complexité du schéma conceptuel.

REMORA : Points forts : conception des aspects dynamiques et des aspects statiques,
passage niveau conceptuel niveau logique,

Points faibles : absence de couverture du cycle de vie,
formalisme d'abstraction au niveau conceptuel relativement complexe,
absence de véritables outils.

IDA : Points forts : approche modulaire du SI,
outils et langages de spécification,
modélisation intégrée des aspects statiques et dynamiques,

Points faibles : absence d'information sur les cycles de réalisation et de management,
pléthore de modèles.

MERISE : Points forts : couverture du cycle de conception,
outils qui accompagnent la conception,
modélisation tient compte des aspects statiques et des aspects dynamiques,
management de la conception et du développement du SI,

Points faibles : traitements privilégiés par rapport aux données,
absence de méthode pour la modélisation des données,
absence de langage d'implémentation,
méthode lourde.

3-2-3 RECAPITULATIF ET TABLEAU DE SYNTHESE :

critères / méthodes	origine		orient	concepts				lang. spec.		outils			étapes		
	1	2	3	4	5	6	7	8	9 / 10	11	12	13	14	15	16 / 17	18
U S E	X		X	B	G	I	I	X	K / L	DC PD PR	PP		RT T	X		X
ACM / PCM	X		X	C	X	I	I	XX	M / O	DC, PDD, TRT	AP			X		
DADES	X		A	D	H	H	H	X	P					X		
NIAM		X	X	E	X	J	X	XX	Q / O	PDD PRT	NB, IS		EF	X		X
REMORA	X		X	E	X	XX	XX	XX	R	PR	PL		SG PC	X	-- X	
IDA	X		X	F	X	X	XX	XX	S	NPR	DS DP	DC DM	ID	X		X
MERISE		X	X	F	X	X	XX	XX	T	NPR	NBM			X	X --	X

SIGNIFICATION DES SYMBOLES

CRITERES :

origine :

1 : universitaire
2 : extra universitaire

orientation :

3 : conception
4 : autre que conception

concepts :

5 : explicite
6 : statique
7 : dynamique

langage de spécification :	outils :	étapes :
8 : graphique	12 : contrôle	15 : conception
9 : nom du langage	13 : documentation	16 : autre
10 : niveau du langage	14 : autre	17 : logique
11 : nature du langage		18 : cycle complet

CONTENU DU TABLEAU

origine :	1 ou 2	X : methode universitaire ou extra-universitaire
orientation :	3	X : orientation conception A : DADES considéré comme un outil à incorporer à d'autres méthodes
	4	B : spécification besoins, conception, implémentation C : conception D : spécification besoins, conception E : conception F : couverture complète du cycle de conception
concepts	5	G : concepts non explicites X : concepts explicites H : concepts peu clairs
	6	I : concepts de la statique assez explicites H : concepts de la statique peu clairs X : concepts de la statique explicites
	7	I : concepts de la dynamique assez explicites H : concepts de la dynamique peu clairs X : concepts de la dynamique clairs XX : concepts de la dynamique puissants et explicites
langages de spécification	8	X : langages explicites XX : langages puissants
	9	K : langage BASIS pour la spécification M : langage BETA P : absence de nom pour les langages Q : langage RIDL R : langage ISDEL S : langage DSL T : existence de plusieurs langages de spécification
	10	L : langage PLAIN pour le relationnel O : langage de haut niveau
	11	DC : déclaratif PD : prédicatif PR : procédural PDD : prédicatif au niveau des données PRT : procédural au niveau des traitements TRT : traitement en pré-post NPR : non procédural

outils	12-13	PP :	pas d'outils au niveau spécification, outils liés à USE
		AP :	aucun outil présenté
		NB :	outils nombreux
		IS :	ISDIS outil le plus important de NIAM
		PL :	automate appelé "pilote" utilisé pour la conception assistée par ordinateur
		DS :	analyseur DSA
		DC :	outil de documentation
		DP :	système de prototypage
		DM :	systéme d'évaluation des performances
		NBM :	outils nombreux liés à MERISE
	14	RTT :	outils RAPID, TROLL, TDI, USE CONTROL SYSTEM
		EF :	ENFORCER (NIAM) outil d'exploitation de la grammaire conceptuelle compilée
		SG :	SGSI : système de gestion de SI
		PC :	processeur d'événements
		ID :	poste de travail IDA
	15	X :	étape de conception couverte
	16	X :	étape logique couverte
	17	X :	autres étapes couvertes
	18	X :	cycle de conception entièrement couvert

Dans le tableau précédent nous avons mis en évidence la position de chaque méthode par rapport aux points suivants :

- origine, orientation,
- concepts,
- langages de spécification,
- outils,
- étapes.

On y voit transparaître certains des points forts et certains des points faibles des méthodes : on voit par exemple que USE est d'origine universitaire, qu'il couvre le cycle de vie mais que par contre ses concepts ne sont pas assez explicites.

3-3 VERS DE NOUVELLES PROPOSITIONS :

En nous appuyant sur le cadre de comparaison que nous avons défini et sur les résultats de notre étude comparative nous pouvons dégager un certain nombre de propositions permettant de définir les caractéristiques d'une bonne méthode de conception de SI :

A / NIVEAU D'ABSTRACTION :

Une bonne méthode de conception de SI doit se situer à un niveau d'abstraction qui permette d'éviter le piège de la dépendance données-programme des méthodes classiques. Pour cela elle doit penser "conception", s'intéresser aux invariants statiques (données) et aux aspects dynamiques (changements d'état de l'organisation, transitions d'un état à un autre, réactions de

l'organisation face à des événements synchronisés ou non), tenir compte de l'évolution de cette organisation; en fait elle doit intégrer le temps ceci aussi bien au niveau conceptuel qu'au niveau logique.

B / FIDELITE PAR RAPPORT AU REEL PERCU :

La méthode de conception de SI doit permettre au SI d'être fidèle au contenu de la réalité organisationnelle étudiée, sans donner une représentation partielle de l'organisation.
La conception obtenue doit reproduire la sémantique du réel perçu, elle doit être fidèle et cohérente ceci indépendamment de tout aspect technique.

C / COUVERTURE COMPLETE DU CYCLE DE CONCEPTION :

Autant que possible et dans le souci de mettre à la disposition du concepteur un ensemble cohérent, une bonne méthode de conception devrait traiter de toutes les étapes du cycle de conception et s'intéresser aux trois niveaux de description communément admis.

Pour que cette couverture soit effective la méthode doit assurer les transitions d'une phase à une autre grâce à des liens, et montrer que l'ensemble est bien coordonné pour permettre au concepteur de naviguer d'une étape à une autre soit en remontant vers certaines étapes de niveau plus élevé à partir d'étapes de niveau inférieur soit en effectuant le chemin inverse (descente).

D / ACCORDER UNE GRANDE IMPORTANCE A L'UTILISATEUR :

Tenir compte aussi bien de l'utilisateur que du développeur doit être une préoccupation majeure. En effet, dans le cas de systèmes interactifs (systèmes appelés à se généraliser), une spécification du dialogue utilisateur-programme doit être effectuée pour assurer une bonne compréhension et de l'utilisateur et du développeur. Cette mutation par rapport à l'informatique classique où l'informaticien se voyait si puissant pour se passer de l'utilisateur assurera une adéquation des traitements aux besoins spécifiés pour une plus grande efficacité de l'informatisation.

A ce propos des guides pour le dialogue utilisateur-programme pourront être élaborés et des possibilités d'expérimentation offertes par le biais du prototypage qui permettront d'effectuer les validations souhaitées.

La méthode doit être enseignable : en fait elle doit être compréhensible sans beaucoup de grandes difficultés à tout concepteur et autre personne ayant une certaine qualification; son formalisme d'abstraction ne doit pas être difficile à appréhender. Son modèle doit constituer un bon "vocabulaire" pour les réunions concepteurs-utilisateurs.

E / APPLICABILITE GENERALE :

Autant que possible une bonne méthode devrait traiter de classes de projets aussi différents les uns des autres, elle ne devrait pas trop se spécialiser; de par la valeur de ses concepts elle pourrait être adaptée à un autre environnement sans trop de difficultés.

F / EVOLUTIVITE :

La méthode doit répondre autant que possible aux évolutions technologiques, et permettre une maintenance aisée de ses logiciels, elle doit être ouverte pour intégrer à un niveau donné d'autres méthodes, d'autres idées, d'autres besoins; cette évolutivité doit se faire sans qu'il y ait un bouleversement important des principes de base.

6 / MOYENS UTILISES :

6-1 CONCEPTS :

Les concepts du modèle de spécification doivent être clairs, explicites, intégrer les aspects statiques et les aspects dynamiques, le temps, avoir un bon pouvoir de représentation, et ne pas être trop difficiles à comprendre et à assimiler. Les schémas produits grâce à ces concepts doivent avoir un bon pouvoir de repésentation qui puisse faciliter le dialogue entre concepteurs et utilisateurs.

En plus de leur pouvoir de représentation les schémas produits (aux trois niveaux de description) doivent être complets, détaillés et bien documentés afin de ne pas autoriser les retours en arrière pour remettre en cause les solutions définies antérieurement.

Pour que le schéma conceptuel obtenu soit minimal, fidèle, cohérent, complet, souple et indépendant des usages, le modèle doit présenter un certain nombre de caractéristiques :

1) le modèle doit être formel :

Le modèle doit être défini formellement afin que la représentation minimale soit cohérente et complète. Un formalisme préexistant doit permettre de définir les concepts du modèle utilisé. A défaut un ensemble de règles constituant un système formel peuvent permettre de définir les concepts.

2) le modèle doit être orienté "conception" :

Pour garantir le caractère d'indépendance du schéma conceptuel par rapport à l'usage de ce schéma, le caractère formel du schéma n'est pas suffisant. Par exemple les réseaux de PETRI modélisent les traitements qui intègrent la dynamique mais sont mal adaptés à la conception du schéma conceptuel tel que nous l'avons défini [FOU 82].

L'orientation "conception" signifie aussi que les "concepts du modèle doivent être tels qu'ils aident le concepteur à extraire de l'analyse de la réalité ou de l'énoncé du problème à résoudre, le noyau minimum qui la ou le caractérise " [FOU 82]. Un modèle peut posséder plus ou moins cette qualité.

6-2 LES LANGAGES :

Les langages de spécification proposés (aussi bien au niveau de la conception, de l'implémentation que de la conception des programmes) doivent former un tout intégré.

Pour satisfaire les qualités attendues d'un schéma conceptuel un langage de spécification doit posséder les qualités suivantes : [THI 85] :

- permettre une représentation abstraite, formelle, structurée de la réalité organisationnelle (dans ses aspects statique et dynamique),

- être de haut niveau et aussi abstrait que possible,

- offrir la possibilité d'une communication compréhensible et lisible ce qui veut dire que le langage doit :

. être prédicatif : s'appuyer sur des propositions et expressions simples,

. être déclaratif : être non procédural et donc proche du langage des utilisateurs; la notion d'algorithme doit leur être transparente,

. reposer sur un ensemble de constructions fréquentes en informatique de gestion telles que les itérations et les conditionnelles et surtout les accès aux données,

. avoir une syntaxe qui permette la description cohérente d'une spécification et les affirmations que l'on retient de l'univers du discours,

. faciliter le passage à une automatisation c'est à dire que, tout en étant un langage de haut niveau, autoriser une traduction dans un langage cible donnant lieu à une solution technique viable.

Les langages graphiques quant à eux doivent être facilement assimilables à des utilisateurs non avertis et permettre une bonne spécification du dialogue utilisateur-machine ; ils doivent constituer un bon outil de communication entre utlisateurs et concepteurs.

En plus de cet aspect "communication", les langages graphiques doivent permettre une représentation aussi fidèle que possible de la statique et de la dynamique du SI.

6-3 LES DEMARCHES :

Elles doivent être le complément naturel des modèles et des langages dans la mesure où une bonne méthode de conception doit proposer à tous les niveaux des démarches rigoureuses appropriées; elles doivent être facilement assimilables et permetre des validations sans imposer un canevas trop lourd à l'utilisateur et une multitude de documents à remplir.

6-4 LES OUTILS :

Autant que possible une bonne méthode de conception doit proposer des outils à toutes les étapes du cycle de vie non seulement pour améliorer la qualité de la conception et de la réalisation mais aussi pour permettre une validation aisée aussi bien de la part des concepteurs que de la part des utilisateurs.

En tant que supports de la conception on peut trouver des outils d'aide à la conception du SI, des outils de contrôle et de validation du schéma conceptuel, des outils d'aide à la spécification, des outils de documentation (dictionnaire des données, de documentation du schéma conceptuel...), des outils de contrôle (des informations

entrées, de la validité des contraintes d'intégrité).

Ces outils ne doivent pas être proposés en ordre dispersé mais former un tout cohérent. Ils doivent assumer un certain nombre de rôles :

- l'outil doit reposer sur un canevas méthodologique [THI 85] ainsi que sur des logiciels conviviaux tels que les éditeurs vidéo et l'utilisation des touches de fonction. L'aide que fournissent ces outils se matérialise par un "pilotage" de la conception ce qui permet par exemple d'effectuer certains contrôles des descriptions fournies ou de générer certains éléments de description. Ces outils peuvent aider le concepteur dans sa réflexion en le faisant intervenir en cas de conflit ou en effectuant à sa place certains choix possibles. Dans tous les cas de figure le concepteur doit rester maître de sa conception,

- contrôler les spécifications entrées, permettre les retours en arrière en cas d'erreur, intégrer les nouvelles spécifications dans le système existant. En particulier l'outil doit déclencher les procédures de vérification introduites lors de la définition des schémas de type objet, événement, opération et se livrer aux contrôles syntaxiques de type classique sur les textes d'énoncés [THI 85],

- faciliter la documentation des concepteurs en faisant des éditions paramétrées ou non du schéma conceptuel selon un format prédéfini. L'interrogation portant sur la structure du SI peut se référer à l'instant de la demande ou à un instant quelconque du passé.

Ces outils doivent constituer un véritable environnement d'A.G.L devant être composé entre autres : [THI 85]

- d'un environnement intégré de spécification comprenant des outils utlisant les mêmes structures de données, les mêmes commandes, les mêmes éditeurs; ces outils travaillant de façon interactive,

- d'un environnement de vérification pour effectuer les contrôles des spécifications entrées et permettre les aller-retours (utilisateur-machine) grâce à un outil convivial,

- d'un environnement de documentation pour avoir des informations sur les possibilités du système (format d'une commande, problèmes de syntaxe,...) et des informations sur les descriptions effectuées lors de la spécification,

- d'un environnement de pilotage qui permet de dire au concepteur ce qu'il doit faire comme actions à chaque étape ou comme choix. Cet environnement définit la marche à suivre, la méthode préconisée pour mener la conception.

L'objectif final étant, rappelons-le, de constituer un atelier de conception-réalisation de SI.

3-4 PERSPECTIVES :

Introduction :

Rappelons que nous avons, dans une partie introductive, défini ce qu'est un SI et montré la complexité de ce concept, ce qui nous a conduit à définir un cadre de comparaison de diverses méthodes de conception de SI qui met surtout l'accent sur l'étude du niveau conceptuel, et dont les thèmes étudiés sont les niveaux d'abstraction couverts par telle ou telle méthode, les moyens (modèles conceptuels de SI : orientés données ou orientés traitement), les langages de spécification, les démarches, les outils utilisés durant la phase de conception, et les étapes couvertes par la méthode.

A la fin de l'analyse effectuée ci-dessus selon notre cadre de comparaison, un tableau de synthèse a été élaboré.

En conclusion de tout le travail effectué nous avons dressé un bilan des points forts et des points faibles de chaque méthode, ce qui nous a permis d'inventorier un certain nombre de caractéristiques d'une bonne méthode de conception de SI.

Nous reconnaissons que notre bibliographie date un peu, ceci s'explique en partie par nos conditions de travail (éloignement très grand vis à vis du CRIN et des centres de recherche, documentation exhaustive difficile à obtenir, ...) qui nous empêchent de suivre au jour le jour l'actualité informatique. Cependant nous avons la ferme intention de tenir ce document à jour et de l'enrichir de façon permanente.

PERSPECTIVES :

- ce document sera d'une très grande utilité pour l'enseignement à l'IAI dont l'un des axes fondamentaux est l'étude et la mise en place des systèmes d'information. Les étudiants comme les professeurs pourront s'en servir à titre de documentation pour enrichir leurs connaissances en matière de conception de SI,

- il sera un guide précieux pour appréhender à leur juste valeur les problèmes de conception qui sont d'une complexité certaine,

- il pourra servir comme l'une des multiples références dans le cadre de la recherche appliquée vers laquelle a décidé de s'orienter l'IAI depuis le conseil d'administration de OUAGADOUGOU (1987),

- son enrichissement et sa mise à jour permanente pourront faire l'objet d'un travail collectif des professeurs de l'IAI ceci en relation directe avec divers centres de recherche comme le CRIN de NANCY, et les sociétés de services ou autres sociétés utilisatrices de ces méthodes.

- la diffusion de ce document au niveau des divers centres informatiques (publics ou privés) des pays membres de l'IAI permettra à ces derniers de s'orienter dans le choix des méthodes de conception appelées à les aider et guider et aider dans la mise en place de SI fiables et évolutifs, ce qui à l'heure actuelle constitue leur préoccupation majeure.
L'IAI remplira ainsi son rôle de diffuseur des connaissances informatiques; un échange fructueux pourra être ainsi entamé et continuellement enrichi avec nos divers centres de calcul.

Four notre part, tout en restant à l'écoute de toutes les avancées dans le domaine de la conception des SI, nous poursuivrons nos investigations afin de mettre à jour de façon permanente l'etude que nous venons de mener.

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

ABR 74 ABRIAL J.R "the temporal dimension of information modelling " Proc of IFIP TC2 WC NICE 1977.

ABR 78 ABRIAL J.R. " A specification language Z " séminaire de HAUTE-NENDAZ MARS 1978.

ABR 80 ABRIAL J.R. , SCHUMANN S.A. , MEYER B. , " Z : a specification language" Proc. Summmer School on the construction of programs BELFORT , CAMBRIDGE University Press 1980.

ADI 85 ADIBA M. " Notion de temps dans les bases de données généralisées " . Journées Bases de Données Avancées SAINT PIERRE DE CHARTREUSE MARS 1985.

ANT 81 DE ANTONELLIS V., ZONTA B. "Modelling Events in Database Application Design " Proc. congrès VLDB 1981.

ANS 75 ANSI/X3/SPARC " Interim Report from the Study Group on Data Base Management Systems " Bulletin of ACM vol 7 num. 2 1975.

ASH 82 ASCHIM F., MOSTVE B.M., " IFIP WG 8.1 Case SOLVED Using SYSDOC and Systemator" conference CRIS 1 IFI 82.

AX 84 OPEN - CRIS conference AFCET PARIS 1984.

BACK 78 BACKUS J. " Can Programming be liberated from the VON NEUMANN styles ? A fonctional style and its algebra of Programs " CACM AOUT 1978.

BAU 78 BAUER F.L. , PEPPER P. , WOSSNER H. "Notes on the projet CIP : outline of a transformation System ".

BEN 76 BENCI G., BODART F., BOGAERT H., CABANES A. , "Concepts for the design of a conceptual schema " Proc of IFIP TC2 WC FREUNDENSTADT 1976.

BEN 79 BENCI G., COLETTE ROLAND "Bases de données : conception canonique pour une réalisation extensible" SCM PARIS 1979.

BOD 79 BODART F., PIGNEUR Y., " A model and a language for functional specifications and evaluation of information system dynamics" IFIP TC 8WG on "formal models and pratical tools for information system design" OXFORD 1979.

BOD 83 BODART F. et al " Evaluation of CRIS-1 I.S developments

methods using a three cycles framework "
conference CRIS-2 IFI 83.

BOD 84 BODART F., PIGNEUR Y. " IDA : une méthode de conception de SI" conference CRIS-3 IFI 84.

BOUZ 83 BOUZEGHOUB MOKRANE INRIA rapport de recherche n° 258 "Une synthèse des méthodes et des outils d'aide à la conception des SI".

BOY 74 BOYCE R.F et al. "Specifying queries as relational expressions : SQUARE " Proc. IFIP TC2 WC NORTH HOLLAND 1974.

BRA 76 BRACCHI G et al. "Binary logical associations in data modelling" Proc IFIP TC2 WC Freundenstadt 1976.

BRA 79 BRACCHI G., FURTADO A., PEGALATTI G., "Constraint Specification In evolutionary data base design"
IFIP TC 8 WC on "Formal model and pratical tools for information system design" OXFORD 1979.

BRE 79 BREUTMAN B., FALKENBERG E., MAUER R. " CSL : a language for defining conceptual schemas"
IFIP TC2 WC on data base architecture VENISE 1979.

BRO 80 BRODIE M.L " Data abstraction, databases and conceptual modelling" Proc. VLDB 1980.

BRO 82 BRODIE M.L, SILVA E
" Active and Passive Component Modelling ACM/PCM"
conference CRIS 1 IFI 82.

BUB 77 BUBENKO J.A " The Temporal Dimension of Information Modelling " Proc of IFIP TC2 WC NICE 1977.

CABA 76 CABANES A. "Un modèle d'accès standard dans les SGBD" bulletin de liaison du Club Banques de données IRIA, NO 16 1976.

CAV 79 CAVARERO J.L "LAPAGE : un modèle et un outil d'aide à la conception des SI " Thèse d'état NICE 1979.

CHA 74 CHAMBERLIN D.D, BOYCE R.F "SEQUEL : a Structured English Query Language" Proc. ACM SIGMOD workshop on Data Description, Access and Control 1974.

CHA 83 CHABRIER J.J., CHABRIER J. "VEGA : un système interactif de spécification algébrique avec erreurs" Rapports CRIN 83-R-036.

CHEN 76 CHEN P. " The Entity-Relationship model towards a unified view of data " ACM transactions on DATA BASE SYSTEMS VOL 1 MARS 1976 PP 9-36.

CHEN 77 CHEN P. "Design and Performance Tools for Database Systems (Proceed of 3rd VLDB conference IEEE 1977)".

CHR 80 CHRISTMENT C., ZURFLUH C., "Bases d'information

généralisées : projet BIG modèle agrégatif, langage de manipulation et interface multi-média" Journées Bases de données Avancées SAINT PIERRE DE CHARTREUSE MARS 1985.

COD 70 CODD E.F " A relational model of data for large shared data banks" CACM vol 13 n° 6 1970.

COD 71 CODD E.F "A data base sublanguage formed on the relational calculus" Proc ACMSIGDIDET Workshop on data description, access and control 1971.

COD 74 CODD E.F "Recent investigations into relational data base systems" Proc IFIP congress STOCKHOLM 1974.

CRE 75 CREHANGE M. " Description formelle, représentation, interrogation des informations complexes Système PIVOINES " Thèse d'état NANCY 1975.

CRIS 1-2-3 "Comparative Review of Information Systems Design Methodologies" NOORDWIJKERHOUT PAYS BAS NORTH HOLLAND Publication 1982.

CRIS-3 " Concevoir vos S.I : une sélection internationale de méthodes de conception : les choix de l'IFIP" PARIS OCTOBRE 1984.

DEL 73 DELOBEL C. "Contributions théoriques à la conception, et à l'évaluation d'un SI appliqué à la gestion " Thèse d'état GRENOBLE 1973.

DEM 85 DEMOLOMBE R. "STREL: une extension du modèle relationnel pour représenter et manipuler les objets structurés" Journées Bases de Données Avancées SAINT PIERRE DE CHARTREUSE MARS 1985.

DON 75 DONZEAU-GOUGE et al "A structure oriented Program Editor : a first step towards computer-assisted programing" International Computing Symposium North Holland Publication 1975.

DUB 84 DUBOIS E. "Cadre et méthode de spécification de SI fondés sur les types de données " Thèse Docteur Ingénieur NANCY 1 CRIN AVRIL 1984.

DUF 80 DUFOURD J.F "maquettes pour évaluer les S.I des organisations" Thèse d'état NANCY 1 CRIN 1980.

FAL 75 FALKENBERG E. "Design and Application of a natural language oriented data base language" adv. course on data base languages and natural language processing, FREUNDENSTADT, GERMANY, AUG 75.

FIN 79 FINANCE J.P "Etude de la construction de programmes: méthodes et langages de spécification et de résolution de problèmes" Thèse d'état NANCY 1 1979.

FLO 77 FLORY A. "Un modèle et une méthode pour la conception logique d'une base de données" Thèse d'état LYON 1977.

FOR 62 FORRESTER J.W "Industrial dynamics"
WILEY AND SONS 1962.

FOU 82 FOUCAULT ODILE "Modèle et outil pour la conception des systèmes d'information dans les organisations, projet REMORA " Thèse d'état NANCY 1 CRIN 1982.

GANE 79 GANE C., SARSON T. "Structured System Analysis"
ENGLEWOOD CLIFFS NJ : PRENTICE HALL 1979.

GEN 88 GENIE LOGICIEL revue n° 11.

GUS 82 GUSTAFSON M.R., KARLSSON T., BUBENKO J.R
"A Declarative Approach to conceptual information modeling" conference
CRIS 1 IFI 82.

GUY 84 GUYARD J., JACQUOT J.P. " MAIDAY : an environment for suided programming with a definitional language" 7th. conference on SOFTWARE ENGINEERING ORLANDO FLORIDE
USA MARS 1984.

IFI 82 IFIP WG 8.1 working conference on
"Comparative Review of Information Systems Design Methodologies".
NOORDWIJKERHOUT PAYS-BAS NORTH HOLLAND
PUBLICATION 1982.

IFI 83 IFIP WG 8.1 Working conference on
"Information Systems Design Methodologies : a feature analysis"
YORK NORTH HOLLAND PUBLICATION 1983.

IFI 84 IFIP 84 "Concevoir vos S.I., une sélection internationale de méthodes de conception : le choix de l'IFIP"
PARIS OCTOBRE 1984.

ISO 81 ISO TC 97/SC5/WG3 "Preliminary report concepts and terminology for the conceptual schema".
JJ VAN GRIETHUYSEN (ed.) FEVRIER 1981.

KEN 75 KENNEDY K., SCHWARTZ J. "An introduction to the set Theory language SETL"
Computers and Mathematics with Applications
1(97) 1975.

KENT 76 KENT W. "Describing information (not data reality)"
Technical Report TR03012 1976.

KEN 77 KENT W. "Entities and relationships in information"
Proc IFIP TC2 WC NICE 1977.

KNU 82 KNUTH E. et al. "SDLA, System Descriptor and Logical Analyser" Conference CRIS 1 IFI 82.

LAN 66 LANGEFORS B "theorical analys of information systems"
LUND 1966, FOURTH EDITION 1973.

LANG 73 LANGEFORS B., "theorical analys of information

systems"
student litterature 4é edition 1973
SWEDEN.

LAU 76 LAURIERE J.L. "Un langage et un programme pour résoudre des problèmes combinatoires" Thèse d'état PARIS VI 1976.

LE 84 LE F., TARDIEU H., TABOURIER Y. "La méthode MERISE" Conférence CRIS3 IFI84.

LEO 81 LEONARD M., LUONG B.T "Information Systems approach Integrating Data and Transactions" Proc. VLDB CANNES 1981.

LEV 80 LEVESON N.G "Applying Behaviour Abstraction to Information System Design and Integrity" Thesis University of LOS ANGELES 1980.

LIS 77 LISKOV B et al. "abstraction mechanisms in CLU " CACM 20 (8) AOUT 1977.

LUN 82 LUNDBERG M. "The ISAC Approach to specification an Information Systems and its application to the organisation of an IFIP working conference" conference CRIS 1 IFI82.

MAC 82 MAC DONALD I.G., PALMER I.R, "System Development in a shared Data Environment the D2S2 Methodology " Conference CRIS 1 IFI 82.

MER 79 MERISE "méthode de définition d'un SI" CII Ministère de l'Industrie JUIN 1979.

MIN 79 MINOT R. "ATM : un système de fabrication des programmes basé sur les concepts de modularité et de types abstraits" Thèse 3ème cycle NANCY 1979.

MOI 77 LE MOIGNE JEAN LOUIS 4è trimestre 77 "THEORIE DU SYSTEME GENERAL" PUF.

NIC 79 NICOLAS J.M "contributions à l'étude théorique des bases de données. Apports de la logique mathématique" Thèse d'état Université PAUL SABATIER TOULOUSE 79.

NIJ 77 NIJSSEN G.M "Current issues in conceptual schema concepts" Proc. OF IFIP TC2 WC NICE 1977.

PAI 79 PAIR C. "la construction de programmes" RAIRO Informatique 15 (2) 1979.

PEC 75 PECOUD "MACSI" Thèse d'état GRENOBLE 1975.

PEL 86 PELLAUMAIL P. "La méthode AXIAL, conception d'un SI" AD. ORGANISATION 86.

PET 77 PETERSON J.L "PETRI NETS" ACM COMPUTING SURVEYS 9(3) 1977.

PET 80 PETRI C.A "Introduction to general net theory"
Lecture notes in Computer Sciences num. 84
Springer VERLAG 1980.

QUE 84 QUERE A. et al. "SPES : un système pour spécifier
et transformer" CGL2 NICE JUIN 1984.

REIX 74 REIX R. "L'analyse en Informatique de gestion"
Tome 2 DUNOD Economie 1974.

RIC 82 RICHTER G., DURCHHOLTZ R.,
"IML-Inscribed High-Level Petri-Nets"
Conference CRIS 1 IFI 82.

RID 78 RIDDLE W. et al "DREAM : a Software Design Aid Sysytem"
Proc. of 3rd. JERUSALEM conference on Information
Technology
NORTH HOLLAND Publication 1978.

ROL 79 ROLLAND C., BENCI G. "Bases de données, conception
canonique pour une réalisation extensible"
SCM PARIS 1979.

ROL 82 ROLLAND C., RICHARD C. "The REMORA methodology for
I.S design and Management"
conference CRIS IFI 82.

ROL 87 ROLLAND C., FOUCAULT O., BENCI G.,
"Conception des systèmes d'information la méthode
REMORA" Editions EYROLLES OCT. 1987.

RZE 82 RZEVSKI G. et al.
"The Evolutionary Design Methodology applied to I.S
IFIP Case" Conference CRIS 1 IFI 82.

SHIP 81 SCHIPMAN D.W "The Functional Data Model and the Data
Language DAPLEX (ACM TODS V6, N1, MARCH 81).

SAL 80 SALES A. "Utilisation de la notion de type abstrait
générique en bases de données relationnelles"
Actes du congrès AFCET NANCY 1980.

SCH 77 SCHMID J.W "Some High Level Language Constructs for Data
of Type Relation" ACM TODS 2(3) 1977.

SEN 75 SENKO M.E "Specification of stored data structures and
desired output results in DIAM II with FORAL
Proc first conf VLDB 1975.

SMI 77 SMITH J.M., SMITH D.C.P "Data base Abstraction :
Aggregation and Generalization"
ACM TOD2(2) JUIN 1977.

SOL 82 SOLVBERG A. "A draft Proposal for Integrating System
Specification Models"
Conference CRIS 1 IFI 82.

SUN 73 SUNDSREEN B. "An Infological approach to data bases"
PHD Thesis STOCKHOLM 1973.

TAR 74 TARDIEU H. "The individual model"
IFIP TC2 WC NAMUR 1974.

TAR 83 TARDIEU H., ARNOLD R., COLETTI R.,
"La méthode MERISE, principes et outils"
Editions d'organisation 1983.

TEI 81 TEITELBAUM T., REPS T. "The CORNELL Program Syntheziser :
A Syntax Directed Program Environment"
CACM 24(9) 1981.

THI 76 THIERRY O. "L'aide à la conception dans le projet REMORA"
Thèse de troisième cycle NANCY 1976.

THI 85 THIERRY O. "LASSIF, Langage de Spécification de Système
d'Information, logiciel d'Aide à la spécification des
systèmes d'information" Thèse d'état
NANCY 1 CRIN 1985.

VEL 84 VELEZ F. "Un modèle et un langage pour les bases de
données généralisées : projet TIGRE "
Thèse de Docteur Ingénieur INP GRENOBLE 1984.

VERRI 75 VERRIJN-STUART A.A
"Information algebras and their uses"
Management Datamatics vol 4 N.5 1975 PP 187-197.

VER 82 VERHEIJEN G.M.A., VAN BEKKUM J.
"NIAM : an Information analysis method"
conference CRIS 1 IFI 82.

WAR 77 WARREN D.H.D., PERREIRA L.M.
"PROLOG : the language and its implementation compared
with LISP" Proc of the ACM Symposium on A.I and
Programmming Languages
ROCHESTER USA 1977.

WAR 81 WARREN D.H.D "Efficient Processing of Interactive
relational data base queries expressed in LOGIC"
Proc of the Seventh International conference
on VLDB CANNES 1981.

WAS 78 WASSERMAN A.I. "A software Engineering view of Data
Base Management"
Proc of VLDB BERLIN 1978.

WAS 81 WASSERMAN A.I et al "Revised Report on PLAIN"
ACM SIGPLAN 16(5) 1981.

WAS 82 WASSEMAN A.I "The User Software Engineering
Methodology : An overview "
Conference CRIS 1 IFI 82.

WIR 71 WIRTH N. "The Programming Language PASCAL"
Acta Informatica 1 1971.

WIR 76 WIRTH N., "Algorithms + Data structures = Programs"
Prentice-Hall, Englewood Cliffs, NEW JERSEY 1976.

YOUNG 50 YOUNG J.W, KENT H.K

"Abstract formulation of data processing problems,
the journal of industrial Engineering"
NOVEMBER-DECEMBER 1958.