

Le Dialogue de validation d'un schema conceptuel des donnees

Laurent Karsenty

► **To cite this version:**

Laurent Karsenty. Le Dialogue de validation d'un schema conceptuel des donnees. [Rapport de recherche] RR-1551, INRIA. 1991. inria-00075010

HAL Id: inria-00075010

<https://hal.inria.fr/inria-00075010>

Submitted on 24 May 2006

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

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

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1551

Programme 3
Intelligence artificielle, Systèmes cognitifs et
Interaction homme-machine

LE DIALOGUE DE VALIDATION D'UN SCHEMA CONCEPTUEL DES DONNEES

Laurent KARSENTY

Novembre 1991



Programme 3

Intelligence Artificielle, Systèmes Cognitifs et Interaction Homme-Machine

**LE DIALOGUE DE VALIDATION
D'UN SCHEMA CONCEPTUEL DES DONNEES**

Structure du dialogue et connaissances mises en œuvre

**THE VALIDATION DIALOG
FOR A DATABASE CONCEPTUAL SCHEMA**

Dialog structure and knowledge involved

Laurent Karsenty

Novembre 1991

TABLE DES MATIERES

1 - INTRODUCTION.....	1
1.1 La problématique de l'étude.....	1
1.2 L'étape de validation dans le processus de conception d'une Base de Données.....	2
1.3 Le Schéma Conceptuel des Données (SCD).....	5
1.4 Apports théoriques des études sur l'activité de conception.....	8
2 - METHODOLOGIE DE L'ETUDE.....	11
2.1 Le terrain.....	11
2.2 Méthode de recueil des données.....	12
3 - RESULTATS.....	13
3.1 Observations générales.....	13
3.1.1 Déroulement global d'une réunion de validation.....	13
3.1.2 Temps variable.....	13
3.1.3 Données et Traitements.....	14
3.1.4 Description partielle du SCD.....	15
3.1.5 Différents types de dialogue.....	19
3.1.6 Les modifications apportées à la solution.....	21
3.1.7 L'issue d'une réunion de validation.....	21
3.2 Les connaissances nécessaires à la validation du SCD.....	22
3.2.1 La référence aux éléments du SCD.....	23
3.2.1.1 Comparaison entre la référence simple et le discours sur le SCD.....	23
3.2.1.2 Les différents types de référence aux éléments du SCD.....	26
3.2.2 Les connaissances ajoutées.....	29
3.2.2.1 Méthode d'analyse des connaissances ajoutées.....	29
3.2.2.2 Classifications des connaissances ajoutées.....	30
3.2.2.3 Un deuxième niveau de référence.....	34
3.2.2.4 L'anticipation des demandes d'information et/ou de justification.....	37
3.2.2.5 Discussion.....	38
4 - CONCLUSION.....	41
4.1 Vers une représentation adaptée aux utilisateurs.....	41
4.2 Eléments de spécification d'un système d'aide à la conception de BD.....	43
4.3 Perspectives de recherches.....	45
BIBLIOGRAPHIE.....	47
ANNEXE : La structure du dialogue de validation.....	50

1 - INTRODUCTION

1.1 La problématique de l'étude

Cette étude analyse une des étapes de la conception de Base de Données (BD), l'étape de validation du Schéma Conceptuel des Données (SCD) auprès des futurs utilisateurs.

Deux motivations sont à la base de cette étude :

1. Les méthodologies récentes en conception de BD mettent de plus en plus l'accent sur la nécessité d'interagir avec l'utilisateur dès les premières phases d'un projet. La communication entre concepteur et utilisateurs va alors s'appuyer sur un certain nombre de documents, dont le SCD, description abstraite et formalisée de la BD visée. Olle & al. (1990) insistent sur la nécessité d'établir un SCD *compréhensible* et *validable* par les utilisateurs non-initiés en conception de BD, sans pour autant préciser les éléments dont dépendent la compréhension et le jugement de ces derniers. On sait en outre que les futurs utilisateurs ressentent un certain nombre de difficultés pour utiliser ce type de représentations, même si les formalismes récents ont été conçus en partie pour résoudre ce problème. Il faut remarquer qu'aucun d'entre eux cependant n'est fondé (à notre connaissance) sur des observations en situations naturelles dans lesquelles un ou plusieurs utilisateurs seraient placés face à un SCD. L'étude présente vise à mieux spécifier les besoins en information requis pour la compréhension et le jugement des utilisateurs, en se basant sur l'analyse de dialogues de validation d'un SCD, entre un concepteur et plusieurs utilisateurs.
2. D'autre part, on envisage de plus en plus la possibilité pour un utilisateur de concevoir sa propre BD, assisté par un outil d'aide. Si l'on se restreint à l'étape de validation, et en considérant qu'un tel outil serait capable de proposer des solutions, on peut se poser les questions suivantes :
 - quelles sont les actions à engager et les types d'interactions à prévoir ?
 - quel serait le type de description de la solution le plus adapté à la collaboration entre l'homme et la machine? Il n'est pas évident, étant donné les moyens nouveaux que fournit l'ordinateur, qu'une description type SCD classique soit la plus adaptée.
 - doit-on prévoir un type de description de la solution, ou plusieurs (autrement dit, doit-on soumettre une solution possible toujours de la même façon ?)

Cette série de questions sera abordée en se basant sur l'idée que le dialogue naturel entre un concepteur expert et plusieurs utilisateurs fournit un modèle de la communication entre le système d'aide visé et son utilisateur. Ce type de situation de dialogue a été recueilli et analysé pour répondre aux problèmes soulevés ci-dessus.

Dans les parties suivantes, on :

- précise les caractéristiques de l'étape de validation et d'un SCD basé sur un formalisme Entité-Association,
- souligne quelques notions extraites de différentes études sur les activités de conception,
- décrit la méthodologie qui a été utilisée,
- présente les résultats de l'analyse de dialogues de validation.

1.2 L'étape de validation dans le processus de conception d'une BD

Il faut tout d'abord souligner que la conception d'une BD intervient le plus souvent dans un projet plus large d'informatisation d'un service, ou de réalisation d'un système d'information. C'est pourquoi il est nécessaire en premier lieu de situer la conception d'une BD à l'intérieur d'un tel projet.

• *Le concept de Système d'Information (SI)*

Ce concept recouvre en fait deux notions (Tardieu & al., 86, p.37) :

- la réalité de l'organisation se transformant et mémorisant des informations, notion qui apparente le système d'information à un *objet naturel* ;
- le système, *objet artificiel*, construit par l'homme pour représenter les actions, la communication et la mémorisation dans l'organisation.

Le point intéressant soulevé par ces mêmes auteurs est que la conception d'un système d'information se situera toujours entre l'analyse de l'organisation et de son système d'information et la conception d'un système informatisée ayant pour but de représenter cette organisation pour en améliorer l'efficacité.

Dans la suite de ce rapport, le terme "système d'information" (SI) sera utilisé pour désigner l'objet artificiel.

• *Les trois composants d'un SI*

Un SI est composé :

- d'un module de dialogue qui va permettre de prendre en compte les requêtes des utilisateurs (demande d'information, demande de mise à jour, etc.) ;
- d'un ensemble de traitements qui vont permettre de traiter ces requêtes et d'accéder aux informations;
- et de différents modules de stockage, parmi lesquels on peut trouver une BD.

• *Définition d'une BD*

Une BD est un ensemble de données reliées, représentant une partie de la réalité organisationnelle, et destiné à un public déterminé, intéressé par son contenu (Elmasri & Navathe, 89).

• *Le cycle de vie de la conception d'une BD*

Trois grandes étapes ponctuent ce processus (cf fig.1) :

- 1) la définition des besoins réalisée à partir d'interviews avec les personnes manipulant l'information dans le domaine d'application (et considérées comme les futurs utilisateurs de la BD) ;
- 2) la modélisation de la BD, décomposée en modélisation des données et modélisation des traitements. La modélisation des données est elle-même décomposable : à partir du schéma conceptuel des données, le concepteur pourra déduire un schéma logique, puis physique, ce qui consiste à introduire progressivement un certain nombre de contraintes liées au fonctionnement du système informatique.
- 3) l'implémentation de la BD. Dès que la BD est opérationnelle, un certain nombre de tests de performance sont faits. Ceux-ci peuvent alors entraîner des modifications si les performances produites ne correspondent pas aux performances attendus. Ces tests peuvent dans certains cas amener le concepteur à revoir son schéma conceptuel, voire à redéfinir les besoins avec les utilisateurs. Le cycle de vie du processus de conception peut donc reprendre même durant les phases les plus avancées.

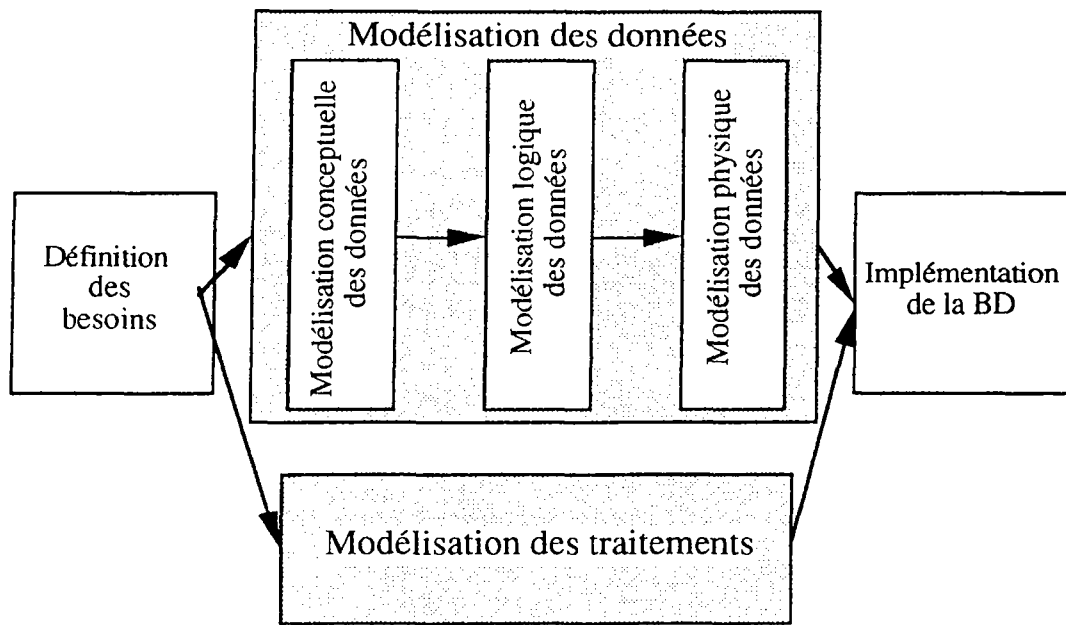


Figure 1 : Le cycle de vie du processus de conception d'une BD

L'activité de modélisation conceptuelle, à l'origine du schéma conceptuel des données, peut elle-même être décomposée en quatre phases (Rolland & al., 88) :

- 1) *l'analyse*, qui demande à identifier et énumérer des classes de phénomènes à prendre en compte.
- 2) *la conceptualisation*, qui correspond à la recherche d'une représentation normative, le schéma conceptuel.
- 3) *la spécification*, qui amène à décrire le schéma dans tous ses détails. Cette phase conduit à une représentation graphique et une représentation en langage déclaratif.
- 4) *la validation*, qui se propose, d'une part de confronter le schéma conceptuel aux besoins en information des utilisateurs, et d'autre part de réaliser des contrôles formels assurant le passage aux phases suivantes.

T.W. Olle et ses collaborateurs (1990) mettent alors l'accent sur la nécessité d'établir une représentation *compréhensible* et *validable* par les utilisateurs, ces propriétés devant faciliter le dialogue entre concepteur et utilisateurs. Le formalisme choisi doit prendre en compte ces exigences.

C'est à cette étape que nous nous intéressons ici.

Cette définition de la tâche permet de remarquer que la validité d'une solution revêt au moins deux aspects: l'adéquation aux besoins des utilisateurs, et le respect des règles de formalisation. Si pour le premier, on comprend aisément en quoi le jugement des utilisateurs est important, cela ne va pas de soi pour le second aspect, pour lequel un utilisateur n'a pas la compétence requise.

On tentera donc notamment de déterminer si le dialogue de validation entre un concepteur et des utilisateurs aborde les deux aspects, ou si il se borne à vérifier la prise en compte des besoins.

1.3 Le Schéma Conceptuel des Données

• *Définition*

Le SCD fournit une description conceptuelle et formalisée des informations manipulées dans le domaine d'application et destinées à être stockées dans la BD. Il peut donc être considéré à la fois comme un modèle de la réalité organisationnelle, et comme la définition prescriptive des données du futur système (Kangassalo, 88).

• *Les modèles conceptuels*

Les qualités d'un formalisme de représentation d'un schéma conceptuel sont :

- sa capacité à capter le maximum de connaissances exprimées dans le domaine d'application ;
- l'adéquation entre les structures conceptuelles qu'il définit et les structures habituellement utilisées par les acteurs impliqués dans la conception d'une BD (au minimum, utilisateurs et concepteur) et dans sa maintenance. Cette adéquation est nécessaire pour favoriser la communication avec les acteurs impliqués dans un projet, et non-initiés aux langages de programmation ;
- il doit permettre de vérifier la cohérence des données, notamment lors des opérations de maintenance de la BD. Pour cela, il doit fournir un certain nombre de concepts qui permettent de représenter des contraintes sur les valeurs.

Une autre exigence, peu exprimée cependant dans les ouvrages consacrés aux modèles conceptuels, est la possibilité d'implémenter un outil d'aide à la conception basé sur modèle conceptuel donné (King & McLeod, 85). Cet objectif explique dans une certaine mesure les limites d'expression des modèles conceptuels.

Les différents formalismes actuels peuvent être distingués selon qu'ils considèrent séparément la modélisation des données et la modélisation des traitements - par exemple, le modèle relationnel (Codd, 70), le modèle Entité-Association (Chen, 76) ou le modèle NIAM (Verheijen & al., 82) - ou selon qu'ils mêlent les deux aspects - par exemple, le langage CSL (Breutmann & al., 79) ou le modèle REMORA (Rolland & al., 88).

Les partisans de la première approche considèrent qu'une modélisation conceptuelle des données indépendante de celle des traitements permet de découvrir une sémantique des données plus appropriée à la réalité, la prise en compte trop précoce des traitements introduisant des distorsions dans l'interprétation des informations à mémoriser.

Les partisans de la seconde approche considèrent quant à eux, qu'il est préférable de représenter données et traitements en même temps, afin de favoriser une plus grande intégrité des données (lors des opérations de maintenance notamment) et une meilleure adéquation aux représentations des utilisateurs. En effet, les concepteurs remarquent que ceux-ci produisent difficilement des énoncés séparant les deux aspects, et ne semblent pas pouvoir envisager l'un sans l'autre.

Dans les situations de validation observées, le concepteur avait élaboré une représentation des données indépendamment de celle des traitements. La validation ne devait porter que sur le SCD, présenté à différents groupes d'utilisateurs. Ces conditions d'observations nous donnaient l'opportunité de vérifier les observations des concepteurs, et d'évaluer l'effet potentiel de cette distance entre d'une part une représentation externe et formalisée de la structure des informations, i.e. le SCD, et d'autre part un ensemble de représentations propres aux utilisateurs mêlant actions et informations.

Chaque formalisme définit un ensemble de concepts de base et de règles de structuration. Le modèle Entité-Association, l'un des plus utilisés, est détaillé ci-dessous.

• *Le modèle Entité-Association*

Ce modèle définit trois composants de base : l'entité, l'association et l'attribut (les définitions de ces termes sont extraites de deux ouvrages : Elmasri & Navathe, 1989 et Teorey, 1990).

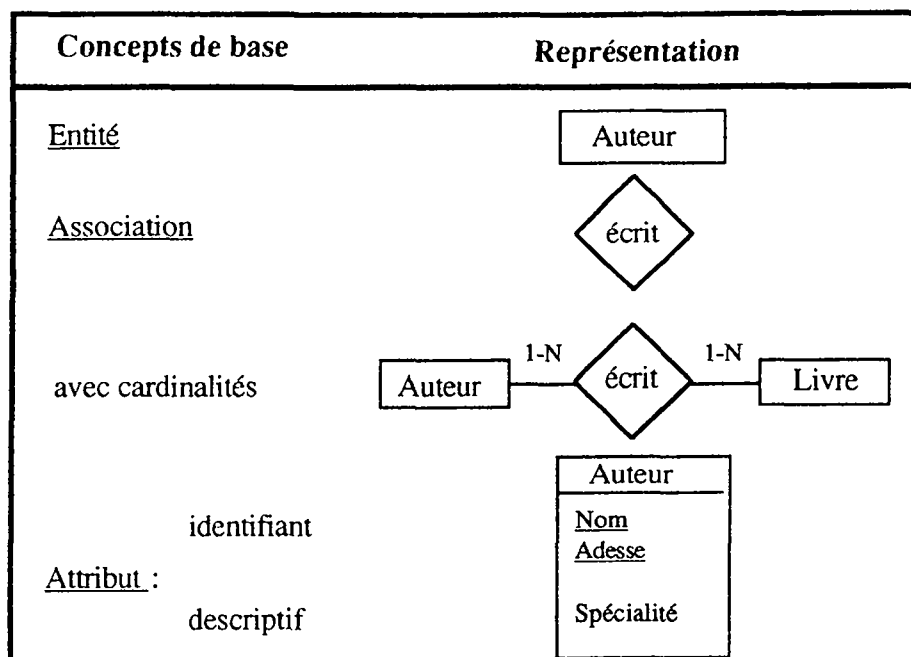


Figure 2 : Les concepts de base définis dans le modèle Entité-Association et leur représentation graphique.

- L'*entité* désigne un objet qui est représenté dans la BD, et qui peut être caractérisé par un ou plusieurs attributs ; elle peut désigner une personne, un lieu, une chose ou un événement. Par exemple, l'Auteur, le Livre, la Commande sont des exemples d'entités. Une occurrence particulière d'une entité (un auteur particulier par exemple) est appelée une *instance* d'entité.

Une entité est graphiquement représentée par un rectangle (cf fig.2) ;

- L'*association* décrit une relation dite du monde réel entre plusieurs objets. Dans une association, chaque entité a un *rôle* particulier. Par exemple, le rôle "Ecrit" définit la fonction de l'entité Auteur dans l'association entre un Auteur et un Livre. En général, on confond le nom du rôle et le nom de l'association. La règle dicte en outre que les noms de rôle doivent correspondre à des verbes (quand cela est possible), et pour faciliter la lecture du schéma conceptuel, doivent dénoter la relation de gauche à droite, ou du haut vers le bas (cf fig.2). Une association peut toutefois être lue dans les deux sens (un Auteur écrit un Livre et un Livre est-écrit par un Auteur).

On représente habituellement une propriété de l'association : les *cardinalités*. Elles décrivent les contraintes sur le nombre d'instances d'entités minimum et maximum reliées dans une association. Par exemple, les cardinalités de l'association entre Auteur et Livre sont 1-N (ce qui signifie qu'1 auteur peut écrire N livres - on peut bien sûr spécifier plus

précisément la valeur de N : 1, 2, 3, ...) et 1-N dans l'autre sens (1 livre peut être écrit par N auteurs).

- *L'attribut* désigne une propriété d'une entité ou d'une association qui comporte un intérêt pour le domaine d'application - un attribut ayant un intérêt si l'information qu'il porte est pertinente pour la gestion du domaine d'application (Flory, 1982). Par exemple, le Nom d'un Auteur ou l'Année d'édition d'un Livre sont des attributs. Une occurrence particulière d'un attribut est une *valeur* d'attribut. Il y a deux types d'attributs : l'identifiant et l'attribut descriptif. Un identifiant (ou clé) est utile pour déterminer de façon unique une instance d'entité ; il peut être constitué d'un ou plusieurs attributs si un seul attribut ne permet pas de respecter cette contrainte (par exemple, l'identifiant pour un auteur pourrait être le couple Nom+Adresse). Tout autre attribut est considéré comme un attribut descriptif. On les différencie graphiquement en soulignant l'identifiant (cf fig.2).

Différentes extensions de ce modèle initial (e.g. Elmasri & al., 85) ont été proposées au cours des quinze années qui suivirent sa parution. Elles partent du constat que le modèle E-R d'origine ne permet pas de représenter certaines notions importantes. L'une des extensions les plus connues a consisté à doter le modèle du concept de *Généralisation*. Cette association particulière spécifie que plusieurs entités avec certains attributs communs peuvent être généralisées en une entité de niveau supérieur, dénommée "entité générique". Par exemple, l'entité *Personne* est une entité générique des entités *Auteur*, *Editeur* et *Lecteur*.

1.4 Apports théoriques des études sur l'activité de conception

L'activité de conception d'une BD, mais ceci semble vrai pour toute activité de conception, est initialement confrontée à un ensemble de buts mal-définis, parfois ambigus et contradictoires (Eastman, 69, et Rowe, 87, sur la conception architecturale, Malhotra & al., 80, Jeffries & al., 81, Kant & Newell, 84, sur la conception de logiciels, Visser, 87, sur la conception d'automates programmables, Hayes & Flower, 80, sur la conception de textes, Dasgupta, 89, sur des problèmes d'architecture de systèmes, Darses, 90, sur la conception de réseaux informatiques). En outre, le concepteur ne dispose d'aucune méthode formalisée qui permette, à partir de ces données, de résoudre le problème de définition des buts. Son activité va donc au départ s'appuyer sur un ensemble d'heuristiques qui l'amèneront à recueillir de nouvelles données, et ainsi à mieux définir les buts initiaux. Le recours à des problèmes analogiques ou la simulation de scénarios d'utilisation de l'artefact visé sont des exemples de ces heuristiques (Guindon, 90).

Une part importante de l'effort déployé pour résoudre le problème sera donc consacré à rechercher des informations manquantes (Simon, 73).

Plusieurs auteurs caractérisent ce type de problème comme un problème *mal-structuré* (Eastman, 69 ; Simon, 81), qu'ils opposent aux problèmes bien-structurés. Ces derniers possèdent en effet des buts clairement établis, une quantité de données initiales relativement peu importante, des règles qui permettent d'atteindre le but et un ensemble de critères pour évaluer objectivement l'atteinte du but.

Si l'ensemble des données existe, mais est trop vaste, comme dans le cas du jeu d'échec (l'ensemble des données est constitué dans ce cas, à la fois des différentes positions des pions, des différents déplacements autorisés, mais aussi des nombreuses interactions entre les différents déplacements possibles), ce sont les capacités de traitement du résolveur qui vont poser problème. Simon (73) définit alors la notion de "rationalité limitée" pour désigner ce phénomène. L'incapacité à envisager l'ensemble de l'espace-problème, qui est une nécessité pour choisir la *meilleure* solution, conduira d'une part le résolveur à introduire des critères de sélection qui, en s'ajoutant à l'ensemble des buts initialement exprimés, permettront de réduire l'espace-problème, d'autre part, et ceci est une conséquence de l'énoncé précédent, à choisir une solution *satisfaisante*.

Ces critères sont désignés par certains auteurs (Janssen & al. 89, Darses, 90) sous le terme de *contraintes de préférence*, et opposés aux *contraintes de validité* qui définissent des propriétés que l'objet à concevoir devra obligatoirement vérifier.

Cette nécessité d'introduire de nouvelles données au cours de la résolution du problème soulève une question en ce qui concerne la validation de la solution produite : est-ce que la validation d'une solution qui repose, même seulement en partie, sur des contraintes de préférence (i.e des éléments introduits par le concepteur) prend la même forme qu'une solution qui repose uniquement sur des contraintes prescrites par ou avec les utilisateurs? On peut penser a-priori, que dans le premier cas, la validité d'une contrainte de préférence devra elle-même être vérifiée afin de valider la solution qu'elle a permis de générer.

L'étude présentée dans ce rapport rend compte de l'analyse de dialogues de validation d'un SCD. On s'est intéressé principalement :

- à dégager un certain nombre de caractéristiques générales à ce type de situation, concernant les rôles pris par les différents acteurs, les actions engagées et le contenu informationnel qu'elles véhiculaient ; on a notamment recherché un effet du type de contraintes à l'origine de la solution proposée, contrainte de validité ou contrainte de préférence, sur le contenu du dialogue.

- à identifier plus clairement les besoins en information des utilisateurs, provoqués par l'objectif de validation du SCD, et la façon dont sont obtenues ces informations. On a été particulièrement attentif au rôle du concepteur face à ces besoins, la relation concepteur-utilisateur telle qu'elle a été observée idéalisant l'interaction entre l'utilisateur et le système d'aide à la conception de BD visé.

2- METHODOLOGIE DE L'ETUDE

2.1 Le terrain

La phase de validation du Schéma Conceptuel des Données (SCD) intervient à la suite d'une série de réunions ayant permis d'affiner les besoins et de spécifier plus en détail un système d'information (SI) destiné à deux services d'une même entreprise. Le fonctionnement de ces deux services est extrêmement lié : le premier service enregistre des commandes de vêtements réalisés par les agents de l'entreprise, le second est chargé de fournir les vêtements commandés.

Le concepteur a produit deux SCD en utilisant le formalisme Entité-Association, un pour chaque service. Les deux SCD regroupent un ensemble d'informations communes et un ensemble d'informations spécifiques, correspondant aux besoins spécifiques d'un service donné.

L'objectif de cette étape est de faire valider ces SCD, afin d'engager la réalisation technique de ces outils.

Cette étude est basée sur l'observation, en situation naturelle, de quatre situations de validation réparties en deux réunions. Deux de ces situations portaient sur la validation d'un premier schéma, représentatif d'un premier service (S1), les deux autres sur le second schéma, représentatif de l'autre service (S2).

Deux groupes d'utilisateurs ont participé indépendamment à ces réunions : le premier groupe, "décideur" en ce qui concerne le service S1 (i.e ayant une connaissance très détaillée du fonctionnement de S1 et ayant défini les principaux besoins, nécessaires à l'activité quotidienne du service), et "concerné" par S2 (dans ce cas, les utilisateurs n'ont qu'une connaissance partielle de S2, et sont à l'origine de certains besoins plus ponctuels) ; le deuxième groupe, à l'inverse, est "décideur" en ce qui concerne S2 et "concerné" par S1.

Nous distinguerons deux types d'utilisateurs en conséquence :

- *l'utilisateur fréquent* : il correspond à l'utilisateur qui a des besoins quotidiens liés à l'utilisation d'une BD et qui est décideur dans le domaine d'application ;
- *l'utilisateur occasionnel*, qui a des besoins plus ponctuels liés à cette même BD.

Un groupe d'utilisateurs donnés est donc tour à tour dans la position de l'utilisateur fréquent (face au SCD qui concerne son domaine d'application) et dans la position d'utilisateur occasionnel (face au SCD qui concerne l'autre domaine d'application).

Pour résumer, on distinguera donc les quatre situations de validation suivantes :

1ère Réunion		2ème Réunion	
SCD1/ Utilisateur Fréquent	SCD2/ Utilisateur occasionnel	SCD1/ Utilisateur Occasionnel	SCD2/ Utilisateur Fréquent

Les groupes de validation comprenaient en outre un chef de projet et un observateur. Chaque participant détenait une copie du SCD élaboré et présenté par le concepteur.

2.2 Méthode de recueil des données

L'observateur disposait d'un magnétophone afin d'enregistrer l'ensemble des dialogues de validation pour les quatre situations de validation. Ces dialogues ont été retranscrits intégralement.

On a en outre rassemblé les schémas conceptuels utilisés par le concepteur, sur lesquels il avait noté les modifications provoquées lors des différentes réunions.

3 - RESULTATS

3.1 Observations générales

3.1.1 Déroulement global d'une réunion de validation

A chaque réunion, deux schémas étaient donnés à valider. Dans toutes les situations, c'est le concepteur qui a le contrôle du déroulement de la réunion : il est à l'origine de la quasi-totalité des éléments de solution auxquels on fera référence (à l'exception de trois cas), et recentre le dialogue lorsque celui-ci s'écarte de l'objectif principal, la validation du SCD.

La validation d'un SCD procède élément de solution par élément de solution : en général, le concepteur fait référence à une entité du SCD, puis à chacun de ses attributs et enfin à la ou les associations à laquelle l'entité participe, ce qui lui permet d'introduire une nouvelle entité.

Les utilisateurs dans un certain nombre de cas n'interviennent pas ou quasiment pas (uniquement des régulateurs du dialogue type "mmh mmh"). Dans d'autres cas, ils peuvent exprimer plus explicitement leur adhésion ("ah ça, c'est bien", "oui, oui d'accord", etc.). Enfin ils peuvent soit poser des questions au concepteur :

- des demandes de justification du type : "pourquoi a-t-on besoin de telle information?"
 - ou des demandes d'information ;
- soit exprimer une contestation.

Enfin, on précisera qu'une partie importante des thèmes de discussion est consacrée spécifiquement aux traitements à réaliser sur les données. Ce point est détaillé en 3.1.2.

3.1.2 Temps variable

Le temps consacré à la validation d'un SCD pour chaque situation est présenté ci-dessous :

1ère Réunion		2ème Réunion	
SCD1/ U. occas.	SCD2/ U. fréquent	SCD1/ U. fréquent	SCD2/ U. occas.
66'	77'	25'	60'

Tableau 1 : Temps total (en mn) consacré à la validation d'un SCD

On constate que ces différences ne peuvent être attribuées ni au type d'utilisateurs, ni au SCD traité. Par contre, les deux situations de validation composant la première réunion prennent plus de temps que lors de la seconde réunion. On remarque notamment le faible temps (relativement aux autres) correspondant à la troisième situation de validation.

Les éléments dont nous disposons dans l'état actuel de l'étude ne permettent pas d'expliquer ces variations.

3.1.3 Données et Traitements

Le SCD représente une description statique de la BD. On observe pourtant que la validation d'un SCD face aux futurs utilisateurs soulève un certain nombre de questions relatives aux traitements à réaliser sur les données. On peut comprendre cela du fait que les principales préoccupations des futurs utilisateurs sont de savoir ce qu'ils vont pouvoir utiliser pour atteindre leurs objectifs et comment ils vont pouvoir l'utiliser.

Le temps total consacré à la validation d'un SCD peut ainsi être réparti selon que les questions traitées relèvent essentiellement des données ou des traitements.

Un thème de discussion relève spécifiquement des données si l'on cherche à comprendre ce qu'une donnée représente, ou si l'on traite de son intérêt.

Un thème de discussion relève spécifiquement des traitements si l'on traite des opérations à effectuer sur ou avec des données. Ici ce sont les actions qui peuvent être précisées ou contestées.

Il s'est avéré parfois difficile de distinguer ce qui relevait spécifiquement des données ou des traitements. Par exemple, quand le concepteur dit : "bon, pour les CEA, on va mémoriser un certain nombre d'informations : le code CEA, la désignation, etc.", on s'aperçoit que les données "code CEA", "désignation" et les suivantes sont présentées dans un contexte faisant apparaître une action sur ces données, d'où l'ambiguïté possible. Dans ce type de cas, on a alors considéré que le thème du dialogue restait les données.

Le tableau 2 présente la répartition du temps total selon la situation de validation et le type de questions traitées.

	1ère Réunion		2ème Réunion		Moyennes
	SCD1/ U. occas.	SCD2/ U. fréquent	SCD1/ U. fréquent	SCD2/ U.occas.	
Temps total	66' 100	77' 100	25' 100	60 100	57' 100
Temps consacré aux données	39'40" 60	51' 66	24' 96	24'30" 35	34'40" 64,25
Temps consacré aux traitements	26'20" 40	26' 34	1' 4	35'30" 65	22'15" 35,75

Tableau 2 : Répartition du temps (en mn et en pourcentages)

Le temps imparti aux questions relevant des traitements représente en moyenne un peu plus d'un tiers du temps total (35,75%). Mais on doit remarquer que ce temps peut être quasiment nul, ou à l'opposé représenter près des deux tiers du temps total.

On remarque en outre que le temps consacré aux données est sensiblement le même dans les deux dernières situations de validation, et que ce temps est inférieur quasiment de moitié par rapport aux temps consacrés aux données dans les deux premières situations de validation.

3.1.4 Description partielle du SCD

Dans aucune des situations de validation la totalité des informations présentées sur un SCD n'a été traitée. Les figures 1 à 4 permettent de visualiser cette observation.

Le tableau 3 présente le taux d'éléments de solution mentionnés par situation de validation. Parmi les éléments de solution, on a distingué : les entités, les attributs, que l'on différencie selon qu'il s'agit d'un identifiant ou d'un attribut autre, les associations, et les cardinalités.

	1ère Réunion		2ème Réunion		Moyennes
	SCD1/ U. occas.	SCD2/ U. fréquent	SCD1/ U. fréquent	SCD2/ U. occas.	
Entité	100	83,3	100	100	95,8
Attribut	46,1	66,6	68	56,3	59,2
Identifiant	33	20	55	30	34,5
Association	82	58,3	90,9	83,33	78,6
Cardinalités	59,1	20	45	44	42

Tableau 3 : Pourcentages d'éléments de solution mentionnés par situation de validation

On remarque le fort pourcentage d'entités mentionnées : ceci s'explique notamment par le fait qu'une entité peut non seulement être mentionnée de façon isolée, mais aussi pour introduire un attribut ou pour proposer une association à laquelle elle participe.

La sélection des informations à présenter s'opère donc principalement sur les autres éléments de solution, et plus particulièrement sur les attributs, les identifiants et les cardinalités.

Enfin, on remarque un effet du type d'utilisateur sur le taux d'attributs mentionnés : ce taux est toujours plus élevé face aux utilisateurs fréquents. Excepté cet effet, le type d'utilisateur et le type de SCD (SCD1 ou SCD2) ne semble pas conditionner les variations observées.

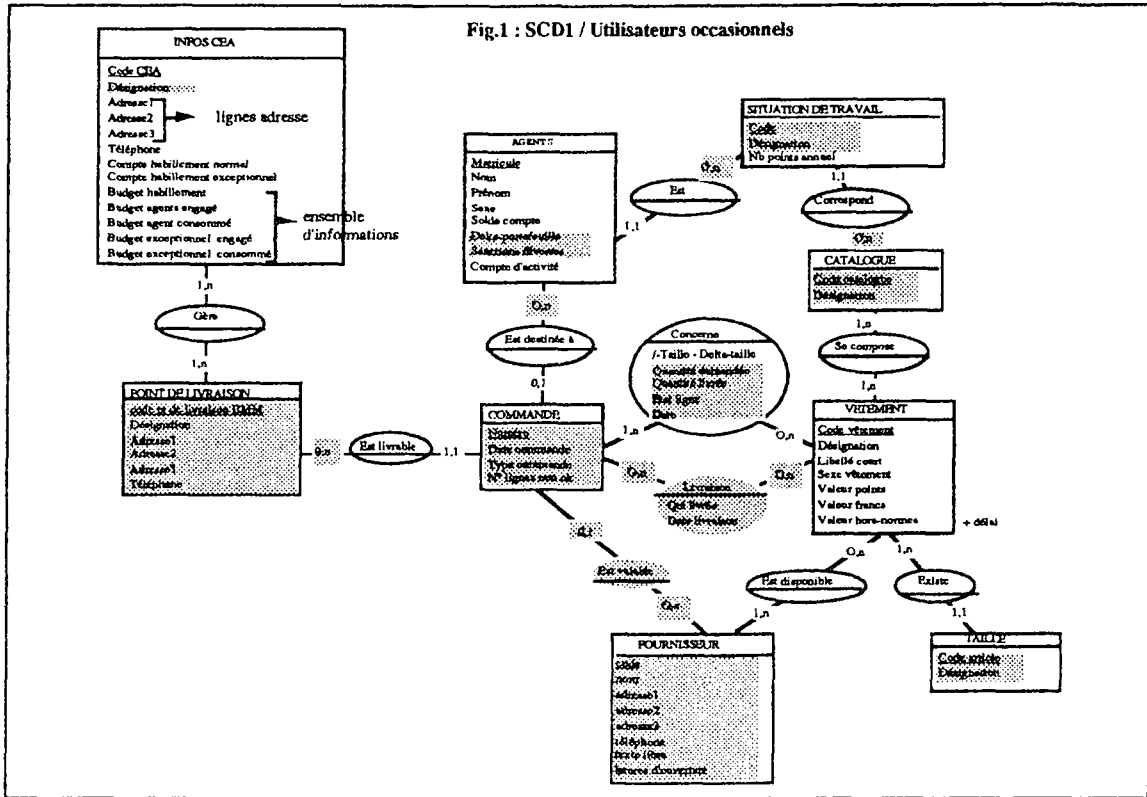


Fig.3 : SCD1 / Utilisateurs fréquents

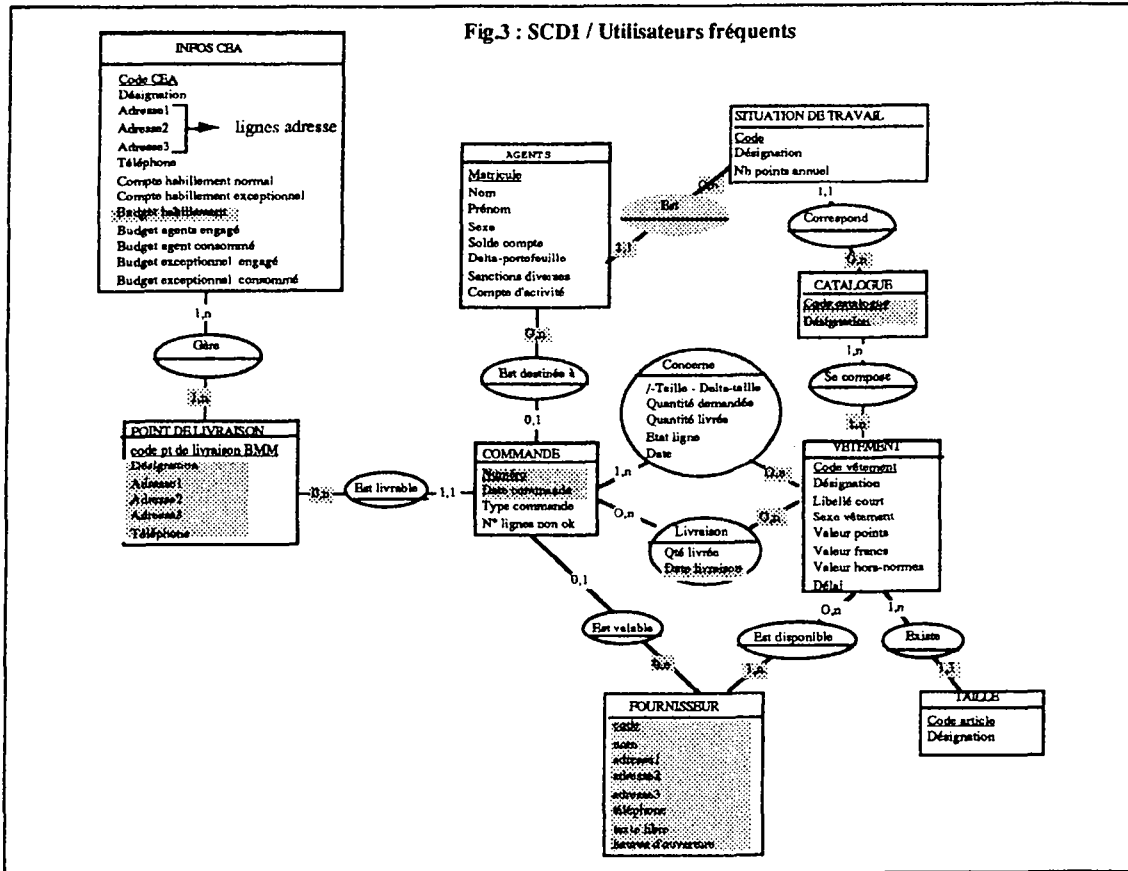
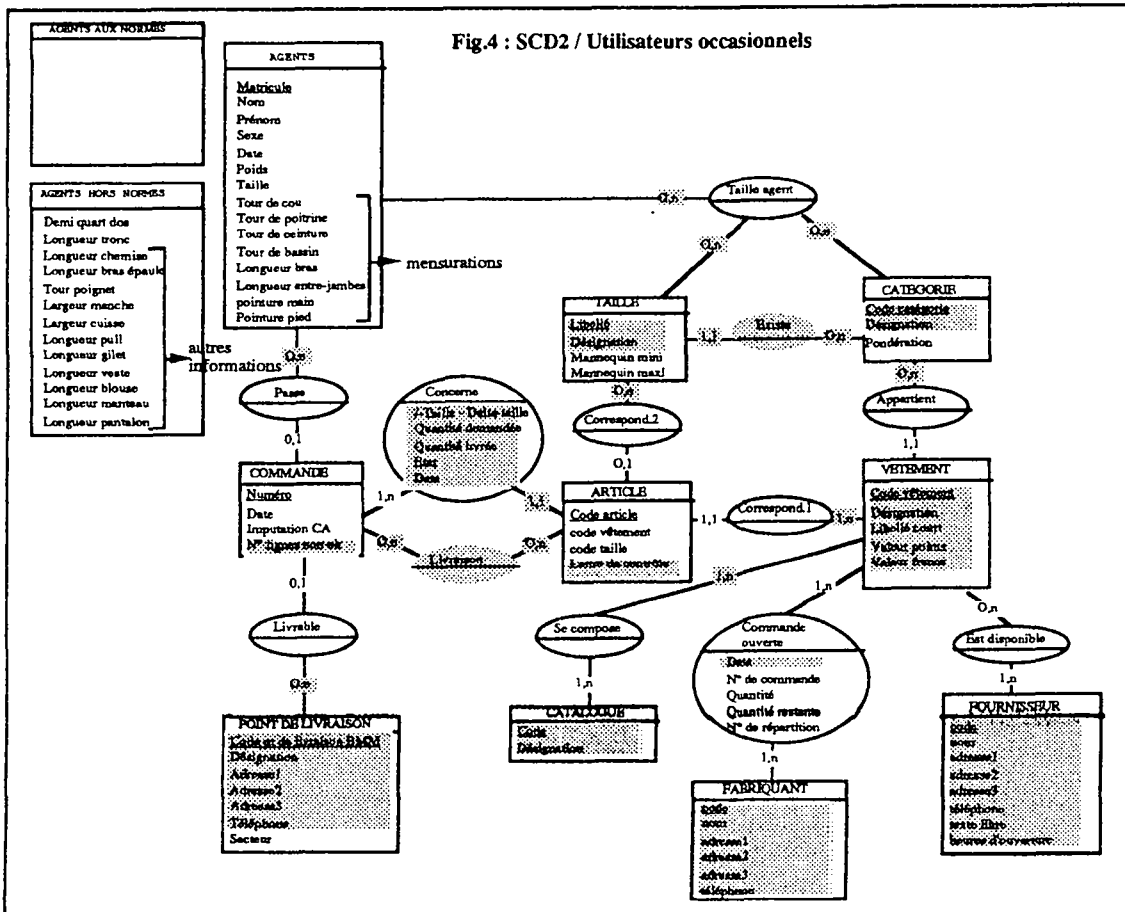


Fig.4 : SCD2 / Utilisateurs occasionnels



informations non mentionnées

] -> mensurations : terme référant à un groupement d'attributs

3.1.5 Différents types de dialogue

Il semble difficile de penser que toutes les interventions des différents membres participant à la validation d'un SCD ne recherchent qu'à évaluer et juger les éléments de solution successivement proposés. A l'inverse, on peut considérer que différents types de dialogues prennent place, chacun étant caractérisé par un objectif et une séquence d'actes de langages bien particulier. On a identifié quatre types de dialogue : le dialogue d'évaluation et le dialogue de négociation, à l'intérieur desquels peut prendre place un dialogue de clarification, et le dialogue d'analyse d'un problème. L'ensemble de la structure du dialogue de validation est synthétisé dans la figure 5. Le détail des actions engagées durant chaque type de dialogue et des transitions entre eux est donné en annexe.

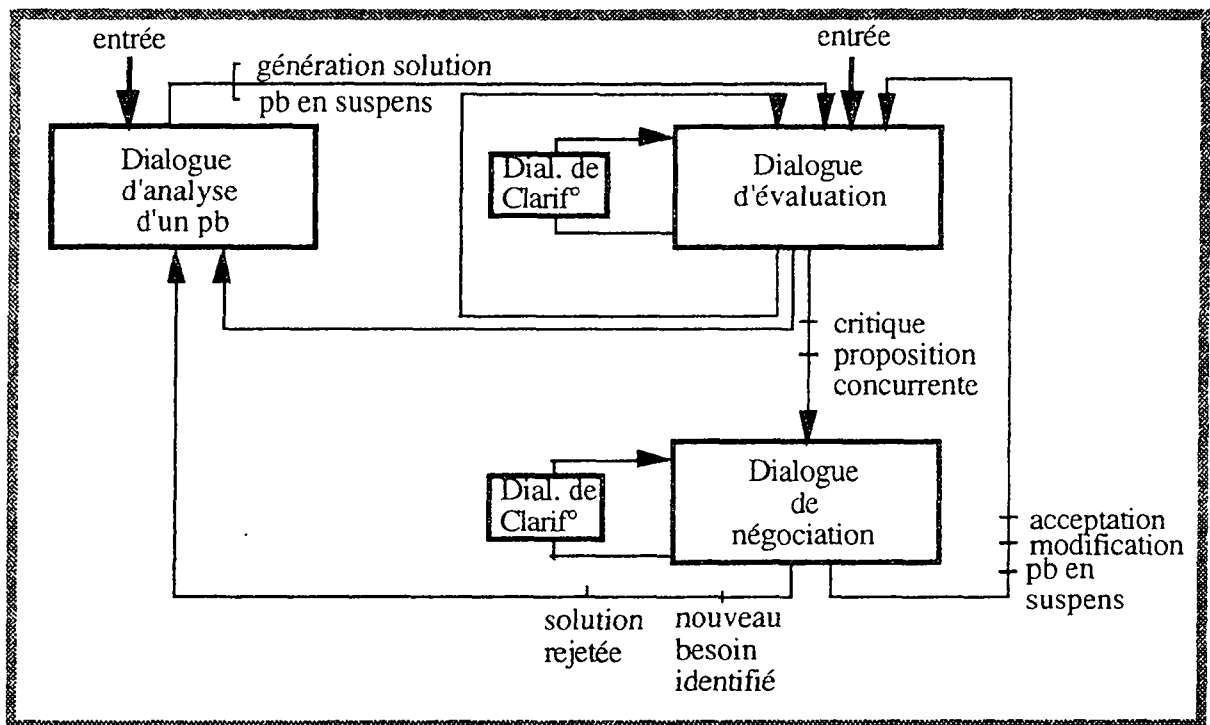


Figure 1 : Structure du dialogue dans une phase de validation

Le dialogue d'évaluation se traduit par une *proposition* d'un élément de solution suivie d'une *évaluation* des utilisateurs. Le concepteur a, à l'exception de trois cas, l'initiative de ce type de dialogue. Autrement dit, c'est lui qui introduit successivement les différents éléments à valider. Ceci dit, les exceptions sont importantes à prendre en compte puisqu'elles mettent en évidence la possibilité qu'a l'utilisateur d'interrompre le flot de propositions. Ceci doit d'ailleurs être mis en relation avec le fait que le concepteur ne mentionne qu'une partie du SCD ; il peut en effet

compter sur ce "relais de l'initiative" pour éviter de décrire en détail l'ensemble des informations contenues dans le SCD.

Le dialogue de clarification est nécessaire à ce niveau si l'utilisateur ne comprend pas un aspect de la proposition : cela peut se traduire par des demandes d'information, ou des demandes de justification lorsque l'utilisateur ne comprend pas l'intérêt d'une solution proposée (exemple de demande de justification : "mais pourquoi on a besoin de BMM dans cette application?") qui amène le concepteur à compléter sa proposition initiale.

On remarque qu'aucune requête de l'utilisateur ne vise à clarifier la signification d'une donnée (par exemple, l'utilisateur aurait pu adresser une requête du type : "qu'est-ce que vous entendez par 'delta-portefeuille'?" ou " je ne comprends pas ce que désigne 'état ligne'"). Elles portent en effet le plus souvent sur les traitements ; exemple : "les informations dans les CEA, quand est-ce que vous les traitez? en temps réel ou en temps batch?"

D'autre part, on constate un nombre très faible de demandes de justification sur les quatre situations de validation (uniquement trois).

Ceci conduit à penser que la proposition exprimée par le concepteur comporte en général toute l'information sur la signification et l'utilité de l'élément de solution concerné, et néglige dans certains cas les informations sur les traitements nécessaires pour que l'utilisateur puisse émettre un jugement. Finalement, cette proposition est dans une grande majorité des cas compréhensible et validable.

Reste cependant à préciser en quoi consiste exactement cette proposition émise par le concepteur ; étant donné l'un des objectifs de cette étude est d'évaluer le caractère compréhensible et validable du SCD, nous devons comparer la proposition émise par le concepteur et l'information prélevable uniquement sur le SCD. Cette question est traitée en 3.2.

La proposition de solution peut être acceptée, ce qui amène le concepteur à introduire la proposition suivante, ou contestée : un conflit apparaît alors et engage le groupe dans un dialogue de négociation. Sans entrer dans le détail des actions engagées durant ces phases (cf. annexe), on soulignera le fait que le concepteur n'accepte jamais automatiquement une contestation - ce qui le conduirait à modifier la solution proposée. Il commence toujours par *argumenter* son choix, ou contre-argumenter la critique émise. Ceci évite d'opter pour une modification alors qu'en réalité l'utilisateur soit n'avait pas totalement compris la proposition du concepteur (donc l'avait critiqué sur des bases fausses), soit n'avait pas envisagé d'autres contraintes à prendre en compte, soit émettait un point de vue contradictoire avec une décision antérieure.

En bref, le dialogue de négociation se caractérise par deux types d'activité : une activité argumentative et une activité de résolution de conflit qui conduit à modifier la solution proposée.

3.1.6 Les modifications apportées à la solution

La modification de la solution est une issue possible du dialogue de négociation, et vise à intégrer les buts exprimés par les membres du groupe et non-pris en compte par la solution proposée.

On observe globalement peu de modifications, pas plus de trois par situation de validation. En outre, elles apparaissent relativement mineures : il n'y a pas de modification massive de la structure du SCD.

Les modifications observées consistent soit à :

- SUPPRIMER un élément du SCD

1er cas : si il contredit un but des utilisateurs (par exemple, l'attribut 'valeur-francs' est supprimé car sa mise à jour implique une élévation de la charge de travail du service) ;

2eme cas : si il ne rend compte d'aucun but des utilisateurs, i.e si il n'est pas jugé utile. Cette situation est possible si l'on considère que le concepteur peut dans certains cas proposer des solutions "standards" qui ne conviennent pas nécessairement.

- AJOUTER un élément au SCD, lorsqu'un nouveau besoin a été élicité, et que ce besoin est suffisamment clair pour proposer une solution.

- REMPLACER le nom d'un élément de solution

1er cas : si celui-ci est refusé par les utilisateurs ;

2eme cas : si il existe déjà un terme dans le domaine d'application pour désigner l'information dénommée par le concepteur. Ceci se manifeste par une intervention d'un utilisateur qui signale l'existence d'un autre nom désignant le même objet.

3.1.7 L'issue d'une réunion de validation

A l'issue d'une réunion de validation, on constate d'une part qu'un ensemble d'éléments de solution a été approuvé par les utilisateurs, certains d'entre eux ayant été modifiés, d'autre part qu'un certain nombre de problèmes restent en suspens.

La validation du SCD, dans son ensemble, n'est donc pas achevée.

La raison principale en est que toutes les connaissances nécessaires pour valider un SCD n'ont jamais été disponibles au cours d'une seule réunion.

Le concepteur planifie alors de contacter certaines sources d'informations, soit pour *vérifier* des connaissances émises, mais dont le statut est hypothétique, soit pour *rechercher* les informations permettant de mieux définir un problème non résolu avec le groupe d'utilisateurs présents.

Il est donc difficile d'envisager qu'une étape de validation puisse se réduire à une réunion de validation. D'autant qu'au cours de la seconde réunion enregistrée, certains besoins potentiels concernant les utilisateurs présents à la première réunion ont été élicités, nécessitant le recours, à nouveau, à ces utilisateurs précisément.

3.2 Les connaissances nécessaires à la validation du SCD

Il y a deux motivations qui conduisent à étudier spécifiquement le problème des connaissances nécessaires à la validation d'un SCD :

1. On a vu qu'à partir du SCD, un utilisateur doit exprimer un jugement. Une des nécessité du SCD soulignée par Olle & al. (90) est d'être compréhensible et validable par des utilisateurs non-initiés en modélisation conceptuelle. Autrement dit, il doit comporter les informations nécessaires pour qu'un tel utilisateur puisse exprimer un jugement. La question est donc de déterminer si le SCD est efficient de ce point de vue.
2. La deuxième préoccupation rejoint la question qui vient juste d'être posée, mais dans une optique de spécification d'un système d'aide à la conception d'une BD, destinée à l'utilisateur-final. Quelles connaissances doit posséder la machine pour réaliser, en coopération avec l'utilisateur, la validation d'un SCD ?

Ces questions vont être traitées à travers un aspect important du dialogue de validation : la génération, par le concepteur, de la *proposition initiale de solution*.

Le concepteur doit se référer aux éléments du SCD pour présenter ses solutions : la proposition initiale correspond à cette référence initiale qui est faite à un élément du SCD.

3.2.1 La référence aux éléments du SCD

3.2.1.1 Comparaison entre la référence simple et le discours sur le SCD;

Le SCD décrit la structure de l'information propre à un domaine d'application. On peut donc, à partir de cette représentation formelle et en partie graphique, extraire un ensemble de connaissances énonçables en langage naturel. Si ces connaissances sont suffisantes pour valider le SCD, y faire référence, simplement en les mentionnant, devrait suffire.

La figure 6, ci-dessous, montre un exemple de référence "simple" aux éléments du SCD.

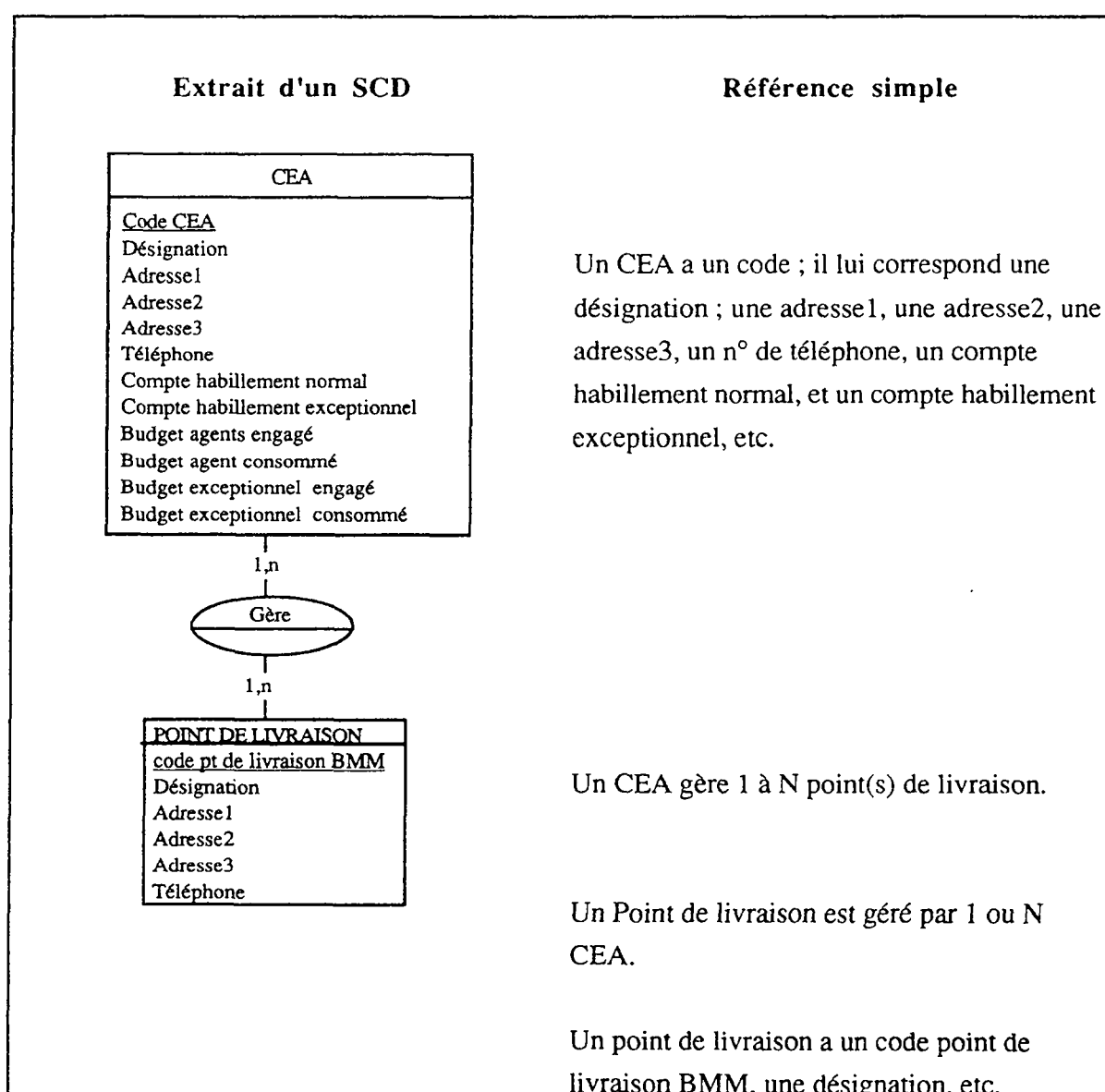


Figure 6 : Références simples d'éléments extraits d'un SCD

Les figures 7 et 8 rapportent le discours sur le SCD livré par le concepteur alors qu'il présentait sa solution.

Comme on le voit sur ces deux exemples (fig. 7 et 8), le discours du concepteur diffère assez sensiblement d'une référence simple aux éléments du SCD, qui n'aurait pour but d'ailleurs que d'aider à lire le SCD.

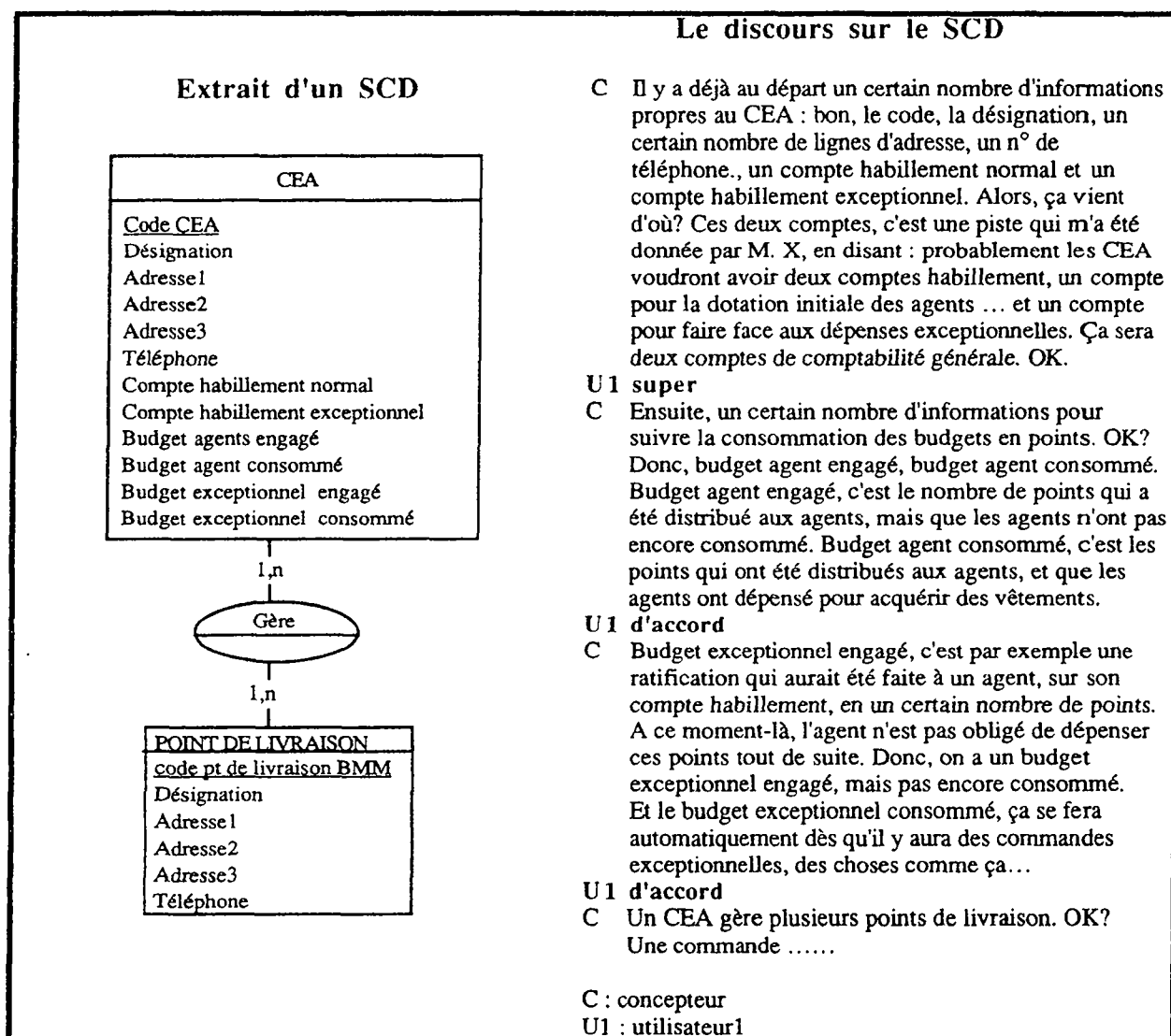


Figure 7 : Extrait du discours sur le SCD donné par le concepteur

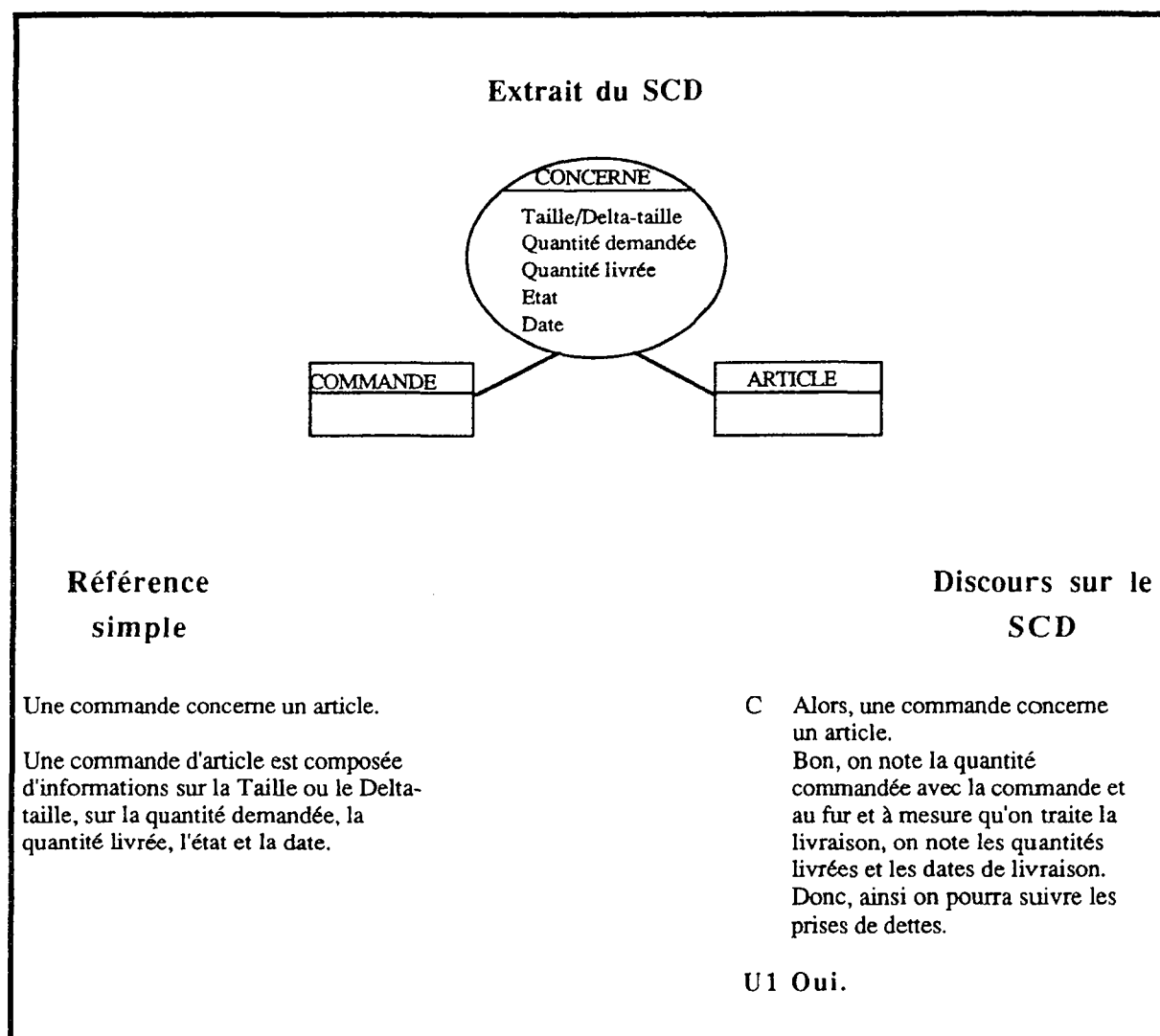


Figure 8 : Extrait d'un SCD sous forme de références simples, et tel qu'il est présenté par le concepteur.

On peut alors considérer l'hypothèse suivante : la distance qui sépare la référence simple du SCD et les propositions initiales effectives reflètent un ensemble de principes liés à la coopération avec des utilisateurs.

L'analyse des propositions initiales doit mettre en évidence certains de ces principes.

3.2.1.2 Les différents types de référence aux éléments du SCD

L'analyse du discours du concepteur, alors qu'il émet une proposition, fait apparaître différents types de référence aux éléments du SCD :

- **l'absence de référence** : on l'a déjà précisé en 2.1.4, le SCD décrit n'est qu'un sous-ensemble du SCD total. Un grand nombre d'informations n'est pas présenté ; l'utilisateur peut toutefois en prendre connaissance par lui-même, ayant une copie personnelle du SCD durant la réunion de validation.

- **la référence simple** : dans ce cas, le concepteur ne fait que mentionner les informations représentées sur le SCD.

On distingue deux formes de référence simple :

- une forme purement *descriptive* ; exemples : "un vêtement, c'est un code, une désignation, etc.", ou "le CEA gère plusieurs points de livraison".

- une forme *active*, dans laquelle le concepteur introduit une action sur les éléments référés ; exemple : "pour l'agent, je mémorise son matricule, son nom, prénom, ..." ou "pour chaque commande, on note la quantité commandée ; ..." (cf. fig.8).

- **la référence à un groupement** : plusieurs éléments du SCD peuvent être désignés par un terme unique. Par exemple, les attributs "adresse1, adresse2, adresse3" sont désignés par le terme unique de "lignes d'adresse". Dans cet exemple, le terme utilisé pour désigner le groupement est très particulier, et sa signification est en rapport (plus ou moins évident) avec les éléments référés.

Dans la plupart des autres cas, le terme utilisé est très général et sa signification est indépendante des éléments référés. Exemple : "pour les agents hors-normes, il y aura une série d'autres informations..." ; "autres informations" regroupe l'ensemble des attributs composant l'entité Agent Hors-Norme, lesquels correspondent à des informations sur les mensurations particulières aux agents hors-normes.

En général, les groupements se font à partir de plusieurs attributs d'une même entité.

Cependant dans certains cas, ils réfèrent à des éléments issus de plusieurs entités et associations. Ils sont alors toujours précédés ou suivis d'une justification.

Exemple (la citation est simplifiée ; se reporter à la figure 3, p.13) : "Alors comment faire si on ne veut pas commander des vêtements pour un agent, mais pour un pool d'agents ? Alors on voit qu'il y a 'type commande' (attribut de l'entité Commande),..., puis je dis qu'une commande est destinée à 0 ou 1 agent (cardinalités), c'est à dire que je peux avoir des

commandes destinées à personnes. Et dans ce cas, je ne mets pas le Delta-taille, mais la taille du vêtement (attribut de l'association Concerné), ..., ce qui me conduit à rajouter : un Vêtement existe en une Taille (association)."

Ces groupements font donc apparaître une structuration des données différentes de celle représentées sur le SCD, et qui apportent des informations importantes sur la signification et l'utilité des données considérées.

Enfin, on peut remarquer que l'intérêt d'une référence à un groupement semble être de *désigner un maximum d'informations avec la communication verbale la plus courte.*

- **la référence étendue:** à partir d'une référence simple ou d'une référence à un groupement, donnée explicitement ou implicitement, le concepteur peut ajouter certaines connaissances qui précisent le ou les éléments référés. Le résultat est une référence étendue. Exemple : "Budget agent engagé, c'est le nombre de points qui a été distribué aux agents, mais que les agents n'ont pas encore consommé".

La référence implicite est notée lorsque le concepteur énonce une connaissance qui se rattache à un élément du SCD, sans que celui-ci soit explicitement mentionné. Cela est possible si l'on considère que les utilisateurs suivent l'enchaînement des informations présentées sur la copie du SCD dont il dispose.

Exemple (se reporter à la figure 1) : " (pour les Agents hors-normes, on notera des mesures supplémentaires qui serviront à lui confectionner des vêtements à sa taille)¹ . Ou un agent peut être normal, auquel cas il n'y a aucune autre information à noter". Le concepteur ici fait référence à l'entité 'Agent Aux-Normes' sans pour autant la nommer.

On note que l'élément du SCD auquel le concepteur fait référence, c'est à dire l'élément auquel s'applique l'extension apportée, ne correspond pas toujours à une entité, un attribut ou une association. L'élément référé peut être :

- un *groupement d'éléments* du SCD. Exemple : "donc ça, ça nous permet de retrouver la taille qu'il faut à l'agent lorsqu'il passe une commande", "ça" faisant référence à plusieurs entités, certains de leurs attributs et les associations auxquelles elles participent.
- la *cardinalité minimum* ou la *cardinalité maximum*. Exemple de référence étendue à une cardinalité maximum et à une cardinalité minimum : "la commande peut être passée par 1 agent : ça c'est pour les commandes normales ; ou par 0 agent : ça, c'est pour traiter les commandes styles BMM" ;

¹ On met entre parenthèses une partie de l'énoncé qui permet de comprendre le contexte dans lequel est exprimée la proposition

- la *localisation d'un attribut* dans une entité donnée. Exemple : "le compte d'activité, c'est propre à un agent ; c'est à dire que plusieurs agents dans une même situation de travail peuvent avoir des comptes d'activités différents, donc c'est pas propre à une situation de travail".

Chacun de ces éléments constitue donc un élément de solution, même si celui-ci dans le cas du groupement, peut être décomposé en plusieurs éléments de solution.

Enfin, on précisera que la référence étendue à un élément du SCD concerne en moyenne, toute situation de validation confondue, **29% des informations représentées** (la moyenne est calculée par rapport à l'ensemble des éléments du SCD, y compris ceux auxquels le concepteur ne fait pas référence). Cette moyenne varie très peu d'une situation de validation à l'autre : entre 24,1 et 31,1%.

Cette dernière information apporte un élément de réponse à l'une des questions qui a été posée, à savoir si le SCD peut être jugé comme une représentation compréhensible et validable par des utilisateurs.

On assume en effet que l'extension apportée répond à un principe lié à la coopération (tiré du principe énoncé par Clark & Wilkes-Gibbs, 86) : *le concepteur doit s'assurer que les utilisateurs ont compris les éléments référés afin que ceux-ci puissent l'évaluer.*

Ce qui nous amène à dire que dans 29% des cas, l'information représentée sur le SCD n'est pas suffisante du point de vue de l'utilisateur.

Dans l'optique d'un système d'aide à la conception de BD, on doit donc considérer la nécessité de stocker les connaissances ajoutées en machine, afin que la collaboration entre l'homme et le système puisse être efficace.

On doit par conséquent être capable de répondre aux deux questions suivantes :

- quelles sont les extensions apportées aux références, i.e quelles sont les connaissances ajoutées ? (traitée en 3.2.2)
- quelles sont les conditions qui nécessitent d'apporter une extension à la référence à un ou plusieurs éléments du SCD ? (ce point n'est pas traité dans ce rapport).

3.2.2 Les connaissances ajoutées

On cherchera tout d'abord à classifier l'ensemble des connaissances ajoutées dans une référence étendue. Les conditions dans lesquelles les extensions sont apportées seront analysées ensuite.

3.2.2.1 Méthode d'analyse des connaissances ajoutées

On a pris en compte l'ensemble des énoncés qui reflètent une *proposition* du concepteur au cours du dialogue d'évaluation. Les connaissances recensées ne concernent donc pas les autres types de dialogue identifiés.

Le discours sur le SCD a été découpé de façon à ce que chaque élément du discours livre une information différente de celles contenues dans l'élément précédent et l'élément suivant.

Ainsi à partir de l'extrait présenté dans la figure 7, on peut découper le discours de la façon suivante (l'ensemble de l'extrait n'est pas reporté) :

Il y a déjà au départ un certain nombre d'informations propres au CEA : ..., 1.[un compte habillement normal] 2.[et un compte habillement exceptionnel.] 3.[Alors, ça vient d'où? Ces deux comptes, c'est une piste qui m'a été donnée par M. X, en disant :] 4.[probablement les CEA voudront avoir deux comptes habillement,] 5.[un compte pour la dotation initiale des agents], 6.[et un compte pour faire face aux dépenses exceptionnelles.] 7.[Ça sera deux comptes pour la comptabilité générale.] Ok?

8.[Ensuite, un certain nombre d'informations pour suivre la consommation des budgets en points]. OK?
9.[Donc, budget agent engagé], 10.[budget agent consommé.] 11.[Budget agent engagé, c'est le nombre de points qui a été distribué aux agents, mais que les agents n'ont pas encore consommé.] 12.[Budget agent consommé, c'est les points qui ont été distribués aux agents, et que les agents ont dépensé pour acquérir des vêtements.]

13.[Budget exceptionnel engagé], 14.[c'est par exemple une ratification qui aurait été faite à un agent, sur son compte habillement, en un certain nombre de points]. 15.[A ce moment-là, l'agent n'est pas obligé de dépenser ces points tout de suite. Donc, on a un budget exceptionnel engagé, mais pas encore consommé.] 16.[Et le budget exceptionnel consommé], 17.[ça se fera automatiquement dès qu'il y aura des commandes exceptionnelles, des choses comme ça...].

On peut alors faire correspondre à chaque segment un *type* de connaissance :

- les segments 1, 2, 9, 10, 13 et 16 correspondent à des références simples ; ils ne dénotent donc aucune connaissance ajoutée par rapport à la connaissance représentée sur le SCD ;
- le segment 3 correspond à une information sur l'*origine* du besoin énoncé à la suite ;
- les segments 4 et 8 révèlent des *buts d'utilisation du futur système* ;
- le segment 7 établit la *fonction* des deux comptes ;
- les segments 11 et 12 font état de *définitions* des différents budgets ;
- le segment 13 correspond à une *définition par l'exemple*, que nous ne différencierons pas d'une définition ;
- le segment 15 révèle une *règle* du domaine d'application ;
- enfin, le segment 17 précise une *procédure de mise à jour* ;

Tous les énoncés du concepteur ne sont pas pris en compte dans la classification. Certains en effet révèlent des connaissances non liées à la solution proposée, mais à la situation de dialogue. Par exemple : "Bon, je crois qu'on a fait le tour de ça", "Je n'ai pas eu la réponse à ma question : est-ce que tous le monde va essayer tous les vêtements?".

Ces énoncés relèvent en fait du contrôle de l'activité, et reflètent un autre niveau de connaissances.

3.2.1.2 Classifications des connaissances ajoutées

Trois étapes ponctuent l'activité de conception d'une BD : l'identification des faits pertinents à prendre en compte, la définition des spécifications ou buts du système, l'élaboration du SCD.

Les connaissances recueillies ou construites par le concepteur peuvent ainsi être réparties en trois sous-ensembles : les *faits*, les *buts du système* et les *éléments de solution*.

Seules les informations sur les deux premières constituent des connaissances ajoutées par rapport à la connaissance représentée sur le SCD.

- 1. Les Faits** : ils regroupent toutes les connaissances propres au domaine d'application recueillies par le concepteur.

Parmi les faits propres à un domaine d'application, on distingue les connaissances sur les objets, sur les procédures et sur les règles du domaine d'application :

- les connaissances sur les objets : le terme "objet" est utilisé ici dans un sens large, et désigne aussi bien des objets abstraits que concrets : une personne, un service, un formulaire, un système informatique ou encore une information particulière sont considérés comme des objets du domaine d'application.

Une connaissance sur un objet peut être donnée sous forme :

- d'une *définition* ("Budget agent engagé, c'est le nombre de points qui a été distribué aux agents, mais que les agents n'ont pas encore consommé").

Certains objets peuvent être considérés comme des classes constituées de sous-classes (par exemple, l'objet 'famille d'article' est composé des familles : vestes, gilets, chemises, etc.). L'énonciation des sous-classes pour clarifier un objet-classe est considérée comme un élément de définition de la classe.

- ou en précisant la *fonction* de l'objet ("...un compte exceptionnel : pour faire face aux dépenses exceptionnelles ;").

- les connaissances sur les procédures : elles font intervenir les actions portant sur les objets du domaine d'application. Par exemple : "pour l'imputation comptable d'une commande normale, on note le compte habillement normal, le compte CEA, et dans la troisième partie, le compte d'activité".

La description complète d'une procédure inclut les éléments suivants :

- l'(es) *événement(s)* qui conditionne(nt) l'exécution de la procédure (une date particulière, l'arrivée d'une commande, ...)
- le(s) *pré-requis* à l'exécution d'une procédure ("si l'agent a suffisamment d'argent sur son compte, il achète le vêtement qu'il désire")
- une *action* ou une séquence *d'actions* ("Donc, actuellement, lorsqu'ils reçoivent des bons, ils les trient manuellement dans différentes boîtes, et un jour ils disent : bon, je vais traiter le secteur 13 ; ils prennent alors la boîte 13 ; ils saisissent ; pendant la nuit, il y a une édition de feuilles et de petites fiches qui permet d'aller rechercher les articles qu'il faut livrer. Et le lendemain, ils font des paquets, et ça part."). On remarque que la description d'une séquence d'action comme celle qui vient d'être citée en exemple, inclut toujours la connaissance sur d'autres éléments de la procédure (par exemple, dans l'exemple cité, l'événement : "lorsqu'ils reçoivent des bons").
- et le *résultat*. de la procédure ("on entre le matricule de l'agent et le code vêtement, et la machine trouve l'article qu'il faut à l'agent").

Le concepteur ne livre en fait que très rarement la connaissance complète sur une procédure : il n'énonce généralement qu'une action, voire l'événement et l'action qu'il déclenche (exemple : "alors justement, au moment où l'agent passe sa commande, il demande à ce que la commande soit livrée à un point de livraison), ou un pré-requis et l'action (cf l'exemple de pré-requis ci-dessus).

- les connaissances sur les règles du domaine d'application : le SCD doit prendre en compte un ensemble de règles qui contraignent les valeurs possibles des données, ou les relations entre différentes données. Les cardinalités représentent un élément de solution qui prend en compte une partie de ces règles (par exemple, le fait : "un vêtement donné peut être fourni par plusieurs fournisseurs" est représenté par la cardinalité maximum N dans l'association 'un Vêtement est disponible chez 0,N Fournisseur').

Mais il existe un ensemble d'autres règles non représentées sur le SCD, bien qu'énoncées dans le discours sur le SCD. Par exemple : "tous les vêtements appartenant à 1 famille d'article donnée ont des dimensions équivalentes" ou "le critère le plus important pour choisir une chemise est le tour de cou".

2. Les buts de conception : ces éléments de connaissance indiquent dans quelles conditions le futur système devra fonctionner, et ce qu'il doit permettre. Ils apparaissent le plus souvent pour justifier un élément de solution (exemple : l'information 'secteur' va permettre de demander à la machine de trier automatiquement les livraisons par secteur), parfois pour justifier un autre but (on doit utiliser l'application BMM - but1 - pour pouvoir imputer les commandes directement sur le compte des CEA - but2 justifiant le but1).

Il faut distinguer le but auquel un élément du schéma répond, et la fonction de l'objet du domaine d'application auquel il fait référence : par exemple, l'objet 'compte habillement exceptionnel' a pour fonction de faire face aux dépenses exceptionnelles ; par contre, la donnée 'compte habillement exceptionnel' permet d'enregistrer toute modification du compte, et ceci afin de pouvoir suivre le montant de ce compte.

Les buts recueillis dans le discours du concepteur appartiennent à différents domaines d'expertise :

- *les buts liés au fonctionnement de l'organisation* :

Il s'agit de buts dépassant le cadre du système à concevoir. Par exemple, on peut vouloir éviter les fraudes ou responsabiliser les employés de l'organisation.

En réalité, un élément du SCD ne permet pas à lui seul de satisfaire ce type de contrainte. C'est son utilisation qui le permet.

Exemple : "On va indiquer l'information "délai" pour chaque vêtement : ça permettra aux agents de prévoir de passer leur commande un certain temps avant, et ainsi on pourra éviter les ruptures de stock que l'on connaît actuellement".

- *Les buts liés à l'utilisation du futur système :*

Ces buts précisent les actions désirées sur le système (exemple 1) et/ou les sorties attendues du système (exemple2).

Exemple1 : "...et puis il y aurait un certain nombre d'informations qui concerne l'état de consommation de leur budget" (connaître l'état de consommation des budgets est l'une des utilisations désirées de la BD).

Exemple2 : "donc, en fonction des vêtements commandés, on va pouvoir proposer à l'agent la liste des fournisseurs chez lesquels il peut aller."

- *Les buts liés au fonctionnement du système :*

Ce type de buts relèvent essentiellement de l'expertise du concepteur, contrairement aux deux autres. Ils concernent l'architecture du futur système, l'accès aux données, des critères comme le temps de réponse ou la quantité d'informations à traiter, etc.

Exemple : "puisque l'on va rentrer à terme dans l'application BMM, il faut que la table point de livraison utilisée dans le CEA soit compatible avec la table qu'on va trouver dans BMM. C'est pour cela que j'ai l'information code point de livraison BMM".

Il faut remarquer que ce type de buts joue le plus souvent un rôle de contrainte, c'est à dire qu'il induit des restrictions sur les solutions possibles au niveau du SCD. Dans l'exemple ci-dessus, le but de compatibilité des données contraint la dénomination de l'identifiant du Point de Livraison, le 'code point de livraison BMM'.

On remarquera que la classification des buts de conception ne fait apparaître aucun but lié à l'activité de formalisation. En fait, le concepteur ne justifie jamais les éléments de solution par ce type de buts. Pourtant, il est évident qu'un certain nombre de buts dictés par le modèle utilisé (le modèle Entité-Association dans le cas observé) sont pris en compte pour générer le SCD. Par exemple, le fait de désigner une association par un verbe dans la majorité des cas, ou de considérer que l'information 'date de la commande' est un attribut de l'association Concerne plutôt que de l'entité Commande relève de buts dictés par le modèle Entité-Association. Des éléments de discussion sur ce point sont donnés plus loin.

Un dernier type de connaissances ajoutées a été identifiée :

- *l'origine*. Il s'agit de la source d'information (une personne ou un document) qui est à l'origine des connaissances livrées par le concepteur (le plus souvent, les buts pris en compte par les solutions proposées, parfois les faits). Exemple : "Le compte habillement normal et le compte habillement exceptionnel. Alors, ça, c'est une piste qui m'a été donnée par X, ...".

3.2.1.3 Un deuxième niveau de référence

La classification des connaissances qui vient d'être présentée a été élaborée pour déterminer ce qui constituait l'extension d'une référence à un ou plusieurs éléments du SCD. On peut alors considérer que la proposition du concepteur reflète dans ce cas une référence d'un second niveau, aux connaissances composant d'une part la représentation mentale du domaine d'application, d'autre part la représentation des buts de conception (figure 9).

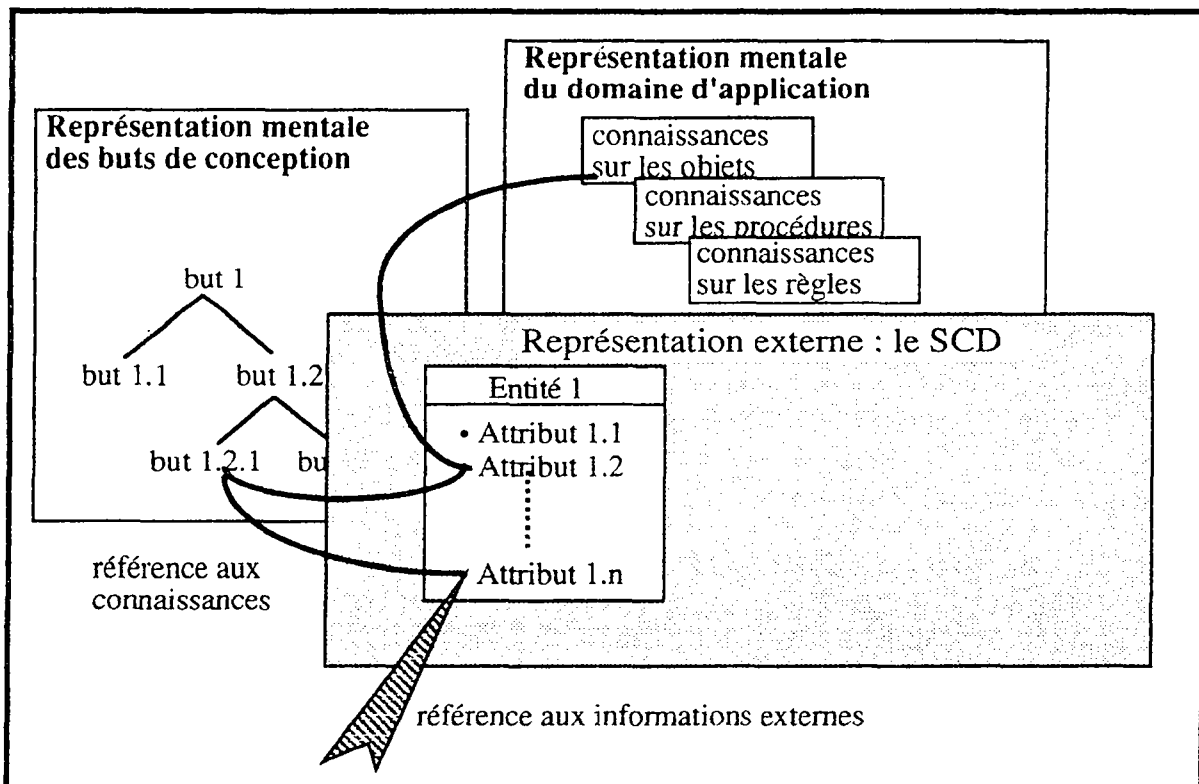


Figure 9 : Les deux niveaux de référence constituant une proposition

Contrairement à la référence aux éléments du SCD (référence aux informations externes), la référence aux autres connaissances ne se fait pas sur la base d'une représentation externe. Pourtant, on peut distinguer les mêmes types de référence (on désignera la référence aux informations externes par REF1, et la référence aux connaissances par REF2) :

- *l'absence de référence aux connaissances* : le concepteur ne donne pas systématiquement une référence aux connaissances ; c'est le cas lorsque l'on observe une référence simple aux éléments du SCD.

- *la référence simple aux connaissances* : elle consiste soit à livrer une connaissance du domaine d'application (i.e la définition ou la fonction de l'élément de solution, une règle qui l'implique, ou une procédure dans laquelle il est utilisé), soit un but de conception directement satisfait par l'élément de solution référé.

Un but directement satisfait par l'élément de solution doit être exprimé sous forme d'une action sur cet élément de solution. Par exemple : "saisir la date de commande", "connaître la liste des fournisseurs". On considère habituellement 4 types d'action sur les données (tiré de Chen, 76) : les requêtes d'information, les opérations de création (ou saisie), mise à jour et suppression. Tout énoncé traduisant l'un de ces quatre types d'action est considéré comme une référence simple.

- *la référence à un groupement de connaissances* : cela s'applique surtout lorsque le concepteur livre une référence à un but de conception ; celui-ci peut regrouper un ensemble de sous-buts qui sont directement satisfaits par l'(es) élément(s) de solution référé(s).

Exemple : "(après la proposition des attributs : 'quantité livrée' et 'dates de livraison') Donc ainsi, on pourra suivre les prises de dette" ; le but "suivre les prises de dette" se décompose au moins en deux sous-buts "connaître la quantité livrée" et "connaître la date de livraison" qui sont pris en compte par les deux attributs proposés.

La figure 10 donne une représentation graphique de la référence simple à un but, et de la référence à un groupement de buts à partir de l'exemple cité ci-dessus.

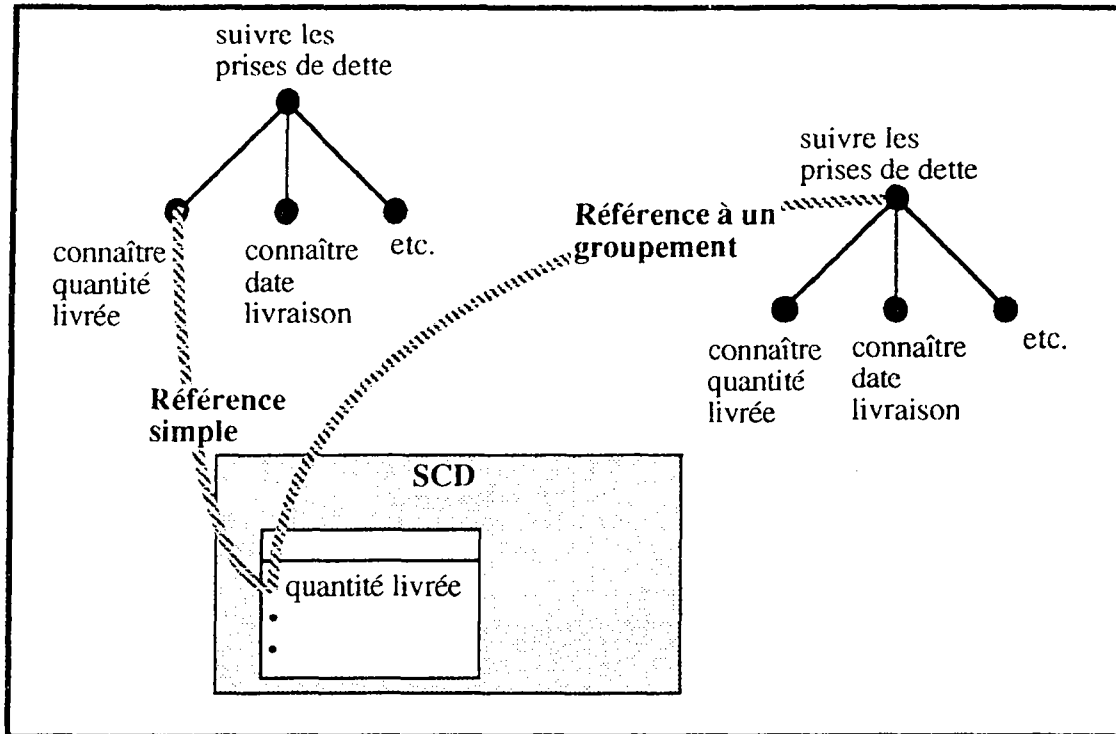


Figure 10 : La référence à un but, et la référence à un groupement de buts

Dans le cas d'une référence à un groupement de buts, le concepteur doit estimer que l'utilisateur possède la connaissance sur la décomposition d'un but donné. Dans le cas inverse, c'est à dire si l'utilisateur ne savait pas à quoi référait exactement le but explicite, la relation entre un ou plusieurs éléments du SCD et ce but n'aurait aucun sens.

On a donc une référence à un groupement de connaissances toute les fois que la proposition du concepteur sous-entend un ensemble de connaissances nécessaires pour la comprendre. Comme dans le cas de la référence à un groupement d'informations externes, l'avantage semble être de désigner un maximum d'informations avec la communication verbale la plus courte.

- *la référence étendue aux connaissances* : la référence étendue aux éléments du SCD vise à s'assurer que l'utilisateur comprend l'élément du SCD ; la référence étendue aux connaissances vise quant à elle, à s'assurer que l'utilisateur comprend la ou les connaissances ajoutées dans l'extension.

Par exemple, le concepteur peut faire référence à un but de la façon suivante (la citation a été simplifiée):

"Les gens de LOG veulent que l'agent connaisse le délai d'approvisionnement pour les vêtements qu'il commandera ; ceci permettra en partie d'éviter les ruptures de stock".

"Connaître le délai d'approvisionnement " est le but directement satisfait par l'attribut 'délai'. Ce but réfère alors à un but lié au fonctionnement de l'organisation : "éviter les ruptures de stock".

Un cas particulier de référence étendue aux connaissances consiste à livrer l'origine d'un fait ou d'un but .

Exemple : "L'agent, on mémorise le compte d'activité : d'après ce que m'a dit X, le compte d'activité c'est..." ; la définition qui va être donnée est donc justifiée en quelque sorte par X qui en est à l'origine.

On remarque alors que ces références étendues aux connaissances sont toujours explicitées lorsque l'origine d'un fait ou d'un but n'est pas l'un des utilisateurs en charge de valider la solution proposée. Très souvent d'ailleurs, cette origine externe au groupe d'utilisateurs est le concepteur : il a pu en effet introduire des buts nouveaux, ou inférer des faits du domaine d'application.

Or la validité d'une solution dépend avant tout sur la validité du ou des buts qu'elle est censée satisfaire et/ou du ou des faits sur lesquels elle repose. Lorsqu'une telle solution est proposée et qu'une référence étendue aux connaissances est livrée, on peut donc penser que le concepteur cherche à faire accepter en premier lieu ces connaissances (qui ont été désignées précédemment par le terme de contraintes de préférences - cf 1.4, p.6).

Il semble donc important de souligner que la validation d'un SCD, mais ceci devrait être retrouvé face à toute solution produite lors d'une activité de conception, recouvre deux niveaux : un niveau concerné par les éléments de solution à proprement parler, et un niveau concernés par les données sur lesquelles reposent ces solutions, lorsque ces données sont nouvelles pour l'utilisateur.

3.2.2.4 L'anticipation des demandes d'information et/ou de justification

Ce second niveau de référence qui amène à ajouter des connaissances non représentées sur le SCD vise à clarifier la signification des éléments référés et/ou à justifier leur utilité. Le fait intéressant est que le concepteur prend l'initiative de ces clarifications et de ces justifications ; il n'attend pas que l'utilisateur exprime une incompréhension. Autrement dit, on peut considérer que les références étendues reflètent une anticipation, de la part du concepteur, des demandes potentielles d'information et/ou de justification.

Ceci n'empêche cependant pas que dans certains cas, l'utilisateur adresse une demande d'information ou une demande de justification. Mais ces cas seraient vraisemblablement plus nombreux si le concepteur ne prenait pas l'initiative de fournir l'information nécessaire.

On peut comprendre le comportement du concepteur, donc l'intérêt d'anticiper les requêtes des utilisateurs, en se référant à une étude de Clark & Wilkes-Gibbs (86) portant sur le processus de référence : les sujets observés étaient placés de chaque côté d'un panneau opaque, sur lequel était aligné un ensemble de figures sans signification. Leur tâche consistait à les arranger dans le même ordre. Pour cela, ils devaient bien sûr faire référence à chaque figure. Dans certains cas, la référence exprimée ne permettait pas immédiatement au récepteur de reconnaître la figure dont il était question. La référence devait donc être modifiée pour que cet objectif soit atteint. Les auteurs observent alors que l'émetteur de la référence préfère modifier de lui-même sa référence, sans attendre d'intervention du récepteur. De cette façon, précisent-ils, il minimise d'une part le temps durant lequel le récepteur a pu ressentir l'incompréhension, d'autre part le nombre d'échanges nécessaires pour atteindre une référence mutuellement acceptable. Ainsi, il aide à *minimiser l'effort collaboratif*.

Ces conclusions sont à rapprocher du *principe de pertinence* proposé par D. Sperber & D. Wilson (86) : pour ces auteurs, toute personne engagée dans un dialogue tente de maximiser la pertinence de ses propositions, sachant que la "pertinence d'une proposition augmente avec le nombre d'inférences qu'elle permet de générer, et diminue avec le taux de traitement nécessaire pour y arriver" (p.249). Les auteurs assument que pour traiter une proposition émise, le récepteur commence systématiquement par faire des inférences dans un contexte immédiatement accessible, composée des propositions récemment traitées. Plus la compréhension d'une proposition nécessite de s'éloigner de ce contexte immédiatement accessible, plus le coût en terme de traitement augmentera. Le principe de pertinence consiste alors à dire que "le récepteur doit en conséquence faire l'hypothèse que l'émetteur a essayé d'être aussi pertinent que possible" (p.249).

Sperber & Wilson ajoute alors qu'un émetteur observant le principe de pertinence, et placé dans un cas où il estime que le contexte nécessaire pour comprendre une proposition n'est pas immédiatement accessible, doit faire quelque chose pour faciliter la compréhension - tel que fournir des informations supplémentaires - épargnant ainsi au récepteur un coût de traitement inutile.

3.2.2.5 Discussion

Etant donné ces éléments théoriques, on peut considérer que le concepteur soumettant des solutions à valider tente de réduire l'effort collaboratif, ou, dit d'une autre façon, d'être pertinent. L'ajout d'information n'est alors nécessaire que dans les cas où le concepteur estimerait que les utilisateurs n'ont pas toutes les informations requises pour émettre un jugement, ou que cela demanderait un coût élevé en terme de traitement.

Ces considérations permettent d'interpréter deux faits : d'une part, l'absence, parmi les clarifications et les justifications données spontanément, de buts liés aux règles de formalisation ; d'autre part les descriptions de la solution sous forme 'active' (i.e mêlant données et traitements) et les nombreuses références aux connaissances clarifiant l'utilisation des données représentées (i.e introduisant les actions sur ces données ; par exemple, les connaissances sur les procédures du domaine d'application ou les buts d'utilisation du système). Ces deux points sont développés ci-dessous.

Les buts liés aux règles de formalisation

On l'a dit en introduction, la validité d'une solution revêt au moins deux aspects : son adéquation aux besoins des utilisateurs, et le respect des règles de formalisation dictées par le modèle de données (ici, le modèle Entité-Association). On a observé que la proposition initiale du concepteur ne faisait jamais apparaître de connaissances ajoutées liées à ces règles. Ceci laisse à penser que la validation recherchée ne se situe qu'au niveau du premier aspect. On peut voir dans ces observations une application du principe du *moindre effort collaboratif* (Clark & Wilkes-Gibbs, 86) : l'utilisateur ne possède pas les connaissances liées à ces règles, et le concepteur le sait ; si ce dernier cherchait à faire valider l'application des règles de formalisation, il devrait d'abord fournir toutes les connaissances requises, ce qui nécessiterait d'une part un effort important de la part de ses interlocuteurs, afin d'assimiler ces connaissances nouvelles (lesquels pourraient en conséquence refuser la tâche soumise, par exemple en dirigeant le concepteur vers une personne ayant la compétence requise), et d'autre part un temps relativement long. Or, on doit souligner que les réunions de validation se font sous contrainte de temps.³

Ces constats ont des conséquences sur la conception d'un système d'aide engagé dans une phase de validation : toutes les règles utilisées pour générer la solution n'ont pas à être présentée en guise de justification. Si l'on accepte qu'une justification doit être *pertinente*, la pertinence étant liée aux connaissances de l'interlocuteur, alors le système devra d'une part identifier son utilisateur, d'autre part sélectionner les règles à présenter en liaison avec une solution.

Les propositions initiales mêlant données et traitement

Les données recueillies donnaient l'opportunité d'étudier l'éventuel effet de la distance entre la description statique des données (le SCD) et les représentations des utilisateurs, habituellement décrites comme des représentations mêlant données et traitements (tout au moins au niveau du langage). En réalité, nous n'avons pas pu constater directement cet effet, étant donné que le

³ C'est d'ailleurs une contrainte que Clark & Wilkes-Gibbs (86) ont recréé artificiellement, en donnant la consigne à leurs sujets d'ordonner les figures dans un temps minimum, ceci afin de se rapprocher des situations naturelles d'interaction verbale.

concepteur, dans un grand nombre de cas, décrivait la solution de façon dynamique, comblant automatiquement la distance potentielle.

Ceci dit, cette observation peut être prise comme un effet indirect de la distance entre les deux types de représentation : le concepteur sait que les utilisateurs interprètent ses propositions par rapport à leurs buts ; autrement dit, le *contexte immédiatement accessible* dont parle Sperber & Wilson (86) est constitué de ces buts. Pour faciliter la compréhension et le jugement des utilisateurs (ce qui a aussi pour conséquence un gain de temps), il ajouterait à sa proposition les connaissances sur les actions.

Pour certains éléments du SCD, ces connaissances ajoutées semblent nécessaires non pas pour faciliter la compréhension et le jugement, mais pour les permettre : c'est le cas de toute solution générée à partir de buts introduits par le concepteur ou par des sources d'information extérieures au groupe d'utilisateurs (par exemple le chef de projet), donc inconnus de ceux-ci. L'exemple suivant illustre ce cas :

C : Dans Point de Livraison, je rajoute l'information 'secteur'. Alors pourquoi j'ai rajouté l'information 'secteur'? Parce que ça sera surement plus pratique pour eux de pouvoir le saisir au moment où ils reçoivent les commandes, et de décider de la même façon qu'avant : je traite tel secteur ; et à ce moment là, c'est la machine qui fait le tri, qui ressort automatiquement toutes les commandes enregistrées pour ce secteur.

U : Oui, ça sera pas plus mal, oui.

Ici, le but introduit par le concepteur est l'automatisation de la tâche de tri des livraisons par secteurs. Comme on le voit, l'accord de l'utilisateur porte sur l'intérêt de ce but ; mais cet accord implique automatiquement l'accord avec l'élément de solution présenté.

Ces faits nous conduisent à penser que seule une représentation intégrant les actions sur les données peut réellement être compréhensible et validable par des utilisateurs. On pourrait alors objecter que dans un grand nombre de cas, le concepteur présente sa solution sous forme purement descriptive, et que cela semble suffire. En réalité, il n'y a pas de contradiction entre ces deux faits, si l'on considère que les cas où les actions ne sont pas explicitées reflètent des situations dans lesquelles le concepteur a estimé qu'elles pouvaient être aisément inférables. Autrement dit, la connaissance sur les actions serait toujours nécessaire pour comprendre et juger une solution, mais il ne serait pas toujours *pertinent* de les expliciter.

Il reste alors à déterminer les moyens qui permettent au concepteur de décider ou non d'ajouter des connaissances sur les actions. Cette question n'est pas traitée dans ce rapport mais constitue un de nos objectifs d'étude.

4 - CONCLUSION

L'étude avait deux objectifs principaux : évaluer le SCD en tant que représentation adaptée à la tâche de validation réalisée par des utilisateurs, et fournir des éléments de spécification d'un système d'aide à la conception de BD.

4.1 Vers une représentation adaptée aux utilisateurs

On a constaté le nombre très faible de demandes de justification de la part des utilisateurs et l'absence de demande d'information sur la signification des éléments du SCD. Ceci laisse à penser que la représentation devant laquelle sont placés les utilisateurs est compréhensible et en grande partie validable. Mais, en s'appuyant sur l'analyse de dialogues de validation, on a montré que le discours du concepteur ne plaçait pas les utilisateurs face à une représentation des données aussi abstraite et statique que le SCD : la représentation des données issue du discours du concepteur inclue des connaissances sur leur signification (définition, fonction d'un objet, etc.) et des connaissances sur les actions à prévoir sur ces données. Pour près d'un tiers des éléments du SCD (29% exactement), la connaissance représentée sur celui-ci n'est pas suffisante pour le faire valider.

Quelle serait alors le mode de représentation le plus adapté aux utilisateurs pour la tâche de validation? Le problème est qu'il n'existe pas un mode de représentation dans le discours du concepteur, mais deux :

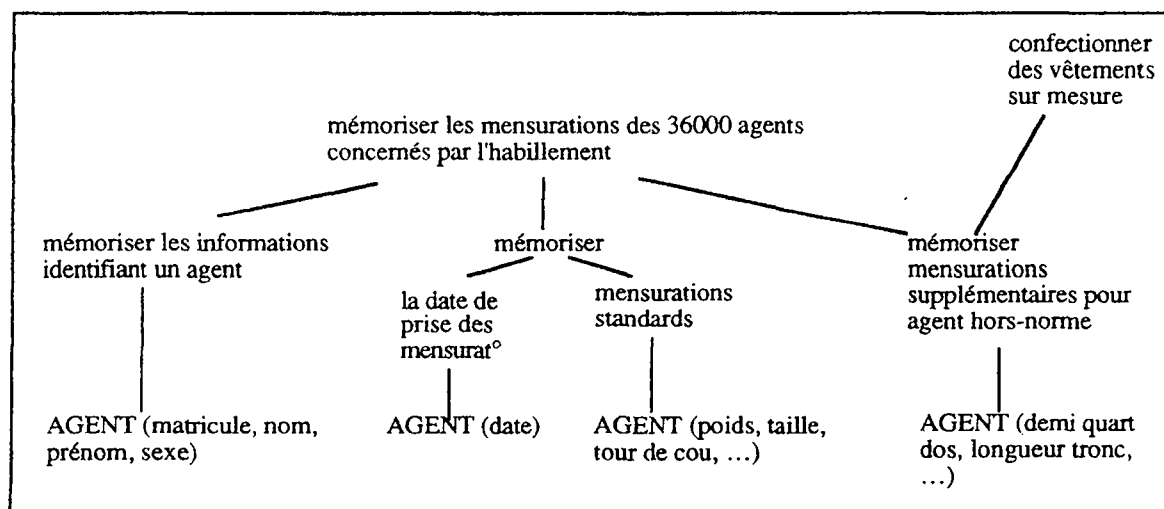
- un mode purement descriptif, basé sur la structure de données telle qu'elle est représentée sur le SCD, du type : "un vêtement, c'est un code, une désignation, ..., et un vêtement est disponible chez un fournisseur" ;
- un mode orienté par les buts du système, du type : "pour traiter les commandes exceptionnelles type BMM, on voit que j'ai 'type commande' dans Commande ; une commande est passée par 0 ou 1 agent, comme ça on est pas obligé de commander pour UN agent ; et dans ce cas, on note non pas le delta-taille, mais la taille du vêtement que l'on veut commander ; ce qui m'a conduit à rajouter : un vêtement existe en une taille..."

Cependant, notre proposition est de considérer que le concepteur utilise uniquement le mode descriptif toutes les fois où il juge que l'utilisateur peut comprendre l'utilité des données

présentées (i.e inférer les buts auxquels elles répondent). Autrement dit, le mode de représentation orientée par les buts serait parfois laissé implicite, mais toujours présent.

En outre, et en se basant sur les dialogues de validation recueillis, on peut se questionner sur l'intérêt de présenter aux utilisateurs un SCD formalisé avec le modèle Entité-Association : d'une part, celui-ci crée, dans certaines situations, une distance avec les représentations des utilisateurs. Par exemple, tandis que pour l'utilisateur, la commande (ce qui est apparentée au formulaire de commande) fait figurer un numéro de commande, une date, une quantité commandée, un matricule-agent, un code vêtement, etc., le SCD représente ces informations de façon disparate (cf fig.1, p.12). En outre, on l'a dit plus haut, cette formalisation n'est jamais justifiée (aucun but lié au modèle conceptuel n'est explicité) : le concepteur n'expliquera pas par exemple pourquoi il est nécessaire de mettre l'attribut 'quantité commandée' dans l'association Concerne plutôt que dans l'entité Commande. Par contre, il mettra en évidence la signification et/ou l'utilité de ces données.

Tous ces éléments donnent l'avantage à une représentation construite autour des buts de l'utilisateur. Par exemple, au lieu de représenter les données sur l'agent telles qu'elles le sont sur le SCD (fig.2, p.12), on pourrait envisager la représentation suivante (celle-ci correspond à la traduction fidèle du discours du concepteur dans une forme graphique) :



Ce type de représentation à l'avantage de permettre à l'utilisateur d'identifier directement l'utilité des données (représentées dans le dernier niveau, en bas), et reprend deux caractéristiques importantes recensées dans le discours du concepteur :

- tous les éléments de cette représentation ne sont pas justifiés ; par exemple, on n'indique pas pourquoi la 'date de prise des mensurations' est nécessaire. Ainsi la représentation reste *pertinente* (au sens de Sperber & Wilson).

- elle peut être lue avec différents niveaux de détail : en s'arrêtant au niveau juste précédant la description détaillée des données, on retrouve les regroupements exprimés par le concepteur.

Par contre, elle ne donne aucune information supplémentaire sur la signification des données. Il est difficile d'imaginer qu'une définition accompagnerait chaque donnée, ou tout au moins les données qui en nécessiterait une. Cela surchargerait la représentation, et la rendrait vraisemblablement moins lisible. Par contre, on peut imaginer que le concepteur fournisse un document supplémentaire correspondant à un lexique, et présentant, pour certaines données, des éléments de définition. L'utilisateur s'y reporterait, à son initiative, dès qu'il ne comprendrait pas la signification de l'une d'entre elles. Bien sûr, l'accès à ces définitions devant être facilité, on devra veiller à les ordonner logiquement, suivant l'ordre alphabétique par exemple.

Cette solution mixte, représentation des données orientée par les buts du système, et lexique des données, fournit donc théoriquement l'information nécessaire à la compréhension et au jugement de l'utilisateur, et garde théoriquement un aspect facilement lisible. Bien sûr, seul un test en situation réelle de validation pourrait réellement conduire à ces conclusions. Il devra donc être envisagé pour la suite de notre étude.

Malgré les avantages (théoriques) de cette description des données, on peut objecter le fait que la documentation à fournir serait plus volumineuse - alors qu'un des avantages du SCD est de donner une représentation très synthétique de la structure des données -, et plus précise, donc demandant une élaboration plus coûteuse en temps - ce qui contredit la plupart des situations de conception où le concepteur est pris de court par le calendrier fixé.

Ces inconvénients restent vrais tant qu'on reste dans l'optique d'une réunion de validation entre un concepteur et un groupe d'utilisateurs. Par contre, si l'on envisage l'interaction d'un utilisateur avec un système d'aide à la conception de BD, ils ne sont plus valables.

4.2 Eléments de spécification d'un système d'aide à la conception de BD

Cette étude a été envisagée dans l'optique de spécifier l'aide à prévoir dans le cas où un utilisateur non-initié en conception entreprendrait de réaliser une BD en coopération avec un système informatique. L'analyse des dialogues de validation nous permet de dresser un certain nombre de fonctionnalités que devrait posséder un tel système d'aide à la conception de BD pour engager la validation d'une représentation des données, et assurer son caractère coopératif :

1. *La nécessité de connaissances supplémentaires*

Les connaissances composant habituellement un SCD ne sont pas suffisantes pour permettre à l'utilisateur de valider les solutions proposées. Le système doit donc posséder un ensemble d'autres connaissances, recueillies au cours des séances interactives précédentes, et précisant la signification et l'intérêt des données retenues : ces connaissances peuvent regrouper des définitions de données, des procédures du domaine d'application ou des règles appliquées dans ce domaine, et les buts du système.

2. *Des propositions pertinentes*

De nombreuses études sur les systèmes experts reconnaissent la nécessité de stocker des connaissances qui n'apparaissent pas dans la représentation classique du programme, ceci afin de doter ces systèmes d'une capacité explicative suffisamment satisfaisante pour que l'utilisateur soit convaincu des solutions proposées (Swartout, 83 ; Clancey, 83 ; Kass & Finin, 88). Mais ces études considèrent que l'explication est donnée à la demande de l'utilisateur, lorsqu'un problème intervient (incompréhension, demande de justification, critique). L'étude présentée dans ce rapport conduit à proposer un autre schéma d'interaction : le système propose une solution en ajoutant, si il le juge nécessaire, certaines connaissances, réduisant ainsi les risques d'incompréhension. Il évite ainsi des traitements à l'utilisateur et réduit du même coup le nombre d'échanges nécessaires pour atteindre la validation. Le résultat est un système plus coopératif.

Même si le système a la possibilité de soumettre des propositions pertinentes, il doit bien sûr permettre à l'utilisateur de demander des informations supplémentaires (demande d'information et/ou demande de justification).

3. *Le relais de l'initiative*

Globalement, c'est le concepteur qui est à l'origine des propositions de solution. Le principal avantage est à envisager par rapport à ce qui vient d'être dit : en ayant l'initiative des propositions, il a l'initiative du choix des informations à présenter et des connaissances à ajouter pour les rendre plus compréhensible. Ainsi il cible le dialogue sur les problèmes importants et facilite la tâche des utilisateurs. Rappelons qu'en outre, une contrainte de temps s'exerce sur ces réunions. La prise d'initiative des propositions par le concepteur va aussi dans ce sens.

Ceci dit, l'utilisateur doit avoir la possibilité d'interrompre ce cours des choses : parce que le concepteur ne présente pas tous les éléments de solution, l'utilisateur doit pouvoir *l'interrompre* . A ce moment là, il doit pouvoir faire référence à un élément (ou éventuellement à plusieurs) et demander une clarification ou une justification, ou exprimer une critique.

4. Les négociations et l'analyse de nouveaux problèmes

La proposition de solution, même si elle est dotée de l'ensemble des connaissances permettant de la comprendre et de la juger, n'implique pas obligatoirement l'accord de l'utilisateur. Il faut donc permettre l'expression de contestation, d'autant que certains auteurs considèrent que l'émergence de conflits est favorable à la résolution de problèmes en groupe (Fisher, 74 cité dans : Walz & al., 1987) : les conflits ont pour effet d'augmenter la participation des membres du groupe et d'élargir le champ des solutions alternatives envisagées.

L'expression d'une contestation (critique ou solution concurrente) ne doit pas conduire directement à une modification de la solution dans le sens désiré par l'utilisateur : en effet, l'utilisateur peut émettre une critique, alors qu'en réalité il n'a pas compris la proposition du concepteur (l'estimation du concepteur sur ce que l'utilisateur sait et peut inférer facilement peut être incorrecte et conduire à des cas d'incompréhension) ; dans d'autres cas, sa critique peut révéler soit la non prise en compte d'un but non explicité mais prescrit auparavant (l'utilisateur a pu l'oublier), soit une contradiction avec un point de vue qu'il a adopté antérieurement ; ces situations expliquent la nécessité de recourir à une *argumentation* avant d'envisager une *modification*.

En outre, on observe que la résolution de conflits conduit l'utilisateur à exprimer de nouveaux buts. Le système doit donc être capable de les identifier et de les traiter : ceci revient à vérifier qu'ils ne contredisent pas les autres buts représentés, et à rechercher une solution.

4.3 Perspectives de recherches

On a assumé que le concepteur cherchait à émettre des propositions facilitant la compréhension des utilisateurs, et à intégrer la contrainte de temps. Ceci le conduit à sélectionner les éléments à mentionner, éventuellement à les regrouper, et dans certains cas à ajouter des connaissances non représentées sur un SCD exprimé dans un modèle Entité-Association. Il applique ainsi des principes liés à la coopération. Mais quels sont les éléments qui permettent à ces principes d'être appliqués?

Plus généralement, cette question doit nous amener à considérer le problème du *contrôle de l'activité*. Celui-ci s'applique à différents niveaux, déterminant :

1. l'ordre des propositions de solution ;
2. le type de référence aux éléments du SCD ;
3. le choix des connaissances à ajouter si ce type de référence est retenue ; en effet, même si le concepteur décide de justifier sa solution, il peut le faire de plusieurs façons : énoncer le but

d'utilisation qu'elle satisfait, décrire la procédure qui l'amène à l'utiliser, présenter un exemple d'utilisation, ou simplement exprimer l'origine (telle qu'on l'a défini dans la classification des connaissances ajoutées) si celle-ci est reconnue pour sa compétence.

4. l'ordre des connaissances à ajouter, si plusieurs connaissances sont jugées nécessaires pour clarifier et/ou justifier un même élément de solution. Il semble en effet que l'ordre des arguments dans une explication puisse faciliter la compréhension, mais aussi accentuer le caractère convaincant de l'argumentation (Mc Keown, 1985).

Une étude est engagée pour répondre à ces différents problèmes. Un élément semble déterminant pour comprendre ces différents choix (2, 3, et 4) : il s'agit du modèle des utilisateurs. Ceci apparaît notamment lorsque l'on compare les propositions données à un groupe et à l'autre : on observe par exemple que pour justifier un élément de solution, le concepteur fera référence uniquement au but du système qu'il satisfait face aux utilisateurs fréquents, alors qu'il décrira d'abord la procédure appliquée dans le domaine d'application qui justifie le but du système, avant d'y faire référence face aux utilisateurs occasionnels. Comme le fait remarquer B. Cahour (91), la pertinence d'une proposition dépend de l'utilisateur à qui elle s'adresse.

L'autre champ d'étude qu'il semble important d'approfondir concerne les dialogues de négociation. Une approche intéressante est développée par Klein & Lu (89) pour traiter ce problème : ils mettent l'accent sur une identification précise du type de conflits, lesquels sont représentés sous forme d'une classification. A chaque type de conflit est alors associée une stratégie de résolution du conflit (par exemple, abandonner l'un des buts impliqués dans le conflit, tenter une proposition alternative, ajouter des détails à la solution, ou trouver un compromis qui satisfasse seulement en partie les buts impliqués dans le conflit).

Sycara (88) considèrent pour sa part deux modalités de réponse à un conflit qui se rapprochent plus directement de nos observations : *l'argumentation*, qui a pour but de changer l'évaluation du critiqueur afin qu'il accepte la proposition, et la *modification* de la proposition dans le cas où l'argumentation échoue.

Le point commun des deux approches est qu'elles s'appuient, de nouveau, sur un modèle des acteurs impliqués dans la négociation. Pour Klein & Lu (89) par exemple, on abandonnera un but à la demande d'un des acteurs, uniquement si cet acteur a l'expertise spécifique appropriée pour prendre cette décision. Quant à Sycara (88), elle propose un système expert en résolution de conflit, PERSUADER, qui utilise un modèle des intentions de l'interlocuteur, notamment pour générer l'argumentation.

BIBLIOGRAPHIE

- Breutmann, B., Falkenberg, E., & Mauer, R. (1979).** CSL: A Language for Defining Conceptual Schemas. In: Bracchi & Nijssen (Eds), *Data Base Architecture*. North Holland, IFIP.
- Cahour, B. (1991).** *La Modélisation de l'Interlocuteur*. Thèse de doctorat. Université Paris VIII.
- Chen, P.P. (1976).** The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1), pp. 9-36.
- Clancey, W.J. (1983).** The Epistemology of a Rule-based Expert System - a Framework for Explanation. *Artificial Intelligence*, 20, pp. 215-251.
- Clark, H. H. & Wilkes-Gibbs, D. (1986).** Referring as a collaborative process. *Cognition*, 22(1), pp. 1-35.
- Darses, F. (1990).** *Gestion de contraintes au cours de la résolution d'un problème de conception de réseaux informatiques* (Rapport de recherche n°1164). Rocquencourt : INRIA.
- Dasgupta, S. (1989).** The Structure of Design Processes. *Advances in computer*, 28, pp.1-67.
- Eastman, C.M. (1969).** Cognitive processes and ill-defined problems: a case study from design. *Proceedings of First Joint International conference on Artificial Intelligence*. Washington D.C.
- Elmasri R. & Navathe S.B. (1989).** *Fundamentals of Database Systems*. Redwood, CA : Benjamin Cummings.
- Elmasri R., Weeldreyer, J. & Hevner, A. (1985).** The Category Concept : An Extension to the Entity-Relationship model. *Data Knowledge Engineering*.1(11), pp. 75-116.
- Flory, A. (1982).** *Bases de Données. Conception et réalisation*. Paris : Economica.
- Guindon, R. (1990).** Knowledge Exploited by Experts during Software System Design. *International Journal of Man-Machine Studies*, 33, pp. 279-304.

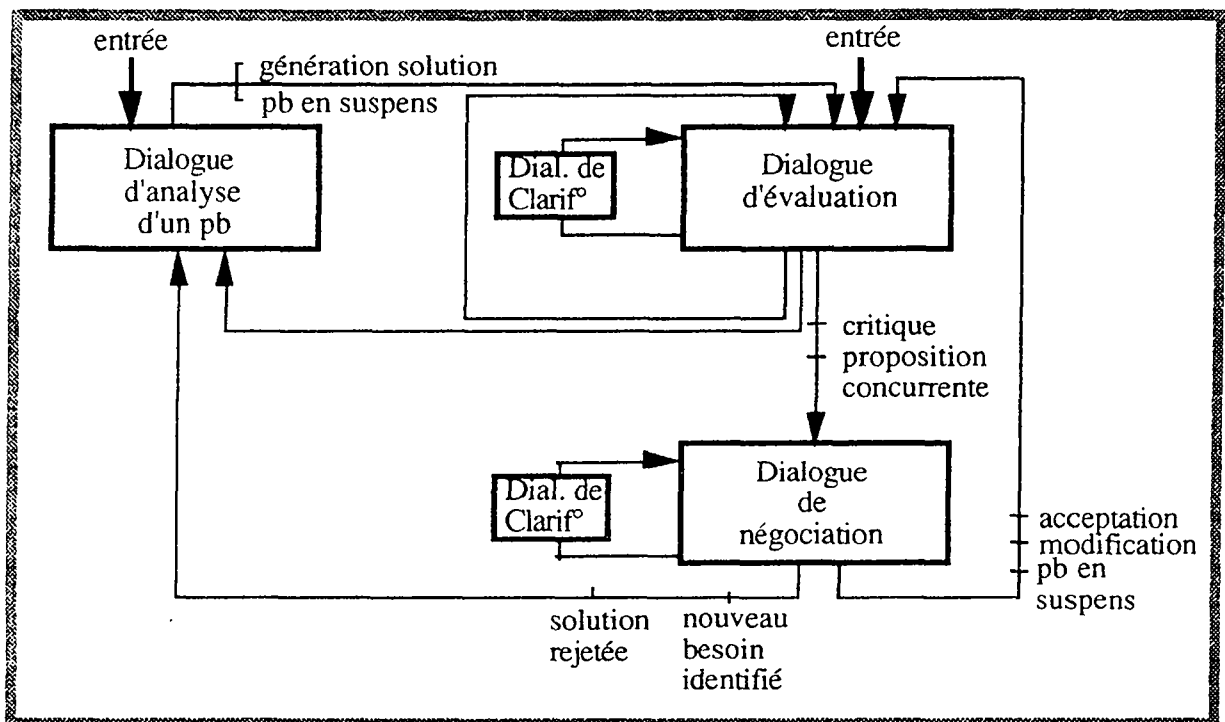
- Hayes, J.R., Flower, L.S., (1980).** Identifying the organisation of writing processes. In: L.W.Gregg, E.R.Steinberg (Eds) - *Cognitive process of writing*. Hillsdale, N.J. : Erlbaum.
- Janssen, P., Jégou, P., Nouguier, B., Vilarem, M.C. (1989).** Problèmes de conception : une approche basée sur la satisfaction de contraintes. *9èmes Journées Internationales : Les systèmes experts et leurs applications*, 29 mai-2 juin, **1**, 71-84.
- Jeffries, R., Turner, A.A., Polson, P., Atwood, M.E. (1981).** The processes involved in designing software. In : J.R.Anderson, (Ed.) - *Cognitive Skills and their Acquisition*. Hillsdale, N.J. : Erlbaum.
- Kangassalo, H. (1988).** Concept D : A Graphical Language For Conceptual Modelling and Data Base Use. *IEEE*.
- Kant, E., Newell, A. (1984).** Problem solving techniques for the design of algorithms. *Information Processing and Management*, **28**, pp. 97-118.
- Kass, R. & Finin, T. (1988).** The need for user models in generating expert system explanations. *International Journal of Expert Systems*, **1**(4), pp. 345-375.
- King, R. & McLeod, D. (1985).** Semantic Data Models. In: S. Bing Yao (Ed.), *Principles of Database Design*, vol.1. Prentice-Hall, N.J.
- Klein, M. & Lu, S.C.Y. (1989).** Conflict resolution in cooperative design. *Artificial Intelligence in Engineering*, **4** (4), 168-180.
- Malhotra, A., Thomas, J.C., Carroll, J.M., Miller, L.A. (1980).** Cognitive processes in design. *International Journal of Man-Machine Studies*, **12**, pp.119-140.
- Mc Keown, K. M. (1985).** Discourse Strategies for Generating Natural-Language Text. *Artificial Intelligence*, **27**, pp. 1-41.
- Olle, T.W. & co-auteurs (1990).** *Méthodologies pour les systèmes d'information*. Bordas et IFIP, Paris.
- Rolland, C., Foucaut, O., Benci, G. (1988).** *Conception des systèmes d'information : la méthode REMORA*. Paris, Eyrolles.

- Rowe, P.G. (1987). *Design Thinking*. Cambridge, MA: MIT Press.
- Simon, H.A. (1973). The structure of ill-structured problems. *Artificial Intelligence*, 4, 181-201.
- Simon, H.A. (1981). *Sciences of the Artificial* (2nd Ed.) Cambridge, MA: MIT Press.
- Sperber, D. & Wilson, D. (1986). Inference and Implicature in Utterance Interpretation. In: T. Myers, K. Brown & B. Mc Gonigle. *Reasoning and Discourse Processes*. Academic Press, London.
- Swartout, W. (1983). XPLAIN: a System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence*, 21, 285-325.
- Sycara, K. (1988). Resolving Goal Conflicts via Negotiation. *AAAI 88. Proceedings of the 7th National Conference on Artificial Intelligence*, vol.1, 245-250.
- Tardieu, H., Rochfeld, A., Coletti, R. (1986). *La méthode MERISE*. Paris : Les éditions d'organisation.
- Teorey, T.J. (1990). *Database Modeling and Design. The Entity-Relationship Approach*. San Mateo, CA : Morgan Kaufmann.
- Verheijen, G. & van Bekkum, J. (1982). NIAM : an Information Analysis Method. In : T.W Olle, H. Sol & A.A. Verrijn-Stuart (Eds), *Proceedings of the First IFIP CRIS Conference*, North-Holland.
- Visser, W. (1987). Strategies in programming programmable controllers : a field study on a professional programmer. In : G. Olson, S. Sheppard & E. Soloway (Eds.). *Empirical Studies of Programmers*. Second Workshop, NJ: Ablex.
- Walz, D., Elam, J., Krasner, H., & Curtis, B. (1987). A methodology for studying software design teams : an investigation of conflict behaviors in the requirements definition phase. *Proceedings of the 2nd Workshop on Empirical Studies of Programmers*. Norwood, NJ: Ablex.

ANNEXE

La structure du dialogue de validation

Cette annexe reprend un point abordé en 3.1.5 et détaille les actions engagées à l'intérieur de chaque type de dialogue, ainsi que les transitions entre eux. La figure ci-dessous synthétise la structure complète du dialogue de validation.



- le dialogue d'évaluation : le concepteur est, à l'exception de deux cas, à l'initiative de ce type de dialogue, qui se traduit par la *proposition* d'un élément de solution suivie d'une *évaluation* des utilisateurs.

L'utilisateur peut manquer d'informations pour évaluer la solution, ce qui se traduit soit :

- par une *demande d'information* - dans les cas où l'information recherchée vise à compléter la représentation du problème ; on remarque que ces requêtes n'ont quasiment jamais de lien direct avec une donnée ; elle porte le plus souvent sur les traitements ; exemple : "les informations dans les CEA, quand est-ce que vous les traitez? en temps réel ou en temps batch ?" ;

- soit par une *demande de justification*. - lorsque l'utilisateur ne comprend pas l'intérêt d'une solution proposée ; exemple : "mais pourquoi on a besoin de BMM dans cette application?".

Ces situations se traduisent donc par des requêtes qu'il adresse au concepteur, lequel fournit en réponse l'information recherchée. Les séquences : requête d'information utilisateur / apport d'information concepteur, caractérise le dialogue de clarification : il vise donc à enrichir la proposition initiale de façon à ce que celle-ci soit effectivement évaluable par les utilisateurs.

L'évaluation d'un utilisateur peut conduire soit à :

- accepter la proposition. Il faut distinguer l'accord implicite et l'accord explicite :

◇ il y a *accord implicite* toutes les fois que l'utilisateur laisse le concepteur enchaîner sur une autre proposition sans intervenir, ou lorsqu'il enrichit spontanément la proposition du concepteur comme dans l'exemple suivant (C : concepteur, U : utilisateur) :

C : donc, le catalogue, ça peut être un des catalogues principaux, ou alors un sous-catalogue créé par le responsable qui dit : mes agents doivent prendre leurs vêtements là-dedans...

U : dans cette liste-là.

C : voilà. (suite)

◇ il y a *accord explicite* dans les cas où l'utilisateur l'exprime, en général de la façon suivante : "Ok", "c'est ça", "oui", ...

L'accord, implicite ou explicite, conduit le concepteur à exprimer la proposition suivante.

- contester la proposition. Cette contestation peut prendre deux formes :

◇ *critique de la solution* : dans ce cas, l'utilisateur exprime un désaccord avec la proposition et justifie ce désaccord : il montre alors un inconvénient de la solution proposée.

◇ *proposition concurrente* : dans ce cas, l'utilisateur propose une solution alternative.

La contestation d'une proposition conduit le couple utilisateurs-concepteur dans un dialogue de négociation.

- le dialogue de négociation : négocier, c'est tenter de résoudre un conflit. Le conflit apparaît après l'expression d'une contestation. On l'a vu, la contestation peut prendre deux formes : la critique, ou la proposition d'une solution concurrente.

Les actions qui vont suivre l'émergence d'un conflit vont dépendre en partie du type de conflits :

- dans le cas d'une CRITIQUE, le concepteur peut soit :

1.1 accepter la critique : dans ce cas, il peut soit directement modifier la solution pour annuler le conflit, soit analyser le problème si l'utilisateur a fait état dans sa critique d'un but ou d'une contrainte inconnu(e) du concepteur, soit enfin re-analyser le problème à l'origine de la solution critiquée, si le concepteur n'envisage aucune alternative, ni aucune modification qui annule le conflit ;

1.2 rejeter la critique en montrant soit que la solution prend en compte le but du critiqueur, soit qu'elle présente moins d'inconvénients que la solution alternative ; le critiqueur peut être d'accord, ce qui annule la critique, ou refuser l'argument du critiqué en avançant un nouvel argument ; ce type de situation n'a été observé qu'une seule fois : le concepteur a accepté le nouvel argument et décidé de modifier sa solution ;

1.3 signaler une contradiction : dans un cas unique, le concepteur signale que l'argument sur lequel est basée la critique de l'utilisateur n'a pas été pris en compte par ce même utilisateur alors qu'il validait un élément de solution précédent ; l'utilisateur justifiera alors sa décision de validation, en montrant que la solution validée respectait un autre but important ; la contradiction est annulée ; le concepteur a alors accepté la critique (on est ramené au cas 1.1).

- dans le cas d'une SOLUTION CONCURRENTE, le concepteur peut soit :

2.1 critiquer la solution concurrente : on retrouve alors exactement les mêmes séquences que dans la situation précédente, excepté que les rôles sont inversés ;

2.2 accepter la solution concurrente : ceci conduit le concepteur à modifier sa solution ;

2.3 accepter de façon conditionnelle la solution concurrente : le concepteur relève un inconvénient mineur de la solution, qui demande à être accepté par une instance extérieure au groupe de validation ; le problème reste momentanément en suspens.

2.4 demander une confirmation : la représentation conceptuelle des données représente le plus souvent des faits propres à l'organisation. La solution est en partie valide si ces faits sont eux-mêmes valides. Lorsque l'utilisateur propose une solution concurrente, le concepteur peut en inférer les faits à l'origine. Par exemple, quand l'utilisateur propose de "mettre la valeur-point dans le Catalogue", le concepteur infère que la valeur-point est propre à chaque catalogue identifié dans le domaine d'application. Ce type de fait inféré peut contredire la représentation mentale du domaine d'application que le concepteur a construit. Dans ce cas, il va demander une confirmation du fait inféré.

Dans la seule situation où cette éventualité s'est présentée, le fait inféré n'a pas été confirmé (la valeur-point n'est pas propre à chaque catalogue, mais commune pour tous les catalogues). Ce qui invalidait la solution concurrente proposée (on est ramené au cas 2.1 : critique de la solution concurrente).

- le dialogue d'analyse d'un problème : une activité de conception est caractérisée, entre autres, par la nécessité de résoudre des problèmes mal-définis. Une partie importante de l'activité est ainsi destinée à rechercher les informations manquantes qui vont permettre de préciser les buts de conception (Simon, 73). Certaines solutions proposées par le concepteur reposent encore sur des problèmes insuffisamment définis, insuffisamment si l'on cherche à s'assurer qu'elles répondent bien aux besoins des utilisateurs. La nécessité d'analyser un problème, i.e de rechercher des informations manquantes, apparaît dans deux types de situation :

cas 1 : lorsque le concepteur a besoin de faire confirmer une connaissance hypothétique dont dépend la validité d'un élément de solution. Une connaissance hypothétique peut soit être une connaissance qui lui a été transmise, mais dont il n'est pas sûr, soit une connaissance qu'il a inféré à partir de ses propres connaissances ;

cas 2 : si un nouveau besoin a été élicité, notamment au cours d'une négociation dans laquelle l'utilisateur peut faire état d'un besoin non connu du concepteur ; ou si une solution critiquée est rejetée, et que le concepteur n'envisage aucune solution alternative ;

Les caractéristiques du dialogue d'analyse d'un problème dépendent en partie du type de situation qui l'a engendré. Certaines, cependant, sont communes :

- ce type de dialogue est toujours initié par une requête du concepteur dirigée vers les utilisateurs ;

- si l'utilisateur ne génère aucune information (i.e si il dit ne pas connaître la réponse), le concepteur reformule toujours sa question (dans les cas observés) en incluant un exemple ; par exemple, la requête initiale "est-ce qu'un même CEA peut être livré à différents endroits?", a été reformulée comme suit : "par exemple, est-ce que les lignes de métro (une ligne de métro correspond à un CEA) sont livrées à leurs deux terminus et au secrétariat de ligne, ou est-ce qu'on livre tout au secrétariat ?" ;
- toute nouvelle information générée par les utilisateurs est prise en note par le concepteur ;

Certaines actions diffèrent suivant que l'on soit dans l'un des deux cas :

- dans le cas 1, la requête du concepteur correspond en fait à une *demande de confirmation* portant sur une connaissance hypothétique ; celle-ci peut être vraie ou fausse. La réponse attendue est donc soit une confirmation, soit une infirmation. Seuls des cas où l'utilisateur confirmait la connaissance hypothétique ont été observés. La confirmation peut être suivie de précisions sur le problème. Ces précisions peuvent refléter de nouvelles connaissances importantes à prendre en compte pour le développement de la solution.
- dans le cas 2, les requêtes du concepteur reflètent une *connaissance de la structure du problème* posé et la *recherche d'une solution* à ce problème.
Par exemple, alors qu'une solution initiale pour régler un problème de rupture de stock a été rejetée, le concepteur cherchera à connaître le délai de réapprovisionnement des vêtements stockés ; l'information recherchée révèle ici la connaissance selon laquelle la rupture d'un stock peut dépendre du délai de réapprovisionnement.
Dans un autre cas, après l'élicitation d'un nouveau besoin (traiter de façon informatique la commande des vêtements pour les agents hors-normes), le concepteur cherchera à mieux définir le problème en posant des questions du type : comment le fait-on actuellement ?, où trouve-t-on les informations nécessaires ?, etc. Ces questions révèlent là-aussi une connaissance de la structure du problème posé par l'automatisation d'une tâche manuelle.

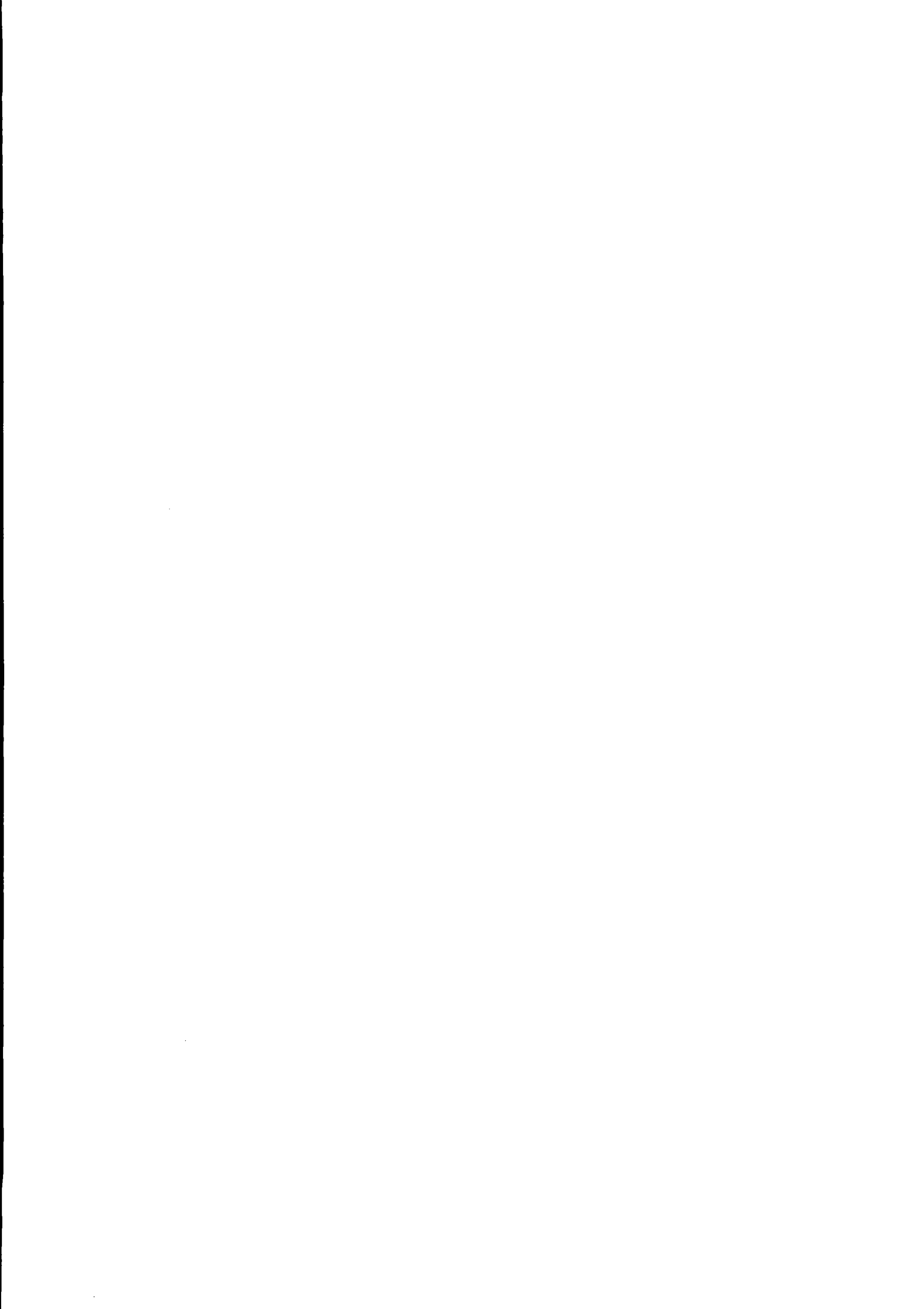
Bien que les requêtes d'information soient formulées par le concepteur, cela n'implique pas qu'il soit seul pour rechercher une solution. Au contraire, c'est le plus souvent l'utilisateur qui en proposera une, éventuellement en la justifiant.

Dans la plupart des cas, on aboutit à la *génération d'une solution*. Les membres du groupe cherche ensuite à évaluer la solution proposée (on est ramené au dialogue d'évaluation).

Dans un cas, le problème restera *en suspens*, le concepteur manquant d'informations pour décider d'une solution.

En résumé, le dialogue d'analyse d'un problème révèle les trois faits suivants :

- 1) le concepteur n'aborde pas une séance de validation avec un ensemble de problèmes bien définis ;
- 2) certains problèmes restent insuffisamment définis à l'issue de la validation du SCD, et demandent que soient consultées d'autres sources d'information ;
- 3) tous les besoins n'ont pas été élicités aux cours des réunions précédant une réunion de validation.



ISSN 0249 - 6399