

Conception de Bases de Données Orientées objet : Approche basée sur l'analyse des Formulaires Et des Scénarios de Transactions

Rafik BELKHODJA* - Mimoun MALKI* - Lamia ZENAIID *
Abdelkader GAFOUR **

*LRISI, Institut d'Informatique, Université Djillali Lyabes Sidi-Bel-Abbes 22000 Algérie.
BELKHOUDJA@yahoo.com

** ACL, Institut d'Informatique, Université Djillali Lyabes Sidi-Bel-Abbes 22000 Algérie.

Résumé: Ce papier présente un processus d'enrichissement de conception de bases de données orientées objet basé sur une analyse couplant les formulaires et les scénarios de leurs transactions. L'analyse des formulaires permet la dérivation de la structure statique des objets du Système d'information alors que l'analyse des scénarios de transactions permet, tout en fournissant un schéma comportemental inter-objets, d'encapsuler les méthodes dans le modèle statique. Un scénario de transactions est défini à partir des transactions d'un ou plusieurs formulaires. Une analyse des liens de dépendance entre ces scénarios a permis de les catégoriser et de les utiliser pour modéliser les liens entre les objets dérivés de ces formulaires. Le processus décrit comporte trois étapes: l'analyse statique qui produit le schéma objet, l'analyse dynamique qui représente le comportement inter-objet et la validation et l'encapsulation du schéma objet statique. Le modèle sous-jacent est dénommé ACE, permet la représentation des objets selon trois plans : Structuration (A), Communication (C), et Héritage (E).

Mots clés : *Conception de Base de Données Orientées Objet, Formulaire, Document, Interface de Schéma, Scénarios, Transactions.*

1 - Introduction

L'analyse des documents est généralement prise en compte dans la phase de recueil de l'information. Cette analyse consiste, d'une part en l'extraction de la partie lexicographique des données, et d'autres part en l'extraction d'ébauche de structure abstraite contenant la description des objets décrits par les documents et les procédures de manipulation des objets. Ce processus de recueil est la première tentative de spécification et de modélisation pour définir une démarche systématique et engendrer un cadre utile à un produit de génie logiciel. Bien que quelques travaux sont actuellement significatifs :

- Dans [Batin,84] Batini et al., proposent une méthodologie de conception à partir des formulaires exploitant essentiellement leur aspect statique.
- Dans [Choob,92] Choobineh et al., proposent un modèle de formulaires qui définit les propriétés des champs du formulaire et les dépendances (fonctionnelles et multivaluées). Un système expert est défini permettant de créer un schéma conceptuel par l'intégration incrémentale d'une collection de documents d'une application.

Alimazighi et al. [Alima,97] ont utilisé l'analyse des formulaires couplée aux scénarios des cas d'utilisation. leur approche représente le scénario par une structure tabulaire pour modéliser l'aspect dynamique. Drake et al. [Drake,92] utilisent, les diagrammes de flux de données (DFD) pour la transformation et les diagrammes états-transitions (STD) pour le contrôle. L'objet de notre travail est de définir un environnement de modélisation de bases de données Orientées Objet à partir de formulaires et des scénarios de leurs transactions.

Cette approche permet de saisir les documents structurés et les scénarios de transactions, de les stocker et d'en extraire des connaissances utiles à la modélisation conceptuelle.

Le système projeté offre les fonctionnalités principales suivantes (Fig. 1) :

- Acquisition de l'image et la structure du document ;

- Modélisation du document en schéma objets ;
- Analyse des scénarios de transactions ;
- Intégration des schémas d'objets des documents en un seul schéma conceptuel.

Ce papier est organisé de la façon suivante : Dans la section suivante, nous décrivons l'interface graphique d'acquisition des documents. Le processus d'acquisition est présenté dans la section 3. Les deux niveaux d'analyse statique P1 et dynamique P2 sont traités respectivement dans les sections 4 et 5. L'intégration P3 est dans la section 6 et la conclusion dans la section 7.

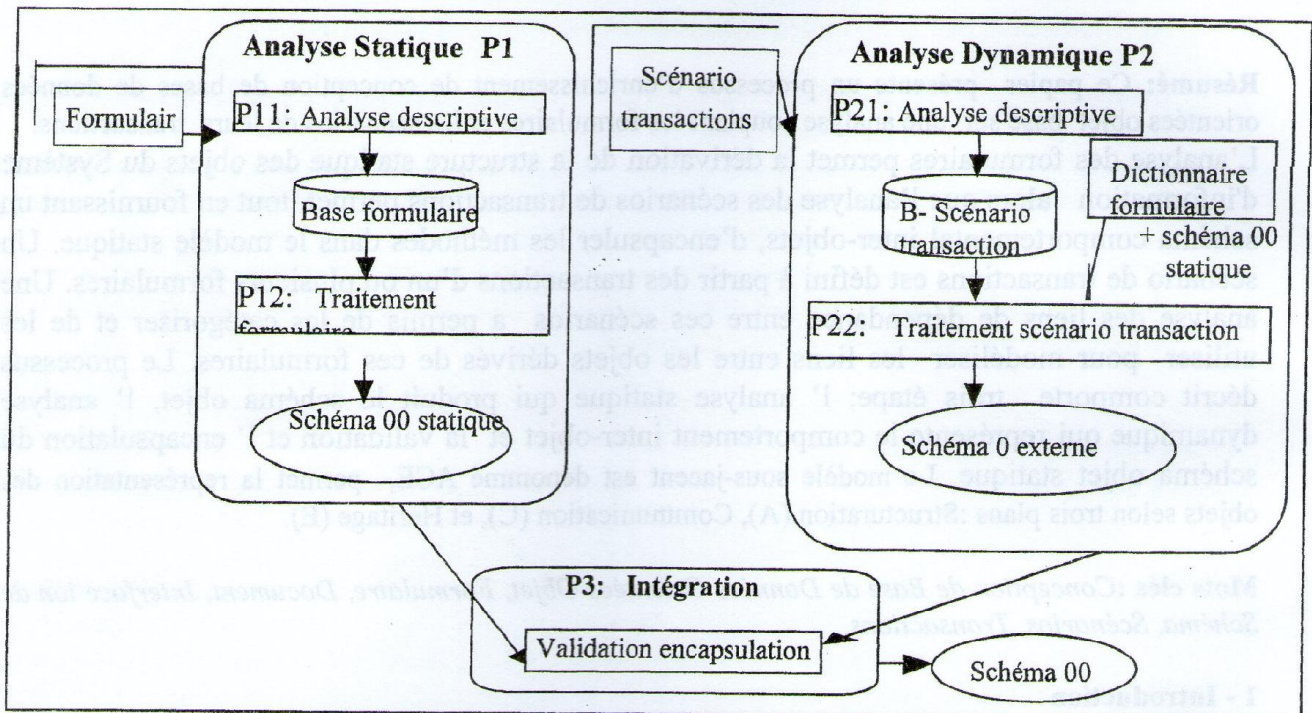


Fig. 1 : Architecture Fonctionnelle du Système proposé

2 - Interface Graphique d'Acquisition de Documents

Un document est vu comme une grande boîte composée de blocs texte, de régions graphiques, de tableaux, de champs d'informations et aussi d'autres boîtes (construction récursive). La boîte et le tableau sont considérés comme structure de composition. Le champ d'information représente une région de remplissage dans un document [Belkh,99].

ENIE Zone Industrielle Sidi Bel Abbes 22000 SBA ALGERIE			FACTURE Date: N°:		
CLIENT Code: Raison Sociale:			COMMANDE Date: N°-Cde: N°-Représent.: Nom:		
TRANSPORTS Mode: Origine: Destination:			LIVRAISON Date: Condition:		
N°	C-prod	Désignation	P.U.	Qté.	Montant
Montant Hors Taxe :			Montant TVA :		
Net à Payer :					

Fig. 2 : Exemple de Formulaire : une Facture

L'éditeur graphique permet de décrire le contenu d'un document en utilisant trois concepts de base : boîte, tableau et champ d'information. La figure (Fig. 2) illustre un document facture rangé en boîte et en tableau. Les concepts (champs, tableaux, boîtes), utilisés par l'interface graphique, permettent de spécifier le contenu du document dans un langage de description de documents. Nous proposons un langage pour représenter les documents indépendamment des modèles de données. Les concepts du langage sont décrits au travers la syntaxe suivante [Malki-97] :

Champ : <Nom-Champ>
Identifiant : <Identifiant-Champ>
Type : Entier/réel/Booléen/caractère
Longueur : <Entier>
Origine : Système/Utilisateur/Calculé<expression-calcul>

Fin-Champ

Tableau : <Nom-Tableau>

Identifiant : <Identifiant-Tableau>
colonne : 1/N
Ligne : 1/N
Champ-Colonne : <Non-Champ>{,<Non-Champ>}
ou
Champ-Ligne : <Non-Champ>{,<Non-Champ>}

Fin-Tableau

Boîte : <Nom-Boîte>

Identifiant : <identification-Boîte>
Composition : {<Identification-Champ>, Bloc-Texte,
{Bloc-Graphique,{<Identification-boîte>}}}

Fin-Boîte

Un **champ d'information** est un attribut simple qui représente une information de type de base (entier, réel, booléen, texte). Un attribut est d'origine différente (système, utilisateur et calculé). A chaque champ est associé un identifiant (interne unique).

Un **tableau** est une abstraction qui permet d'associer des champs. Un tableau est composé de lignes et de colonnes. A chaque colonne ou ligne est associé à un ou plusieurs champs.

Une **Boîte** est une abstraction qui permet d'associer un bloc-texte, un bloc-graphique, un tableau, des champs et des autres boîtes.

L'interface graphique conduit à la reproduction du document enrichi d'une spécification plus formelle de ces objets composants avant de passer à la modélisation du modèle de données.

3 - Acquisition des Documents

Le processus d'acquisition consiste à saisir les informations générales du document, son image graphique, les informations de ses composants et le dictionnaire des données contenu dans les documents. Ce processus est réalisé par trois opérations : insérer un document, décrire le contenu du document et élaborer un dictionnaire des données.

3.1 - Opération d'insertion d'un Document

Cette opération consiste à décrire globalement le document (titre, sous-titre, type, résumé, propriété, l'état date d'acquisition et version du document), lui donner une identification unique par rapport aux autres et générer son image qui est obtenue à partir de l'interface graphique décrite précédemment.

3.2 - Opération de description du contenu du document

Afin de faciliter l'opération d'interprétation et de modélisation du document, on transforme les concepts du document utilisés dans l'interface graphique en une structure arborescente dont les nœuds représentent les objets du document et les arcs représentent les liens de composition. Le Document est le nœud composé racine à laquelle toutes les informations sont attachées. Un champ d'information est vu comme un objet atomique, le tableau et la boîte sont vus comme des objets composés.

Il y a deux types d'arcs : l'arc **p** lie un objet composé à ses composés atomiques et l'arc **o** lie un objet composé à ceux de ses composés qui sont eux même composés (Fig. 3) :

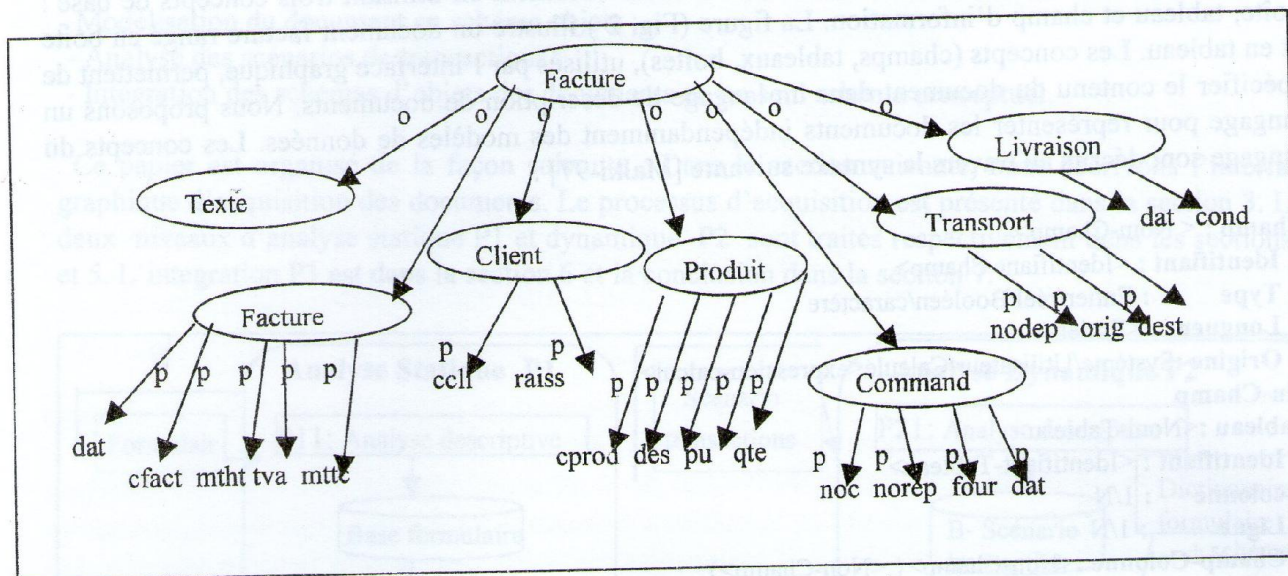


Fig. 3 : Arborescence du document facture

3.3 - Elaboration d'un dictionnaire des documents

Pour assurer une description aussi complète que possible du champ d'application, nous utilisons un dictionnaire qui est mis à jour au fur et à mesure que nous accueillons un nouveau document. La structure du dictionnaire est : Terme, définition, liaisons sémantiques et formules de calcul.

4 - Modélisation des Documents

Cette phase consiste à faire une interprétation sémantique des formulaires individuellement dans le contexte de l'approche orientée objet. Elle regroupe un ensemble de procédures qui permettent d'analyser les éléments composants le document et leur donner une interprétation sémantique et structurelle. Ce processus de modélisation commence par simplifier le contenu du formulaire. A partir de cette simplification nous modélisons la structure du formulaire : **identification** des classes d'objets et **attachement** de leurs propriétés (attributs). Les contraintes d'intégrités associées aux éléments du document (dépendance fonctionnelle, dépendance multivaluée, unicité et non nullité) sont également identifiées dans ce processus.

4.1. Présentation des Concepts du modèle ACE

L'approche que nous proposons pour modéliser une base de données à partir de formulaires, utilise le modèle ACE proposé dans [Ayach,96]. Le modèle ACE est d'abord un modèle de représentation car il permet de spécifier le schéma objet d'un système, appelé O-schéma, suivant trois plans : structuration(A), communication(C) et héritage(E) où chaque composante peut être décrite indépendamment des autres. Les données sont représentées dans le plan structuration. La dynamique est représentée dans le plan communication et les liaisons d'héritage entre les objets sont représentés dans le plan héritage.

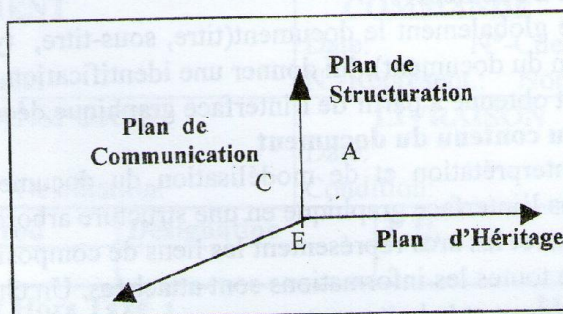


Fig. 4 : les trois plans du modèle ACE

On utilise trois constructeurs pour représenter la structure des classes ACE :

- **Type** : Permet le regroupement des composantes de différents types ;
- **Ensemble** : permet de regrouper des objets homogènes ayant le même type.

- **Liste** : permet d'avoir une structure regroupant des éléments homogènes.

Les liens de composition introduisent une sémantique de cardinalité entre un objet et ses composants. Ainsi, si l'attribut est :

- **Monovalué** , le lien est alors représenté par une flèche simple et traduit l'existence d'une dépendance fonctionnelle entre l'objet et son composant;

- **Multivalué** , le lien est représenté par une flèche double lorsque les valeurs sont structurées en un ensemble et une flèche triple lorsque elles sont organisées sous forme de liste. Ces deux derniers liens traduisent l'existence d'une dépendance multivaluée.

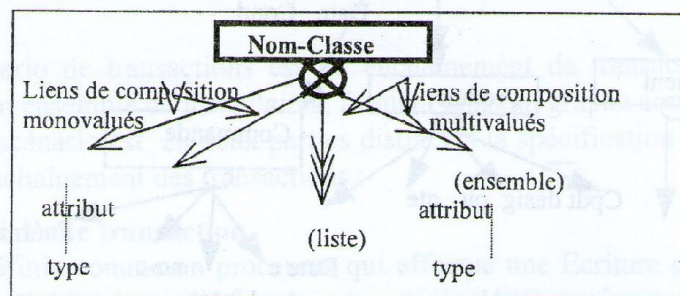


Fig. 5 : Représentation d'une classe

4.2. Simplification du contenu du Formulaire

Sur la structure arborescente du contenu du document, on applique un processus de simplification semi-automatisé qui donnera une nouvelle structure arborescente simplifiée du contenu du document en appliquant les règles :

- Tous les objets qui sont composés uniquement de textes ou de graphiques (ou des deux), sans aucun champ d'information sont enlevés de l'arborescence .
- Tous les champs de type calculé appartenant à des classes d'objets identifiés sont enlevés.

4.3 Identification des classes d'objets

Dans cette phase, nous modélisons la structure simplifiée dans le plan A du modèle ACE : Nous identifions les classes d'objets, leurs composants et leurs liens de composition. Pour cela , nous commençons par identifier les attributs déterminants (clés), les attributs spécifiques et les contraintes d'intégrités associées aux éléments du document (dépendances fonctionnelles, dépendance multivaluées, unicité, et non nullité). Pour plus de détail voir [Malki-97]. La modélisation de la structure arborescente du document Facture en un modèle orienté objet ACE est présentée dans la figure (Fig. 6).

5- La modélisation de la dynamique orientée scénarios de transactions

La modélisation de la dynamique orientée scénarios a montré ses insuffisances dans la gestion des opérations dans les scénarios (cas d'utilisation) [Alima,97]. La modélisation orientée transaction est, elle aussi, très complexe. Car, il faut modéliser le type de partage de données inter-transaction et le type de validation inter-transactions. Le couplage des deux modèles va régler la complexité existante dans chacune des deux approches car la gestion inter-opérations dans les cas d'utilisation manque de rigueur et ne possède aucun modèle de soutien pour minimiser les aléas de l'utilisation informelle des opérations. Par contre le modèle de transactions possède les outils nécessaires pour rendre formel le modèle des scénarios : Les diagrammes des scénarios de transactions offrent une vision globale du Système d'information car ils mettent en évidence les relations entre les scénarios de transactions. Cependant, le type de communication entre les scénarios est asynchrone, non déterministe et faiblement typé. De plus, un diagramme des scénarios de transactions DST spécifie un sous-ensemble de tous les enchaînements possibles entre les scénarios, mais en aucun cas tous les enchaînements et encore moins les enchaînements autorisés. L'absence de sémantique d'exécution est remplacée par le rajout d'événements [Menam,84] afin de modéliser des événements déclencheurs, ou encore le rajout d'un nouveau type de flux, le flux de contrôle, spécifiant l'enchaînement des scénarios OMT[Rumba,91], dans OOA avec les diagrammes ADFD [Coad,90], ou encore dans [Ward,85].

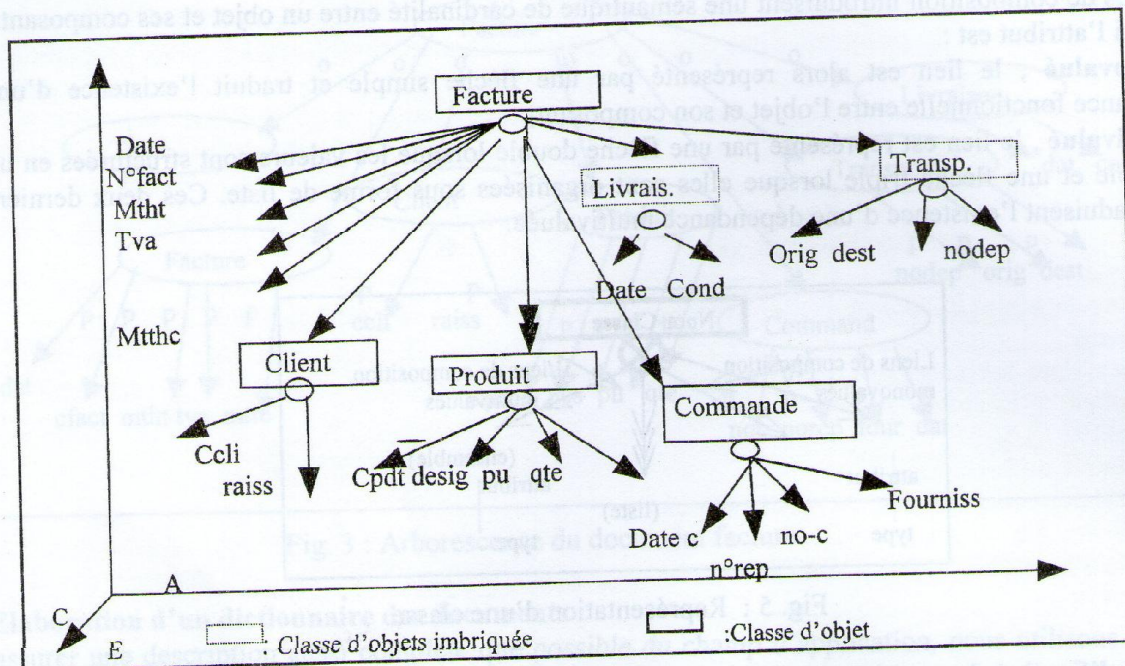


Fig. 6 : Schéma objet partiel.

5.1 - Le modèle de transaction

Une transaction est la plus petite unité de traitement pour une application utilisant une base de données. Le modèle de transaction plat est le modèle de base. Il impose un ensemble de propriétés assurant la cohérence des données. Dans ce modèle, une transaction doit vérifier les propriétés Atomicité, Cohérence, Isolation et Durabilité (ACID).

5.2 - Le couplage des scénarios de transactions à l'analyse des formulaires

Notre démarche distingue la spécification individuelle de chaque transaction, de la spécification de l'enchaînement de ces transactions. Cet enchaînement forme un graphe connexe dans chaque scénario. En considérant qu'un scénario de transactions est identifié par un événement déclencheur (conditions de son exécution). Cette approche nous donne, ainsi, une vision globale des différents scénarios de transactions du système d'information. Les liens entre les scénarios sont les messages d'événements de déclenchement, d'inhibition, d'exclusion, de dépendances... Le graphe de cet enchaînement est adapté des graphes de flots d'activités [Ward,85]. Tout en respectant la notion de trace d'événement dans la spécification inter-scénarios du système d'information. Comme le modèle transactionnel et le modèle des scénarios sont des modèles de synthèse qui appartiennent au même plan de développement extra-hiérarchique dans la décomposition du Système d'Information, la finalité de notre étude est d'avoir comme résultat un modèle objet externe dérivé du modèle inter-objet, du type modèle comportemental inter-objet de scénario [Booch,92], déduit lui-même du modèle d'enchaînement des scénarios de transactions IFO2[Teiss,94.] Ce modèle objet externe est, lui-même utilisé pour la validation et pour l'encapsulation des méthodes, engendrées par les transactions, dans le modèle objet de la base de données. Cette validation va se faire par une confrontation «visuelle graphique» des deux schémas (Fig. 12), en respectant la logique de la conception par utilisation en faisant participer l'utilisateur au développement.

Les dépendances de comportement ont un apport transactionnel, comme les dépendances inter-scénario, et un apport propre au scénario présenté, comme les chemins pour accéder à un objet donné (lien de référence, liens de causalité, ...).

Ainsi, après avoir capté tous les objets des différents scénarios, le graphe d'enchaînement des transactions représente le modèle comportemental dans le plan C du modèle ACE. Et il est stocké dans la base des scénarios de transaction décrite par le méta-modèle d'enchaînement des transactions. Le graphe comportemental (C) est du type trace d'événement et modélise l'ordre des envois d'événements supportés par les requêtes des transactions entre les différents objets du système d'information.

5.3 - Analyse descriptive des scénarios de transactions : (P21 : Fig. 1)

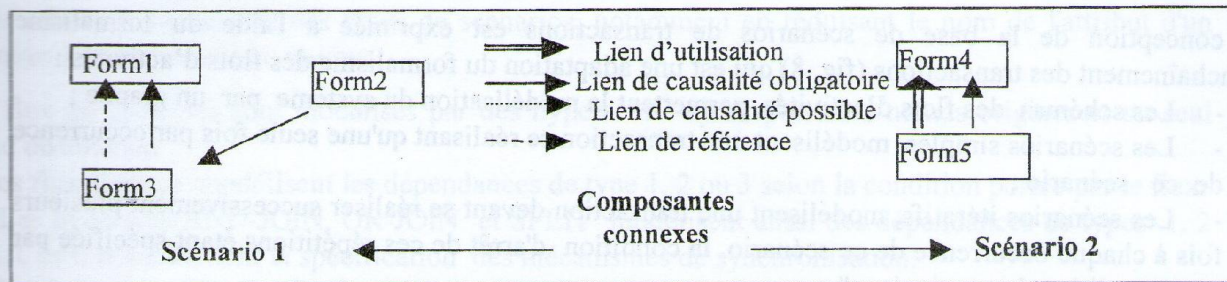


Fig. 7 : Graphe des scénarios

Définition : Un scénario de transactions est un enchaînement de transaction effectué par un ou plusieurs acteurs sur un ensemble de formulaires, formant ainsi un graphe connexe.

La spécification d'un scénario est en deux parties distinctes la spécification de chaque transaction et la spécification de l'enchaînement des transactions :

5.3.1 - la spécification d'une transaction

Une transaction est définie comme un processus qui effectue une Ecriture et (/ou) une Lecture qui nous permet de dériver l'objet du modèle de données manipulé (Connaissance issue des formulaires). Elle appartient au modèle externe et elle invoque des traitements dans le modèle objet «interne » de la base de données tels que : Règles de gestion, contraintes d'intégrité, calcul ou élaboration d'information dérivée, traitement et édition des erreurs ou plus généralement traitement et édition des exceptions.

5.3.2 - Conception de la base de données des scénarios de transactions.

La conception de la base des scénarios de transactions (Fig. 8) est la spécification de l'enchaînement des transactions. Elle doit définir : les conditions d'exécution de la transaction dans l'environnement particulier du système d'information et l'ordonnancement entre les transactions du système d'information.

- La première étape permet de spécialiser une transaction pour un système d'information donné en l'enrichissant des informations. Ces informations sont notamment les droits d'exécutions, les conditions initiales que doivent vérifier les données de la base, des conditions permettant de contrôler le déroulement correct de la transaction, et enfin le délai acceptable de son exécution. Cette démarche aboutit à l'encapsulation de la transaction avec ses nouvelles informations dans un objet-scénario. Au cours de cette démarche, la transaction n'est connue que par son interface.

- La seconde étape, Ces scénarios sont organisés entre-eux par la spécification des dépendances inter-scénarios. Nous regroupons ces dépendances en deux catégories :

- Les dépendances d'enchaînement direct, permettant la désignation d'un scénario à déclencher la fin d'un autre ;

- Les dépendances d'enchaînement indirect, liées à l'historique, spécifiant des liens entre des transactions non consécutives.

A partir de ces deux catégories nous déduisons les types de dépendances suivantes :

- 1) A la fin du scénario A, quelle que soit sa terminaison, il faut exécuter le scénario B
- 2) A la fin du scénario A, si celui-ci a réussi (état=committed), alors il faut exécuter le scénario B.
- 3) A la fin du scénario A, si celui-ci a échoué (état=Aborted ou fail), alors il faut exécuter le scénario B.
- 4) A la fin du scénario A, si celui-ci a échoué, alors il faut exécuter le scénario B comme scénario B pour compenser les effets de A. L'état du scénario A devenant alors COMPENSATED.
- 5) Le scénario A ne peut être exécuté si le scénario B n'a pas été exécuté auparavant .
- 6) Si le scénario A a fini son exécution dans l'état 2a alors le scénario B doit avoir fini dans l'état 2b.
- 7) Si le scénario A a fini son exécution dans l'état 2a alors la prochaine exécution du scénario B , et seulement la prochaine doit finir dans l'état 2b.
- 8) Si le scénario A échoue, alors l'ensemble du système considéré doit être stoppé et une erreur doit être indiquée.
- 9) Les scénarios A et B ne doivent pas être exécutés simultanément, l'un pouvant commencer son exécution que si l'autre a terminé son exécution.
- 10) Le scénario ne peut plus être exécuté durant tout le système si le scénario B a déjà été exécuté.

La conception de la base de scénarios de transactions est exprimée à l'aide du formalisme d'enchaînement des transactions (fig. 8) qui est une adaptation du formalisme des flots d'activités :

- Les schémas des flots d'activités, permettent la modélisation du système par un graphe ;
- Les scénarios simples, modélisent une transaction se réalisant qu'une seule fois par occurrence de ce scénario ;
- Les scénarios itératifs, modélisent une transaction devant se réaliser successivement plusieurs fois à chaque occurrence de ce scénario, la condition d'arrêt de ces répétitions étant spécifiée par une "condition de terminaison"
- Les scénarios parallèles, modélisent une transaction s'exécutant plusieurs fois en parallèle à chaque occurrence de ce scénario ;
- Les super-scénarios, permettant de décomposer un scénario par un autre graphe d'enchaînement.

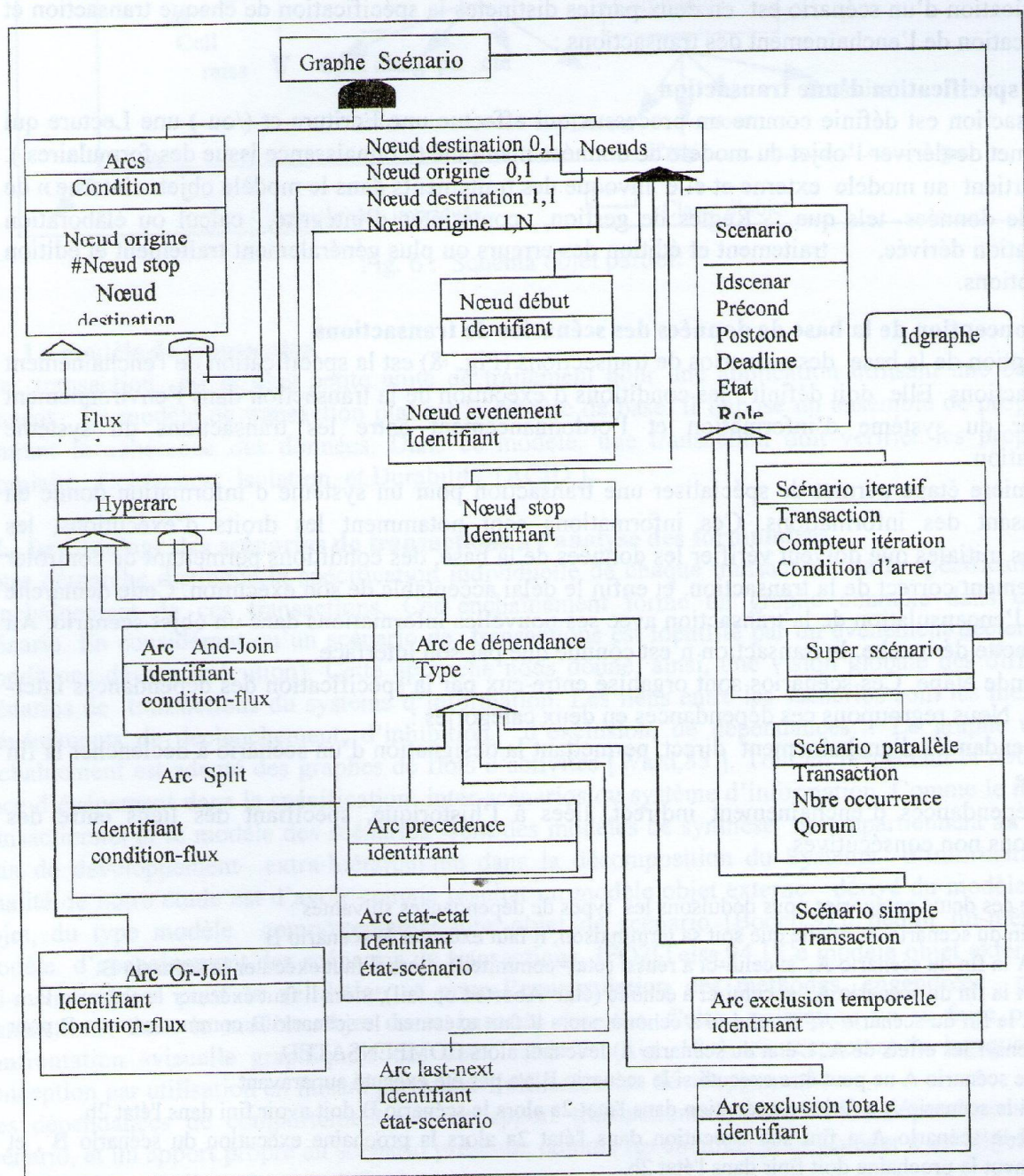


Fig. 8 : Meta-modèle de la base des scénarios de transactions

Les flux de contrôles modélisent l'ordonnancement des scénarios selon une condition de transition portée par chaque flux.

Nous adaptons l'ensemble des types de scénarios, notamment en réduisant le nom de l'attribut d'un scénario à une transaction unique.

Les flux de contrôles sont modélisés par des hyperarcs ayant plusieurs nœuds origines et un seul nœud destination :

- les flux simples modélisent les dépendances de type 1, 2 ou 3 selon la condition portée par le flux.
- Les hyperarcs AND-JOIN, OR-JOIN et SPLIT modélisent aussi des dépendances de types 1, 2 ou 3. Ces flux permettent la spécification des mécanismes de synchronisation.
- Les hyperarcs ETAT-ETAT modélisent des dépendances de type 6. Ces arcs permettent aussi une combinaison logique de type AND ou OR entre différents états de scénarios origines de l'arc ;
- Les hyperarcs Last-Nest permettent la modélisation de dépendances de type 7 ;
- Les hyperarcs précedence permettent la modélisation de dépendances de type 1 ;
- Les hyperarcs Exclusion-Totale permettent la modélisation de dépendance de type 10 ;
- Les hyperarcs Exclusion-TEMP modélisent les dépendances de type 9.

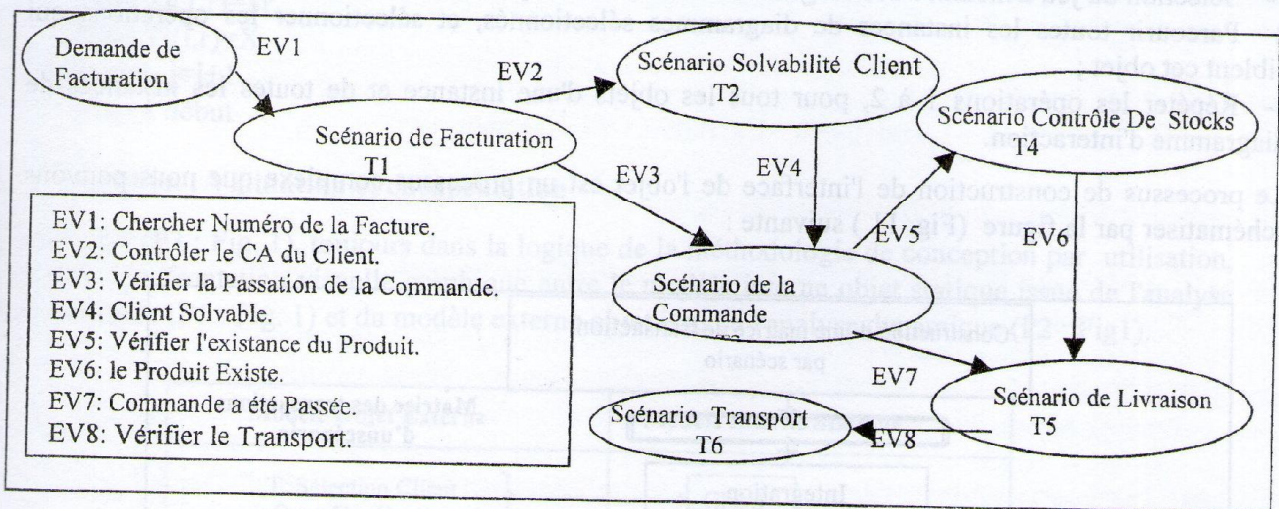


Fig. 9 : Modèle du Scénario de Facturation

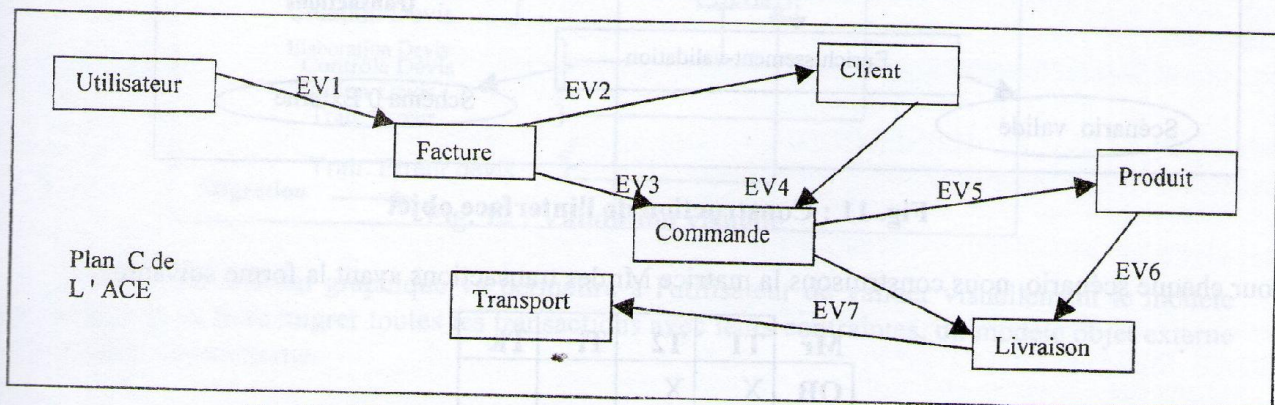


Fig. 10 : Diagramme d'Objets du Scénario de Facturation

Le méta-modèle de l'enchaînement des transactions (Fig. 8) nous permet d'avoir une structure d'objet [Bouze,92] pour les scénarios afin de concevoir la base des scénarios. A partir de laquelle nous allons extraire le modèle d'enchaînement inter-scénario (Fig. 9) et déduire le modèle comportemental inter-objet (Fig. 10). Donc, après avoir conçu cette base des scénarios de transactions. Nous allons traiter ces scénarios (P22 : Fig. 1) par un processus dynamique [Alima,97], pour permettre l'encapsulation des transactions et de construire le schéma objet externe.

5.3.3 - Le processus de modélisation dynamique : (P22 : Fig. 1)

Le processus de l'aspect dynamique comporte les étapes suivantes :

- 1- Extraire un scénario de la B-Scénario transaction (Fig. 8) en structure tabulaire, afin de lister tous les formulaires ciblés dans la spécification d'une transaction ;

- 2- Analyser le schéma statique objet et les formulaires puis sélectionner les objets dérivés de cet ensemble de formulaires, grâce aux références qui ont été conservées ;
- 3- Présenter à l'utilisateur la structure prédéfini du diagramme de l'interaction ;
- 4- Construction de l'instance du diagramme ;
- 5- Les opérations 1 à 4 sont répétées jusqu'à épuisement des tables composant le scénario-transaction.

Remarque:

- Le scénario ici peut-être constitué d'une ou plusieurs transactions ;
- le passage de la construction du schéma objet-externe est évident pour confronter ce schéma externe avec le schéma objet statique et rajouter tous les traitements, contraintes, ... invoquées par la transaction
- le processus d'identification des transactions qui est la séquence des étapes suivantes :
 - 1- Sélection du jeu d'instance des diagrammes d'interaction qui nous intéresse et d'un objet;
 - 2- Parcourir toutes les instances de diagrammes sélectionnés, et sélectionner les opérations qui ciblent cet objet ;
 - 3- Répéter les opérations 1 à 2, pour tous les objets d'une instance et de toutes les instances de diagramme d'interaction.

Le processus de construction de l'interface de l'objet est un processus complexe que nous pouvons schématiser par la figure (Fig. 11) suivante :

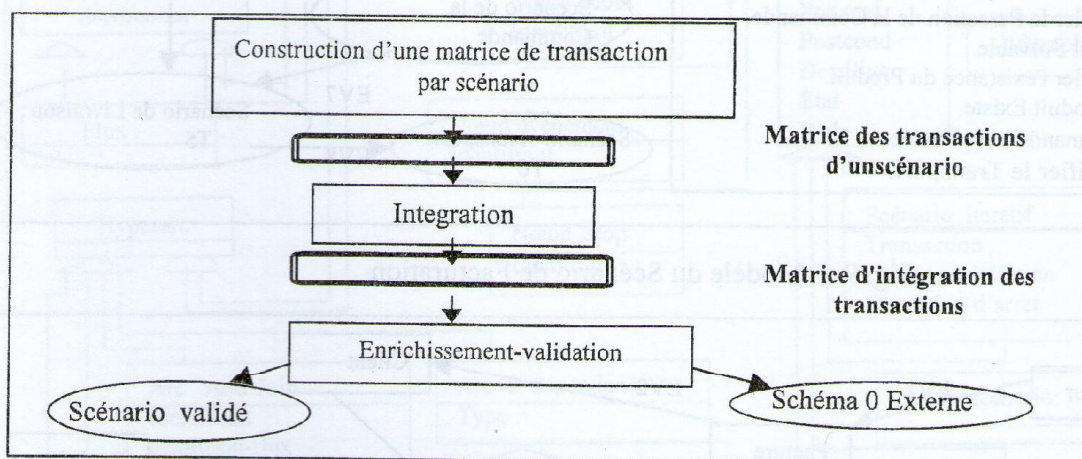


Fig. 11 : Construction de l'interface objet

Pour chaque scénario, nous construisons la matrice Mr des transactions ayant la forme suivante :

Mr	$T1$	$T2$	Ti	Tk
OB_1	X	X		
Ob_i			X	
Ob_n	X		X	X

Le résultat de la première phase de ce processus est un ensemble de matrices $Mr(i,j)$ dont le nombre est égal à un des scénarios.

$Mr(i,j)$:
 r varie de 1 à N , N le nombre de scénario
 i varie de 1 à M , M étant le nombre d'objet
 j varie de 1 à K , K étant le nombre de transaction

Puis on se focalise sur une opération et on construit de manière algorithmique la matrice d'intégration $Tj(i,r)$ suivante :

Tj	S1	Si	Sk
OB 1	X		
Obi		X	
Obn	X		X

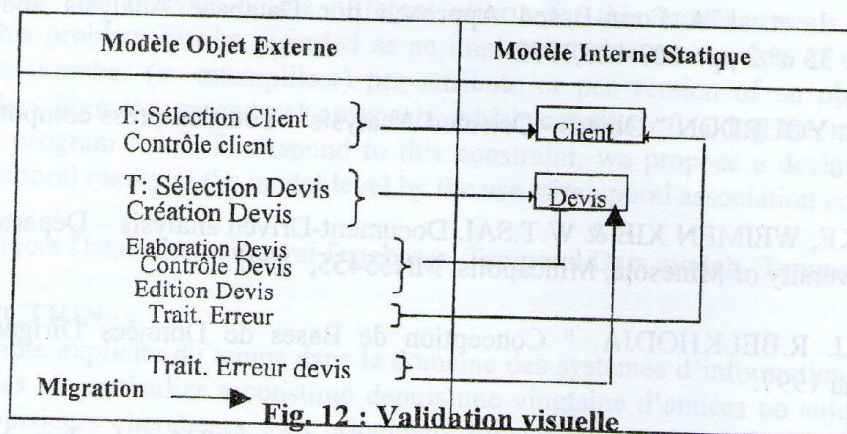
Il y a autant de matrices d'intégration que de transactions identifiées lors de l'étape précédente.

L'algorithme d'intégration est le suivant :

Initialisation: $r = 1, j = 1$
 Début : Lire $Mr(1,j)$
 Sélectionner $Mr(i,1)$ où $Mr(i,j) = 'X'$
 Faire $Tj(i,1) = Mr(i,1)$
 Faire $Tj(1,r) = Sr$
 Faire $Tj(i,r) = X$
 $R = r + 1, j = j + 1$
 Aller à début

6 - Intégration : Validation et encapsulation

Ce processus (P3 : Fig. 1), toujours dans la logique de la méthodologie de conception par utilisation, va être une confrontation visuelle graphique entre le modèle interne objet statique issue de l'analyse des formulaires (P1 : Fig. 1) et du modèle externe objet issu de l'analyse dynamique (P2 : Fig1).



Cette confrontation graphique va permettre à l'utilisateur de valider visuellement le modèle objet statique et va faire migrer toutes les transactions avec leurs contraintes, du modèle objet externe vers le modèle objet interne.

7- Conclusion :

Cette étude rentre dans le cadre de la méthodologie de conception basée sur l'analyse des formulaires et des scénarios de leurs transactions. Elle rentre dans le cadre de la conception par utilisation et peut constituer un outil dans un Atelier de Génie Logiciel.

Les perspectives vont permettre d'introduire le concept de l'hypertexte dans l'analyse des formulaires en hypertextualisant l'ensemble des formulaires et en conceptualisant la structure hypertexte pour obtenir le schéma orienté objet.

Références

- [Alima-97] Z. Alimazighi et C. Rolland. Génération d'Objets par une Analyse couplant les Formulaires et les Scénarios. In Proc. Of INFORSID'97
- [Ayach,96] Ayache M. « Méthodologie CASSiopéE : intégration et évaluation » Ingénierie des sys. d'Information vol. 4 n° 2 pp. 137-160, 1996
- [Batin,84] Batini C .et al. " A Methodology for Conceptual Schema Design of Office Database"Info. Syst. , vol. 9 n°. 3/4 pp. 251-263, 1984
- [Belkh,99] R.Belkhodja et M.Malki « Approche basée sur l'analyse des formulaires et des scénarios de Transactions » LRISI 99/01 Université Djilali Lyabes
- [Booch,92] G. Booch . "Conception Orientées Objets et Applications" Addison - Wesley 1992
- [Bouze,92] Mokrane BOUZEGHOUB, Georges GARDARIN, Patrick VALDURIEZ " Du C++ à Merise Objets - Objets " Editions Eyrolles 1992
- [Choob,92]Choobineh J. et al."A Form-Based Approach dor Database Analysis and Design" Comm. of the ACM vol. 35 n° 2 pp. 107-120,1992
- [Coad,90] P. COAD, E. YOURDON " Objecs - Oriented Analysis" Yourdon press computing series - Prentice - Hall 1990
- [Drake,92] J.M. DRAKE, WEIMEN XIE & W.T.SAI. Document-Driven analysis – Département of Computer Science, University of Minesota, Mineapolis, MN55455, 1992.
- [Malki,97] M.MALKI. R.BELKHODJA. " Conception de Bases de Données Dirigée par des Formulaires ", CNBatna 1997.
- [Menam,84] S. McMENAMIN, J.PALMER. " Essentiel Systems Analysis" Yourdon Inc , 1984
- [Rumba,91] J. RUMBAUGH, M.BLAHA, W.PREMERLANI, F.EDDY, W.LORENSEN " Object-Oriented Modeling and Design" Prentice - Hall International Editions , 1991.
- [Teiss,94] M. TEISSEIR. « Le modèle IFO2 : de la modélisation compotementale à la dérivation » Thèse de Doctorat en Science de l'Université d' Aix Marseille II, 1994
- [Ward,85] PAUL WARD, STEPHEN MELLOR. " Structured Development for real-time." Volume 1-3, Yourdon press : Englewood Cliffs, NJ 1985.
- [Yourd,89] E. YOURDON " Modern structured Analysis" Prentice - Hall International Editions , 1989.