

Génération d'objets : une approche basée sur l'analyse de documents et scénarios

Zaia Alimazighi

Laboratoire des Systèmes Informatiques (LSI)
Département d'informatique, Faculté de Génie électrique
et Informatique, USTHB, BP 32 El Alia, Alger
Email: alimazighi@wissal.dz

1. Introduction

Une gestion rationnelle d'une entreprise passe par la maîtrise, par les décideurs, des informations traitées, elle nécessite donc la construction de systèmes d'information (SI), représentant tous les aspects de la vie d'une organisation. Le cycle de développement de SI comporte deux grandes phases [DUB 89], [OLL 89] :

- L'ingénierie des besoins qui aboutit à un schéma conceptuel de SI, représentation abstraite de la réalité.
- L'ingénierie des systèmes dont le résultat est la solution implémentable du schéma conceptuel.

La réalisation de la première phase est celle qui pose un certain nombre de problèmes, car le processus d'abstraction fait beaucoup appel à l'interprétation par le concepteur, des connaissances du monde réel. Beaucoup d'outils ont été développés afin de faciliter l'accomplissement de cette phase. C'est ainsi que l'utilisation des techniques de l'intelligence artificielle [ALI 95], dans la compréhension du langage naturel, avait permis le développement d'outils favorisant la génération de schémas conceptuels à partir d'énoncés. Nous citons ALECSI [ROL 86], développé sur la méthode REMORA, ou SECSI [BOU 85] permettant la génération de schéma conceptuel de base de données relationnelles. Tous ces outils développés sur la base de méthodologies bien définies, sont restés utilisables par une classe restreinte de spécialistes.

Il existe aujourd'hui, sur le marché, un grand nombre de méthodologies, et, contrairement à ce que l'on peut penser, cette prolifération n'a pas pour autant réglé la délicate phase d'acquisition des besoins des utilisateurs. La multiplicité des concepts sous-jacents à ces méthodologies n'a fait qu'accroître l'embarras des ingénieurs d'application pour la résolution de cette phase. Ce foisonnement fait que les chercheurs se trouvent placés devant trois alternatives :

- Développer une méthode universelle résultat d'une intégration de plusieurs méthodologies; c'est la démarche adoptée par Rumbaugh, Booch, et Jacobson, en vue de proposer une méthode unifiée [RUM 96], à partir d'OMT [RUM 95], de OOD [BOO 91], et de OOSE [JAC 92] et qui a aboutit à la notation unifiée UML [MUL 01].
- Développer une ingénierie des méthodes qui permettrait à l'ingénieur d'application de construire " sa méthode " à partir de fragments de méthodes connues. Cette approche sous-entend la nécessité, d'un méta-modèle de processus et, de construire une bibliothèque de fragments dans un cadre de réutilisation. C'est l'approche adoptée par l'équipe du CRI dans le cadre du projet européen Nature [ROL 93].
- Développer des méthodologies par type d'application. Cela sous-entend qu'il faille appliquer, en chaque lieu, la méthode la plus adaptée au projet. On les qualifie de méthodes situationnelles [ROL 97], [BOU 97] [BEN 01].

Les deux dernières alternatives ne sont pas exclusives, dans la mesure où une démarche situationnelle (troisième alternative) peut être développée à partir d'un modèle de processus (deuxième alternative).

En nous plaçant dans la troisième alternative, nous nous sommes intéressés, dans l'ingénierie des besoins, à définir un processus original d'extraction de l'information qui impliquerait le plus tôt possible l'utilisateur dans la phase de développement de SI afin de bénéficier au maximum de ses connaissances. Dans ce cadre, nous nous sommes intéressés aux processus basés sur les scénarios (SEP, Scenario-based Engineering Process) qui définissent des méthodologies centrées-utilisateurs pour l'acquisition des besoins. Les informations contenues dans une description de scénario sont diverses, hétéroclites, mais riches. Elles peuvent concerner tant les différents agents d'une organisation, que les phénomènes du monde réel en termes d'événements et d'actions. De tout temps les gestionnaires des entreprises élaboraient des scénarios quand il fallait projeter des objectifs, à court, moyen ou long terme. Des méthodologies de conception de systèmes d'information appartenant à la génération passée (approche systémique) les utilisaient de manière interprétative dans la phase d'analyse, afin de concevoir les modèles de traitements (par exemple la méthode Merise). Aujourd'hui un intérêt nouveau lui a été porté par les développeurs, tant dans le domaine général de l'ingénierie des besoins [MCG 97], [CAR 92], [GOU 95], [POT 94], [CBA 99] etc. que celui particulier des nouvelles méthodologies de conception de systèmes (approche orientée objet) [JAC 92], [RUM 94], etc. Par ailleurs, nous nous plaçons dans le domaine particulier des applications de gestion, et, lors de notre analyse des entreprises, nous avons constaté que les véhicules incontournables de l'information, dans la plupart des cas,

étaient les documents et en particulier, les documents. Historiquement, ils ont été le point de départ de l'analyse fonctionnelle des premières méthodologies de conception de SI. L'analyse de leur contenu et de leur circulation permettait de décrire les données et procédures de traitement ; ainsi une méthodologie comme Merise [TAR 91] les a largement utilisés dans les diagrammes de flux et diagrammes de circulation et traitement des informations (DCTI). Plus tard, des outils sophistiqués ont été conçus ou développés, sur l'analyse de documents, afin d'en extraire des schémas E/R [CHO 88], [TAL 90], [BAT 88], [BAN 92]. Tous ces travaux ne se sont intéressés qu'à l'aspect statique de l'information. Or, les récents développements méthodologiques ont favorisé l'émergence de modèles permettant de mieux cerner la complexité des SI en spécifiant tant l'aspect statique que dynamique. Ce sont les méthodologies orientées objets.

Notre approche combine l'analyse de scénarios avec celle des documents de gestion afin de modéliser complètement une application de gestion.

Ainsi, nous proposons un processus qui, à partir d'un ensemble de scénarios bien définis, permet :

- l'enrichissement d'une description statique de données dérivée d'une analyse de documents,
- d'adjoindre une description dynamique en se basant sur une analyse de scénarios,
- de valider le modèle de données et les scénarios.

Une présentation des fondements de notre approche se trouve en section 2, suivie d'une description globale du processus, composé de deux étapes, abstraction et intégration. Une analyse détaillée du processus d'abstraction est développée en section cinq.

2.Fondements de la démarche proposée

2.1. Les scénarios dans les méthodes d'analyse orientée objet

Cette dernière décennie a vu une prolifération de méthodes d'analyse orientée objet. Nous nous sommes intéressés à celles qui ont utilisé le concept de scénario dans la phase d'ingénierie des besoins.

Nous avons pris un échantillon de quatre approches qui nous ont semblé les plus intéressantes à analyser : OBA (Object Behavior Analysis) de Rubin, OBJECTORY de Jacobson, OMT de Rumbaugh et l'approche de Lubars et al. Notre objectif est d'analyser la capacité des méthodes à définir un modèle objet à partir de l'élaboration de scénarios.

2.1.1.OBA (Object Behavior Analysis) [RUB92]

Rubin [RUB92] définit une approche en quatre étapes :

- **Etape 0** : Cerner le contexte de l'analyse ; dans cette phase l'analyste se préoccupe d'identifier les objectifs et les ressources nécessaires à l'analyse
- **Etape1** : Compréhension du problème ; lors de cette phase les scénarios sont élaborés en utilisant une représentation tabulaire
- **Etape 2** : Définition des objets ; lors de cette phase les objets sont identifiés, ainsi que leurs attributs.
- **Etape 3** : Classification des objets et identification de leurs relations ; les objets sont structurés en hiérarchie.
- **Etape 4** : Modélisation de la dynamique du système ; les différents états de chaque objet sont répertoriés, les événements et opérations sont définis.

La représentation de Rubin est une table à quatre colonnes qui décrit les interactions entre l'initiateur de l'action, et le participant avec son service.

L'identification des objets s'effectue à partir de l'analyse des participants des scripts. Toute partie qui fournit un service est un objet potentiel. Il classe les participants en trois catégories :

- Les participants qui ne sont pas des initiateurs et qui ne peuvent pas être des objets.
- Les participants qui font référence à des objets nommés du monde réel et qui donnent naissance à un ensemble d'objets.
- Les participants qui sont aussi des initiateurs et qui peuvent devenir des objets à condition qu'ils ne soient pas surchargés, c'est-à-dire remplissant plusieurs rôles.

La classification des objets et l'identification de leurs liens sont effectués à travers l'analyse des interactions entre les différents participants. Rubin utilise les techniques d'abstraction, de spécialisation, de factorisation pour construire le schéma des objets.

Dans l'étape 4, les différents états d'un objet sont identifiés à partir des pré et post conditions des scripts. Une recherche des clauses faisant référence aux objets identifiés, permet de lui associer des états. Lorsque toutes les clauses ont été parcourues, alors l'ensemble des états d'un objet peut être identifié. Un glossaire est alors élaboré. L'ensemble des états d'un objet permet d'identifier son cycle de vie. Rubin associe un événement à l'invocation d'un script, qui à travers les pré et post conditions induit les changements d'états des objets élaborés à partir du scénario. Les actions définies dans les scripts sont par défaut interprétées dans un ordre séquentiel. L'auteur prévoit d'étendre les scripts avec des diagrammes d'enchaînement des activités.

2.1.2.OBJECTORY [JAC92], [JAC94]

Dans OBJECTORY, la phase d'analyse permet d'élaborer le modèle objet à partir des cas d'utilisation. Jacobson suggère de d'identifier pour chaque cas d'utilisation les objets dont il a besoin pour s'exécuter. Nous rappelons que trois types d'objets sont définis, objet interface, objet contrôle, et objet entité. Il ne propose pas de démarche précise pour leur construction, mais seulement des principes généraux basés sur les types de comportement pris en compte par le cas d'utilisation.

- Les activités peuvent décrire un comportement directement relié aux tâches invoquant un contact avec l'acteur. L'objet interface est responsable de ce type de comportement.
- Les activités internes bien définies qui n'invoquent pas de liens avec l'acteur peuvent définir un comportement devant être pris en compte dans un objet contrôle.
- Les occurrences d'entités manipulées par le cas d'utilisation telles que des produits, des documents, des contrats, etc. deviennent des objets entités.

Lorsque la catégorisation des objets est réalisée à partir d'une analyse informelle des cas d'utilisation, une description leur est associée en se basant sur le principe que chaque objet qui participe à un cas d'utilisation a un rôle précis en termes d'opérations d'événement, et d'attributs.

Plusieurs types d'associations entre les objets sont déterminés : connaître «acquaintance», constituer «consistsOf», hérite «inheritance». «l'acquaintance» signifie qu'un objet connaît l'existence d'un autre objet. L'association «consistsOf» signifie qu'un objet est constitué de plusieurs autres objets. L'association «inheritance» signifie que des propriétés peuvent être réutilisées. Aucune technique n'est précisée sur la manière de construire ses associations. Les rôles des objets sont représentés à l'aide du diagramme d'interaction.

2.1.3.OMT [RUM94]

Originellement, cette méthode ne comportait pas d'analyse de scénarios. Le modèle objet était élaboré à partir d'une analyse informelle des besoins des utilisateurs (interviews, documents etc.). Un dictionnaire des informations permettait de déterminer les propriétés et liens des objets. Le modèle dynamique OMT était construit à partir du diagramme des états des objets définis dans le modèle objet. Avec l'influence de OBJECTORY, Rumbaugh a préconisé l'utilisation des scénarios pour compléter le modèle dynamique. Il propose une démarche de construction des scénarios comportant les phases suivantes :

2.1.2.OBJECTORY [JAC92], [JAC94]

Dans OBJECTORY, la phase d'analyse permet d'élaborer le modèle objet à partir des cas d'utilisation. Jacobson suggère de d'identifier pour chaque cas d'utilisation les objets dont il a besoin pour s'exécuter. Nous rappelons que trois types d'objets sont définis, objet interface, objet contrôle, et objet entité. Il ne propose pas de démarche précise pour leur construction, mais seulement des principes généraux basés sur les types de comportement pris en compte par le cas d'utilisation.

- Les activités peuvent décrire un comportement directement relié aux tâches invoquant un contact avec l'acteur. L'objet interface est responsable de ce type de comportement.
- Les activités internes bien définies qui n'invoquent pas de liens avec l'acteur peuvent définir un comportement devant être pris en compte dans un objet contrôle.
- Les occurrences d'entités manipulées par le cas d'utilisation telles que des produits, des documents, des contrats, etc. deviennent des objets entités.

Lorsque la catégorisation des objets est réalisée à partir d'une analyse informelle des cas d'utilisation, une description leur est associée en se basant sur le principe que chaque objet qui participe à un cas d'utilisation a un rôle précis en termes d'opérations d'événement, et d'attributs.

Plusieurs types d'associations entre les objets sont déterminés : connaître «acquaintance», constituer «consistsOf» , hérite «inheritance». «l'acquaintance» signifie qu'un objet connaît l'existence d'un autre objet. L'association «consistsOf» signifie qu'un objet est constitué de plusieurs autres objets. L'association «inheritance» signifie que des propriétés peuvent être réutilisées. Aucune technique n'est précisée sur la manière de construire ses associations. Les rôles des objets sont représentés à l'aide du diagramme d'interaction.

2.1.3.OMT [RUM94]

Originellement, cette méthode ne comportait pas d'analyse de scénarios. Le modèle objet était élaboré à partir d'une analyse informelle des besoins des utilisateurs (interviews, documents etc.). Un dictionnaire des informations permettait de déterminer les propriétés et liens des objets. Le modèle dynamique OMT était construit à partir du diagramme des états des objets définis dans le modèle objet. Avec l'influence de OBJECTORY, Rumbaugh a préconisé l'utilisation des scénarios pour compléter le modèle dynamique. Il propose une démarche de construction des scénarios comportant les phases suivantes :

- Identification des limites de l'application, et des objets de son environnement (objets externes).
- Classification des objets externes par les rôles qu'ils jouent dans l'application. Chaque rôle définit un acteur.
- Pour chaque acteur, déterminer ses différentes manières d'utiliser le système. Chaque manière définira un cas d'utilisation. Un cas d'utilisation peut se décomposer en plusieurs scénarios.
- Pour chaque cas d'utilisation, identifier l'événement associé à un acteur et qui initie un cas d'utilisation.
- Ecrire une description en langage naturel du cas d'utilisation.
- Définir les cas d'exceptions.
- Répertorier les fragments communs à plusieurs scénarios afin de construire les scénarios de base et ceux additionnels (association "add").

2.1.4. L'approche de Lubars et al. [LUB93]

L'approche décrite dans [LUB93], est composée d'une intégration de deux techniques :

- Une approche ascendante qui construit des fragments de modèle orienté-objet correspondant à des segments dans les documents décrivant les besoins des utilisateurs.
- Une approche descendante conduite par les scénarios.

2.1.4.1. L'approche ascendante

Le document est constitué de deux types de paragraphes : Ceux contenant les informations sur les objets et leur attributs et qui permet de construire le modèle objet et ceux contenant les informations sur leur comportement. Le processus démarre par une phase de classification et numérotation des phrases. A chaque phrase descriptive d'objets, un fragment de modèle est associé. A chaque phrase descriptive de comportement, un fragment de diagramme d'états est associé. Les fragments construits contiennent les références aux phrases dont ils dérivent.

2.1.4.2. L'approche descendante

Dans cette approche, les scénarios sont utilisés. Les auteurs de cette méthodologie font référence aux méthodes OMT [RUM91] et OOA [SHL92] qui utilisent toutes deux un modèle événement. Le scénario est un ensemble de phrases en langage naturel retraçant les événements qui surviennent dans une situation donnée.

Ces deux approches sont combinées pour construire le modèle objet. En effet, l'approche ascendante permet de définir des fragments de modèles, sans relation apparente entre eux ; les scénarios fournissent l'information permettant de les relier. En fait, dans cette approche, le passage aux objets (structure et comportement) reste intuitif et interprétatif.

2.2. Les systèmes de Workflow [KHO 98], [SAA 96]

Les systèmes de Workflow automatisent la gestion des flux de tâches dans une organisation. Nous reprenons à cet effet la définition de Naffah [NAF 94] : un Workflow est " un travail coopératif impliquant un nombre limité de personnes devant accomplir, en un temps limité, des tâches articulées autour d'une procédure définie et ayant un objectif global ". Les origines du workflow remontent à la gestion électronique de documents (GED) [KHO 92]. La GED consiste à traiter et à organiser la documentation de manière à la retrouver rapidement. Initialement elle ne prenait en compte que l'aspect statique de la vie d'un document. Mais un document n'est pas une entité qui est simplement traité puis classé. Il circule entre les différents agents d'une organisation, où il peut être lu, modifié, classé. Ainsi, le Workflow devient le complément naturel de la GED en prenant en compte l'aspect dynamique de la vie d'un document.

2.2.1. Les documents

Une application de gestion peut faire appel à de nombreux documents. La structure d'un document peut être complexe, et il peut contenir tant du texte, des données structurées, que de l'image. Il exprime surtout de la connaissance statique (par exemple, nom d'un client, référence de produit,...) et parfois de la connaissance dynamique (description de procédés opérationnels, règles de gestion dynamiques,...).

Un document, en plus d'avoir un contenu, circule à travers les différents postes de travail d'une organisation. Lorsqu'il circule, un document peut en induire d'autres, et nous pouvons ainsi définir des ensembles de documents inter-reliés. Deux types de liens existent entre des documents, des liens sémantiques (de composition, de référence...), et des liens d'induction (un document implique dans le temps, la génération d'un ou plusieurs autres documents). L'ensemble des documents, reliés par ces deux types de liens, constitue le réseau de documents. Une application peut faire intervenir plusieurs réseaux.

Le réseau de documents sous-entend un ensemble d'activités de traitement qui représente un ou plusieurs scénarios d'utilisation du réseau par les utilisateurs. Ainsi, une analyse de documents peut permettre d'identifier des processus de travail.

2.3 Les modèles Objets

Beaucoup de travaux existent aujourd'hui sur l'approche orientée-objet qui tient une place importante tant dans les domaines des langages de programmation et systèmes d'exploitation que celui des SI et génie logiciel [BOO 91], [RUM 95], [SCH 88], [SPA 89]. Cependant même s'ils ne proposent pas tous les mêmes concepts, ils incluent généralement les principes d'objet ayant des caractéristiques structurelles et comportementales, de classe, de liens structurels entre les objets et classes, d'héritage, de polymorphisme et d'encapsulation. Nous retiendrons les concepts suivants pour notre méthodologie.

- Deux concepts de base : *objet* et *classe*,
- Trois liens structurels entre les classes : liens de *composition*, de *référence*, d'*héritage*,
- Deux concepts représentant la dynamique du SI : l'*événement* et l'*opération*.

Ces deux derniers concepts sont dérivés de la méthode REMORA [ROL 87]. Nous rappelons brièvement que dans cette approche la propagation des changements d'états du SI à la suite d'exécution d'opérations était initiée par les événements. Le principe de base qui régit cette méthodologie est celui de causalité, qui exprime que des changements d'états particuliers d'objets constituent des événements qui induisent des opérations qui, elles-mêmes, impliquent de nouveaux changements d'états d'objets, et ainsi de suite...

3. Modélisation conceptuelle

L'approche méthodologique que nous avons développée [ALIM 99], basée sur les scénarios, les documents, et les Workflows est une démarche centrée sur l'utilisateur placée dans l'environnement suivant :

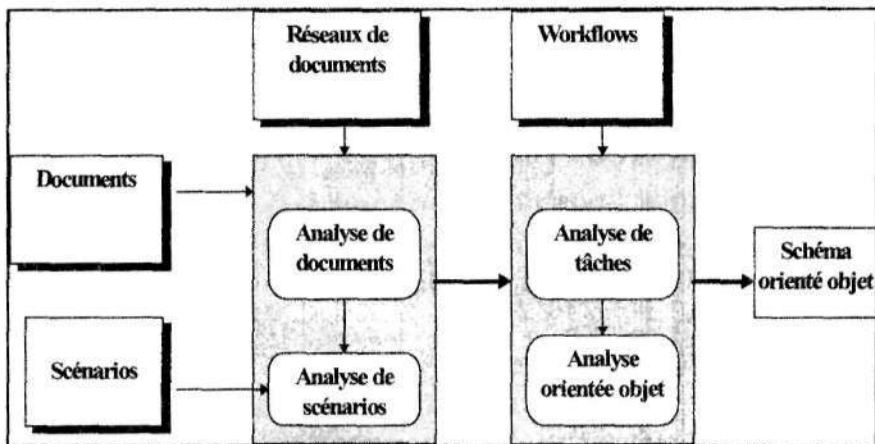


Figure 1: Architecture générale du processus

Les phases d'analyse de documents et d'analyse de scénarios ont pour produits résultats une base de données de documents et une base de données de scénarios et nécessitent la définition de trois modèles basés sur le modèle de base qui est l'orienté objet :

- Un modèle abstrait de Documents et,
- Un modèle de réseau de Documents, qui permettent l'élaboration de la base de données de documents.
- Un modèle de Scénario qui permet la construction de la base de scénarios.

La phase d'analyse de tâches nécessite un couplage entre le système de workflow déduit à partir du réseau de documents, et la base de scénarios.

3.1.1. Le modèle abstrait de Documents

Le modèle abstrait de document a été défini sur la base d'un découpage spatial et logique. Il devient une collection d'unités topologiques (analyse de l'espace) et d'unités logiques (analyse de ses éléments) (cf. [ALIM 96a]).

Définition : Le Modèle Abstrait de Documents (cf. figure 2)

Un document est un ensemble d'unités topologiques, organisées en arbre, vérifiant les propriétés suivantes :

1. Chaque nœud de l'arbre est structuré par un lien de composition simple ou multiple d'unités logiques.
2. Tous les nœuds non terminaux définissent un type de document.
3. Les nœuds terminaux ou feuilles sont les valeurs de ces unités, et représentant le contenu du document, soit une instance.
4. Chaque nœud de l'arbre est représenté par un objet.

Nous en rappelons la définition d'un lien de composition :

Un lien de composition entre deux classes définit une relation de dépendance structurelle forte entre chaque objet de la première classe, dit objet composite, et un ou plusieurs objets de la deuxième classe, dits objets composants. Un lien de composition peut être simple ou multiple. Un lien de composition simple indique qu'un objet composite est composé d'un et d'un seul objet composant, alors qu'un lien de composition multiple indique qu'un objet composite peut être composé de 1 à N objets composants [LEE 94].

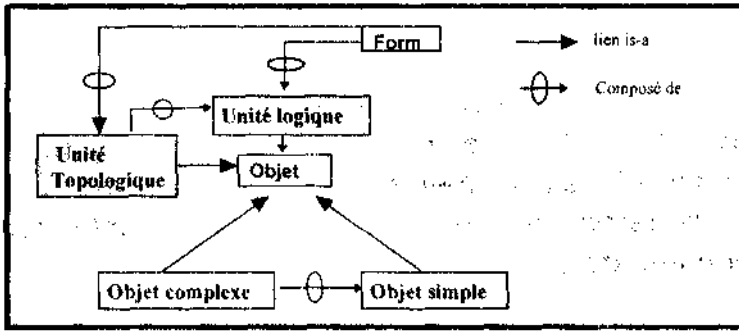


Figure 2. Le modèle abstrait de Documents

Nous avons défini en annexe, un exemple d'application qui est le scénario de gestion manuelle de redistribution des éléments d'un entrepôt. Nous donnons ci-après (cf figure 3) un exemple de description du formulaire F_2 qui décrit la localisation des éléments de plusieurs entrepôts.

Entreprise de distribution d'Alger

ETAT DE LOCALISATION DES ELEMENTS

Nom Entrepôt: Hamma Date: 20/09/2002
 Adresse: Alger

Nom élément	Endroit-Entrepôt
Vis 12mm	PA12
Baril essence	PA15
Ordinateurs	PD32
Bananes	PC12

Adresse- Entrepôt Cible
Hydra
Ain-Bénian
El harrach
Belcourt
Les halles
Mohammedia

Figure 3. Description du formulaire F_2

L'analyse de ce document permet de dégager quatre unités topologiques (cf figure 4).

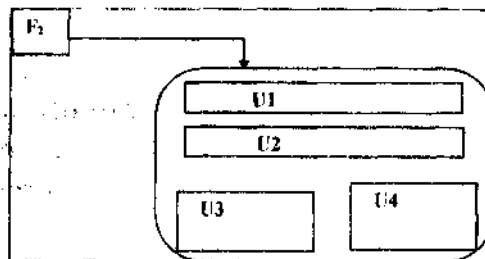


Figure 4. Schéma de la structure topologique

Cette structure topologique sera la même pour toutes les instances du formulaire " Etat de localisation des Eléments ".

Remarque

La définition des unités topologiques s'effectue pour l'instant d'une manière informelle. Il est fait appel à la vision spatiale qu'a l'utilisateur de son formulaire. Cependant, dans une étape ultérieure, rien n'exclut la possibilité d'utiliser une interface graphique pour élaborer la structure topologique du document.

L'arborescence associée à ce document est la suivante :

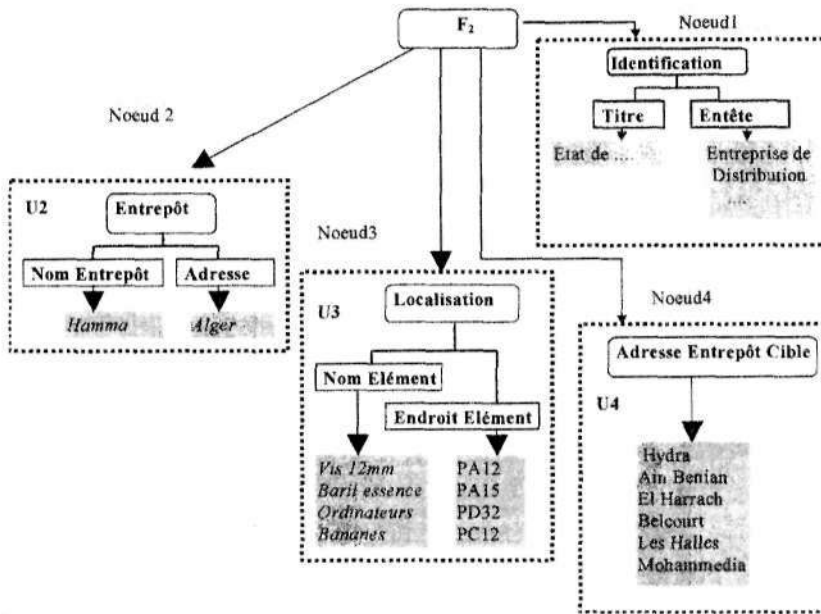


Figure 5. Schéma de la structure logique

Les noeuds 1, 2, 3, 4 représentent les quatre unités topologiques U₁, U₂, U₃, U₄, définies par l'analyse. Dans ce cas, toutes les unités topologiques sont des objets simples. Le détail des noeuds 1, 2, 3, 4 représente la structure logique du document " Etat de Localisation des Eléments ". Les noeuds feuilles représentent le contenu, c'est-à-dire une instance de ce document.

3.2. Le modèle de réseau de documents

3.2.1. Le réseau de documents

Deux types de liens existent dans un réseau de documents :

- les liens sémantiques : de composition, de référence, de généralisation et d'utilisation ;
- le lien d'induction.

- Les liens sémantiques

• Le lien de référence

Très fréquemment, un document référence un ou plusieurs autres documents. Il aura pour paramètre la rubrique de référence.

Exemple : une facture se réfère à un bon de commande.

• Le lien de composition

Un document peut, pour des raisons de facilité, être décomposé en plusieurs sous-documents.

Exemple : une facture peut comporter plusieurs pages.

• Le lien de généralisation

Un document peut être défini comme général, et en fonction de son utilisation, il faudra en définir des spécialisations.

Exemple : une liste de clients peut être subdivisée en deux listes : celle des clients dont le chiffre d'affaires est supérieur ou égal à un certain seuil et celle des autres.

• Le lien d'utilisation

Un document donné utilise des rubriques qui se trouvent dans d'autres documents : il y a transfert d'information d'un document vers un autre. L'information transférée peut être de type rubrique ou contrainte d'intégrité (CI). Le document aura pour paramètres, la (ou les) rubrique(s) transférée(s), la (ou les) CI utilisée(s).

Exemple : dans une facture, l'adresse du client est ramenée du Bon de Commande que réfère ce même document. Le prix unitaire d'un produit est transféré de la nomenclature des Produits.

- Le lien d'induction

Le lien d'induction exprime une dépendance causale entre les documents. En effet, la création d'une instance de document peut impliquer la création d'autres instances.

Exemple : l'arrivée d'un bon de commande implique la génération, dans un certain ordre, de bon de livraison, facture...

Par analogie avec l'induction des événements dans la méthodologie orientée-objet OOM définie dans [SOU 96], nous distinguons différents paramètres pour ce lien d'induction :

- sa modalité : obligation ou possibilité
- son existence : conditionnelle ou inconditionnelle
- sa cardinalité : simple ou multiple
- sa contrainte temporelle : induction immédiate ou différée

- *La modalité*

La modalité du lien d'induction fait la distinction entre obligation et possibilité. En effet, l'obligation exprime qu'une instance de document **doit** être créée, alors que la possibilité exprime le fait que l'instance **peut** être créée.

Exemple : il est possible qu'un bon de commande induise un bon de livraison, alors qu'obligatoirement, un bon de livraison induit une facture.

- *L'existence*

L'existence d'un lien d'induction entre deux documents F_1 et F_2 peut être conditionnelle ou inconditionnelle. Elle est conditionnelle si une instance de F_2 est créée seulement si une certaine condition est vraie. Au contraire, le lien d'existence inconditionnelle implique que l'instance de F_2 sera créée systématiquement après la création d'une instance de F_1 . Ainsi, le paramètre d'induction sera la condition de création.

Exemple : La condition que les quantités commandées soient suffisantes pèse sur la possibilité du lien entre bon de commande et bon de livraison..

- *La cardinalité*

La cardinalité du lien d'induction exprime le fait qu'une instance de document induit une ou plusieurs instances d'un autre document. Ainsi, le paramètre sera le nombre d'instances de documents induits.

Exemple : une instance de bon de commande peut induire plusieurs instances de bon de livraison, si la livraison est fragmentée.

- *La contrainte temporelle*

L'induction entre deux documents peut être immédiate, c'est-à-dire qu'il n'y a pas de délai entre leurs états, ou bien différée si au contraire un laps de temps est nécessaire entre les deux. Ainsi, le paramètre d'induction sera un délai.

Exemple : un document de relance est élaboré, une semaine après l'émission d'un ordre d'insertion, si celui-ci n'est pas exécuté.

Une certaine chronologie de remplissage existe entre les instances de documents. Ainsi, il faudra d'abord générer un bon de commande, puis le bon de livraison associé et, enfin, la facture. Nous pouvons, donc, construire un réseau de documents dont les nœuds seront les documents eux-mêmes et les arcs, les liens de référence, de composition, de généralisation, d'utilisation, et d'induction.

3.2.2. Représentation du réseau de documents

Dans la représentation objet que nous utilisons, chaque document est une classe d'objets, reliée par les types de liens définis précédemment. Nous pouvons schématiser cette représentation par la figure ci-après.

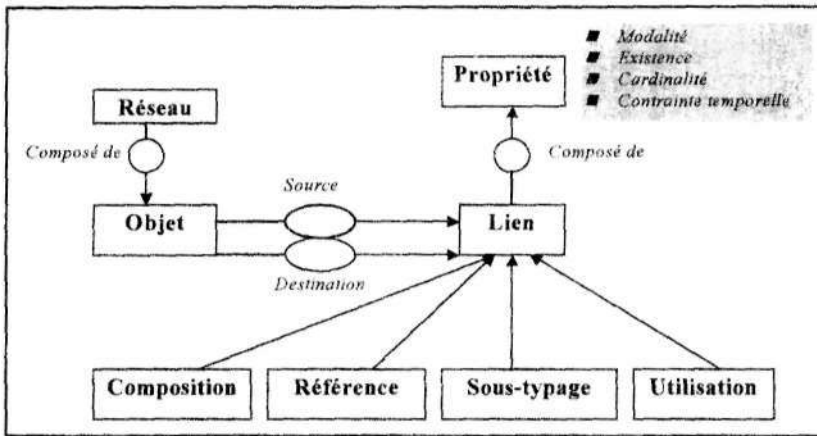


Figure 6. Le modèle objet de réseau de documents

Nous avons associé à chaque lien, un symbole graphique apparaissant dans la représentation suivante :

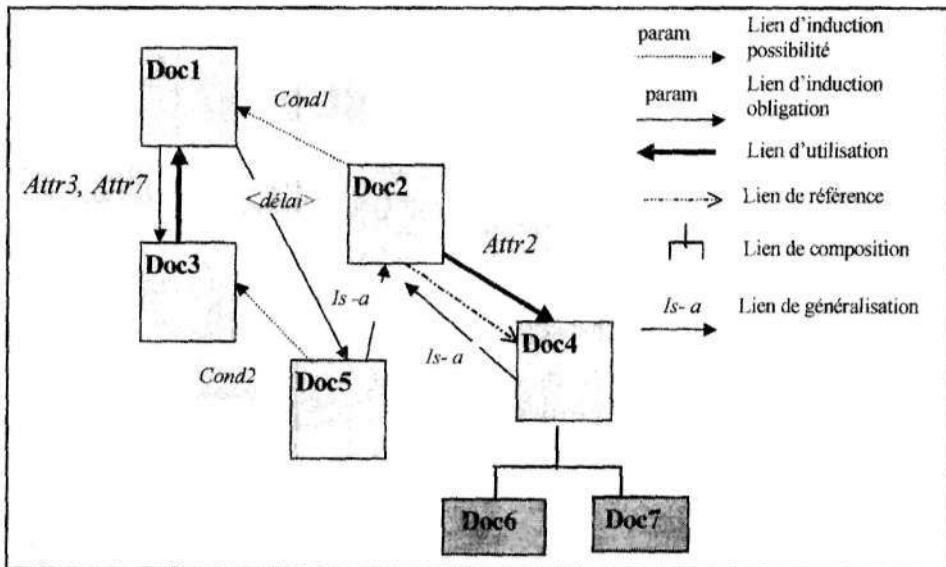


Figure 7. Un exemple de réseau de documents

Nous trouverons, en annexe (cf. figure 15), un exemple concret de réseau de document. Le réseau de documents a été exploité suivant deux axes :

- l'analyse directe des liens sémantiques et d'induction a permis de déduire une certaine dynamique des objets à partir de celle associée au réseau. Nous trouverons dans [ALI 97b] une étude se référant à un processus d'induction de la dynamique des objets à partir d'un réseau de documents.
- Il est complété par la définition d'un scénario.

3.3. Le modèle de scénario

Nous partons du principe qu'une description en langage naturel d'un scénario est peut être très significative pour un utilisateur, mais très difficilement formalisable. Des travaux ont été réalisés dans ce domaine [ROL 98] qui proposent une approche de construction de spécification textuelle de cas d'utilisation. Nous proposons une démarche de spécification de scénarios à partir d'une représentation tabulaire, plus facilement automatisable.

Définition (cf. figure 8)

Un scénario est une description de toutes les opérations effectuées par un acteur sur un ensemble de documents.

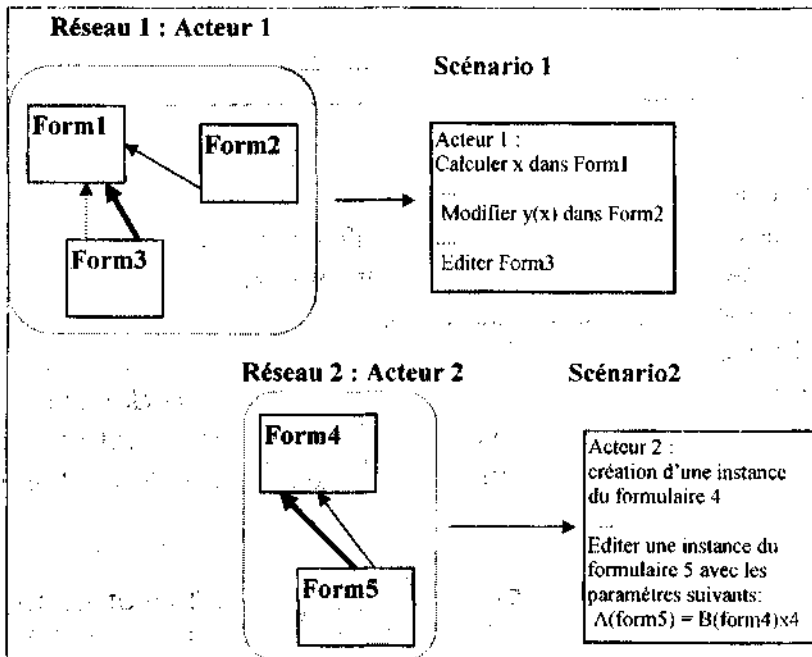


Figure 8. Scénarios

On trouvera en annexe, un exemple de scénario inspiré du cas d'utilisation de la redistribution manuelle des éléments dans des entrepôts, défini dans [JAC 92].

Pour des raisons d'homogénéisation avec l'analyse de documents, nous avons choisi de transcrire cette description en une représentation tabulaire dont la structure est la suivante (cf. figure 9) :

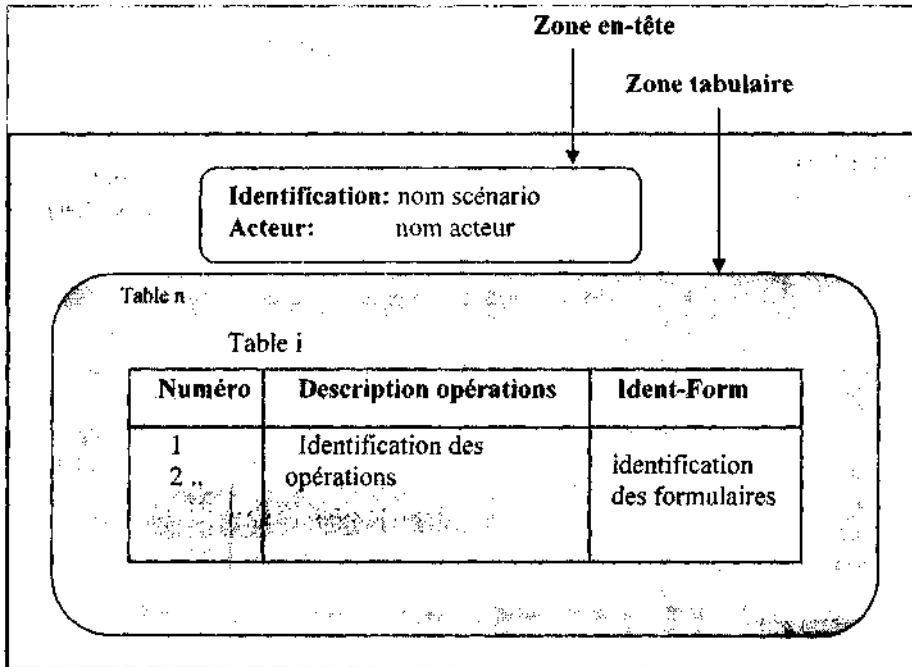


Figure 9. Représentation structurée d'un scénario

Un scénario comporte deux zones:

- une zone en-tête qui contient l'identification du scénario, c'est-à-dire un titre, et celle de l'acteur qui peut être n'importe quel utilisateur du système que nous voulons développer.
- la partie tabulaire du scénario contiendra trois colonnes : une première colonne pour identifier les lignes du tableau, une deuxième contiendra l'identification des opérations effectuées par l'acteur, et enfin, la troisième colonne fera référence aux documents de gestion utilisés dans le scénario (consultés, modifiés, créés).

La description d'un scénario peut être complexe. Il peut contenir plusieurs sous-scénarios formant une famille. Ces sous-scénarios peuvent être structurés en utilisant les formes d'abstraction courantes : séquence, alternative ou répétition. Une analogie avec les formes d'abstractions des modèles sémantiques [BRO 82] donne la correspondance respective suivante : séquence-agrégation, alternative-hiérarchie, répétition-collection.

Dans notre représentation , les composants seront des tables qui, modélisées comme objet, pourront être structurées comme agrégat, hiérarchie ou collection (cf. figure 10).

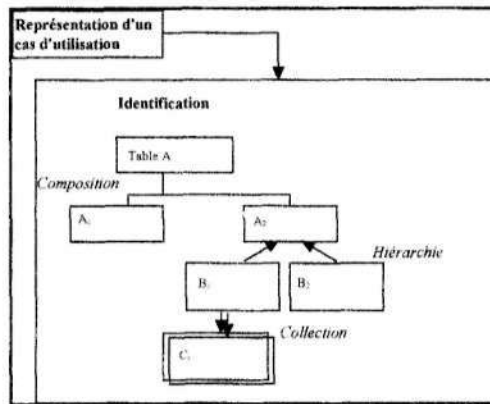


Figure 10. Structure hiérarchique d'un scénario

3.3.1. Le modèle de scénario

Nous avons précisé, dans la section 2, la nécessité du travail préalable de l'utilisateur pour fournir une description structurée des scénarios. Ainsi, la description en langage naturel du scénario sera transformée en une description structurée comme ensemble de tables. Cette structure est similaire à celle d'un document. Ainsi, un scénario sera considéré lui-même comme un document particulier. Les instances de ce document seront les instances du scénario. Ainsi, par similitude avec un document, nous définirons :

- **Un type de scénario** : il correspond structurellement à deux zones qui définissent le scénario soit les deux nœuds de l'arborescence (cf. figure 11).
- **Une instance de scénario** : elle sera déduite des opérations effectuées sur les instances de documents.

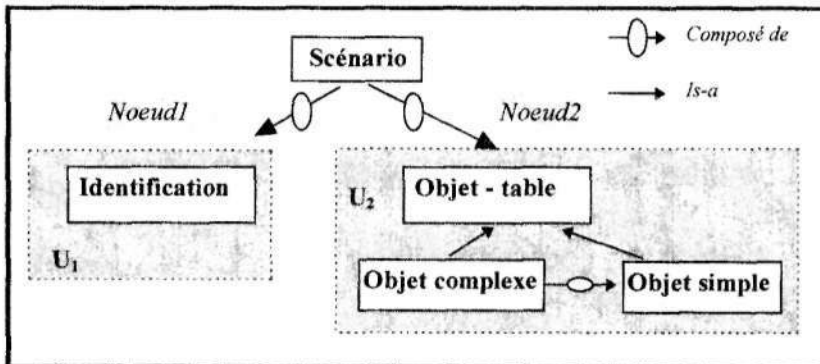


Figure 11 : Le modèle de Scénario

Exemple

Le scénario de la gestion de redistribution manuelle des éléments d'entrepôts peut être modélisé par la structure suivante (cf. figure 12) :

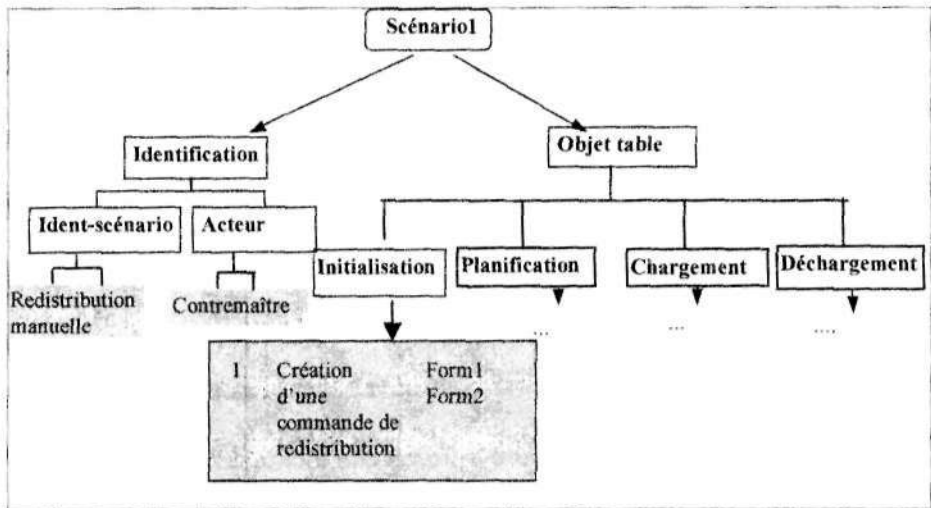


Figure 12. Un exemple de modèle de Scénario

Remarque

Pour ce cas particulier de scénario, l'objet-table est un objet complexe, composé d'un ensemble d'objets-tables simples qui sont les sous-scénarios d'initialisation, de planification, de chargement et de déchargement. Nous trouverons en annexe une description en langage naturel des sous-scénarios initialisation et planification, puis une description structurée du premier (cf. figure 16).

4. Présentation générale du processus

4.1. Architecture fonctionnelle

Le processus global se subdivise en deux grandes étapes (cf. figure 13) :

- une étape d'abstraction qui a pour produits résultats :

- une base de documents et une base de scénarios, construites à partir d'un modèle abstrait de documents,
- une description structurée d'un ensemble d'objets, dérivée de l'analyse de documents.

- une étape d'intégration qui a pour produit résultat les objets du SI. Cette étape est composée de deux phases :

- une intégration statique : les objets construits à l'étape précédente sont intégrés dans un schéma, sans tenir compte des liens qui peuvent être dérivés par l'interaction de documents. Le produit-résultat sera un schéma

statique objet c'est-à-dire un noyau de SI à compléter, enrichir et valider dans les étapes ultérieures.

- une intégration dynamique : elle fait intervenir d'une part le réseau de documents, et d'autre part les scénarios. Les scénarios sont utilisés pour construire les interfaces des objets en utilisant une table des opérations [ALI 97a], inspirée des diagrammes d'interactions de Jacobson repris dans la notation unifiée UML. Alors que les liens sémantiques entre documents vont être utilisés pour raffiner les liens sémantiques entre objets du SI, les liens d'induction vont permettre une induction d'un certain comportement d'un objet. L'analyse de ces liens nous a permis de définir une dynamique des objets [ALI 97b], basée sur le principe de causalité traité dans la méthode Remora.

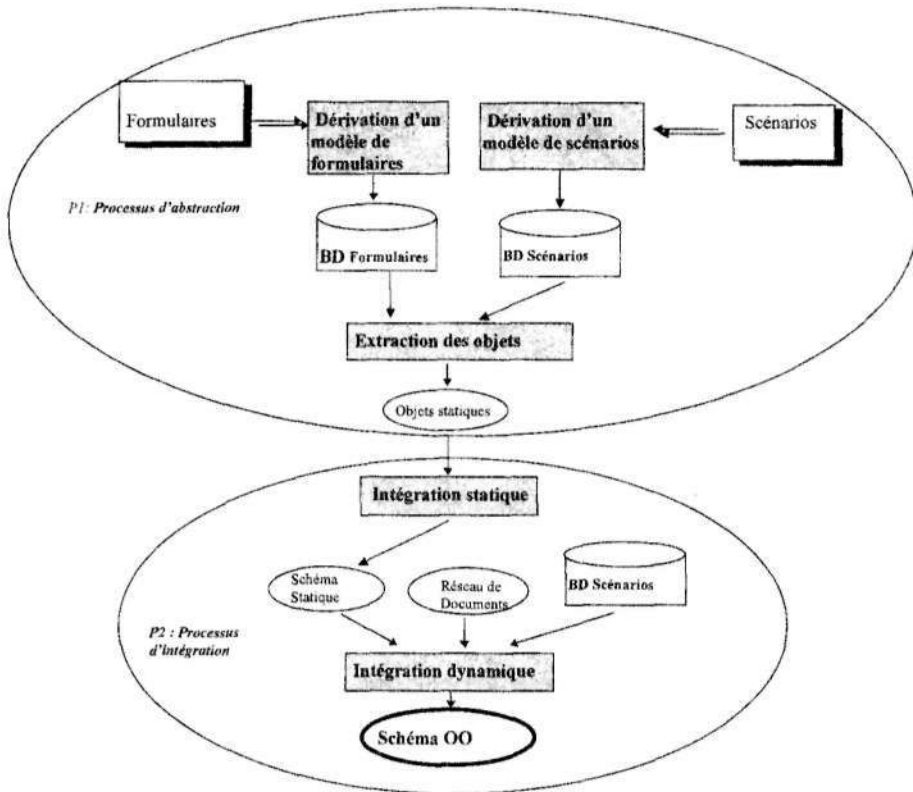


Figure 13 : Architecture fonctionnelle de notre système

Alors que le modèle abstrait défini peut prendre en compte n'importe quel type de documents, à travers les notions d'unité topologique et d'objets complexes, le processus de construction des objets du SI que nous avons élaboré est basé sur une restriction aux documents types. Cette restriction nous a été dictée par un souci d'efficacité, et la prise en compte des autres

types de documents s'effectuera dans une étape ultérieure.

La dérivation des objets à partir de la base de documents s'appuie sur le concept de **fragment de modèle** et de règles et heuristiques .

4.2. Extraction des objets

4.2.1. Concept de fragment de modèle et principe de construction

Définition

Un fragment de modèle est un schéma contenant :

l'ensemble des objets dérivés d'une instance de document ainsi que leur liens,

la liste des propriétés en instance d'analyse : une propriété en instance d'analyse est une propriété qui n'a pas encore été rattachée à un objet du document et sera donc mise en attente.

Principe de construction des fragments de modèles de SI

Le point de départ de celle-ci est la structure arborescente qui est la représentation abstraite du document. Le système de règles, que nous utilisons pour dériver les objets, est subdivisé en trois groupes dont nous fournissons une définition informelle et une énumération non exhaustive.

4.2.2. Présentation du système de règles

Le premier et le deuxième groupe de règles définissent l'expression de la relation de composition. Le troisième groupe permet l'expression des relations de référence et d'héritage.

• Premier groupe de règles : identification des objets

Les trois règles suivantes sont basées sur l'analyse de l'arborescence exprimant le document et sur la correspondance noeud-objet.

Règle 1 : Tous les champs appartenant à un même noeud topologique, caractérisé par une structure simple (composition dont tous les noeuds fils sont des noeuds terminaux) peuvent représenter un objet simple. Le nom de l'objet est le nom du champ composite, les attributs de l'objet sont les noms des champs composants.

Règle 2 : Tous les champs appartenant à un même noeud topologique caractérisé par une structure complexe (plusieurs niveaux de noeuds) peuvent représenter un objet complexe.

Règle 3 : Tout objet complexe peut être décomposé en un ou plusieurs objets simples ou complexes en utilisant l'arborescence topologique et la règle 1.

- **Deuxième groupe de règles : raffinement des objets par analyse des attributs**

Les règles précédentes ont déjà permis de rattacher les attributs aux objets. Un travail de raffinement est nécessaire afin de déceler des objets non homogènes.

Les deux règles suivantes ont été définies sur la base d'une analyse de termes :

« Un terme est un symbole conventionnel représentant une notion définie dans un certain domaine de savoir » [FEL 87].

Le domaine de savoir que nous considérons est le domaine de l'application de gestion, d'où la nécessité de réaliser une représentation terminologique de ce domaine.

Règle 4 : Identification des clés candidates d'un objet

Tous les champs commençant par les termes numéro, num, nom,... peuvent identifier un objet.

Règle 5 : Rattachement des attributs par analyse de la partie droite

Tous les champs ayant une partie droite semblable ou rappelant le nom de l'objet sont rattachés au même objet.

La règle suivante permet l'élimination de propriétés complexes particulières, c'est-à-dire celles dont la complexité est une combinaison mathématique de propriétés simples.

Règle 6 : Chaque attribut calculé est, de manière itérative, remplacé par les attributs élémentaires, retrouvés grâce au parcours de l'arbre arithmétique de son expression de calcul.

La règle suivante est basée sur le fait que le nombre d'occurrences des champs est un facteur déterminant dans la relation de composition

Règle 7 : Les attributs d'une instance de document ayant le même nombre d'occurrences sont susceptibles d'appartenir au même objet.

- **Troisième groupe : raffinement des objets par analyse des liens entre les objets**

Règle 8 : Si une valeur d'un champ identifiant d'un objet O_1 est toujours associée à une valeur d'un champ identifiant d'un objet O_2 dans un même document, alors il existe un lien de référence entre O_1 et O_2 et l'attribut identifiant de O_1 est intégré dans la liste des attributs de O_2 .

Règle 9 : Il existe un lien d'héritage entre deux objets O_1 et O_2 si l'arborescence de O_2 se retrouve dans l'arborescence de O_1 , O_1 et O_2 pouvant appartenir au même document ou à deux documents différents.

Le processus d'abstraction, sur la base d'un modèle abstrait de document, permet donc de générer une base de documents, contenant tant une représentation des types de documents utilisés (structure topologique et structure logique), que leurs instances ou noeuds feuilles de l'arborescence. Ce même modèle, appliqué aux scénarios, qui ne sont qu'une description des traitements effectués sur des réseaux de documents, permet la génération d'une base de scénarios, ensemble de description de tables. Par ailleurs, en utilisant des règles de dérivation, un ensemble de fragments de modèle associé aux instances de documents est défini. Les fragments de modèle, la base de scénarios et le réseau de documents, sont le point de départ du processus d'intégration, qui permet la génération du schéma des objets du SI.

4.3. Intégration

Le processus d'intégration se base sur une vision globale du réseau de documents afin de construire le schéma des objets. Il se compose de deux étapes qui correspondent à une intégration à deux niveaux :

- **Une intégration statique des objets** : les objets sont construits sans tenir compte des liens qui peuvent être dérivés par l'interaction des documents. Le produit résultat sera un noyau de SI à compléter, enrichir et valider dans les étapes ultérieures.
- **Une intégration dynamique des objets** : L'intégration dynamique comporte deux grandes phases :
 - une phase de définition des liens inter-objets qui utilise le réseau de documents,
 - une phase de construction des interfaces des objets qui utilise une représentation structurée d'un scénario et les diagrammes d'interaction associés.

4.3.1. L'intégration statique

4.3.1.1. Définition

Habituellement, lorsqu'on fait de l'intégration de schémas, les problèmes rencontrés sont de plusieurs types et principalement :

- syntaxique : conflit de noms, synonymie...
- sémantique : différence d'appréciation dans deux vues différentes, par exemple de l'information est classée entité dans une vue et association dans l'autre..
- cohérence des contraintes d'intégrité (CI) après l'intégration.

Dans le cas présent, le principe d'intégration est considérablement simplifié dans la mesure où les problèmes sémantiques disparaissent. En effet, la dérivation des objets s'effectue de la même manière pour n'importe quelle instance de documents, c'est-à-dire, par l'application des règles et heuristiques qui ne laisse pas place à de l'interprétation sémantique.

Les problèmes syntaxiques recensés sont, de façon non exhaustive, les suivants :

- synonymie et homonymie de nom d'objets
- synonymie et homonymie de noms d'attributs

Les problèmes associés aux contraintes d'intégrité peuvent porter sur :

- les incohérences sur les formules de calcul ou sur toute autre règle de gestion
- les incohérences sur les domaines de valeurs...

L'intégration statique est définie à partir de la notion de sous-schéma intégré.

Définition

Un sous schéma intégré (SSI) est un fragment de modèle obtenu itérativement par intégration de fragments. Le dernier SSI obtenu après épuisement des instances de documents sera le schéma statique objet.

4.3.2. Description du processus

Il s'effectue suivant les étapes suivantes :

4.3.2.1. Initialisation

Le vocabulaire utilisé dans l'ensemble des documents est loin d'être homogène. Dans une première phase il s'agit donc de baliser le domaine d'investigation. Ce travail sera fait sur la base d'un échantillonnage des documents. Cet échantillon permet de construire une représentation terminologique du domaine. Un travail conjoint avec l'utilisateur permet de réaliser une équivalence de terminologie. Le résultat de cette phase permettra de créer un dictionnaire d'équivalence de termes qui sera utilisé dans la phase suivante.

4.3.2.2. Traitement des objets redondants ou synonymes

Tout objet dupliqué (c'est-à-dire ayant le même nom) dans un fragment et dans un SSI sera fusionné. Un travail d'union de ses propriétés est effectué afin de garder un recouvrement maximal. Il sera gardé les références aux instances de documents dont il dérive. Deux objets ayant des noms différents peuvent être synonymes et la vérification s'effectue par la table de terminologie : ils seront alors fusionnés et le même nom leur sera attribué.

4.3.2.3. Traitement des attributs en instance d'analyse

Deux catégories d'attributs en instance d'analyse sont définies, ceux résultant du fragment de modèle en entrée, et ceux apparaissant dans le sous schéma intégré. Il sera appliqué à ces rubriques les règles de rattachement des attributs définis pour l'étape de construction d'un fragment.

4.3.2.4. Raffinement des objets par analyse des liens sémantiques inter-objets

Dans cette étape, on se basera sur le réseau de documents, et tous les objets auxquels nous faisons allusion seront les objets primaires.

* **Le lien de référence** : Le lien de référence entre deux documents F_1 et F_2 se transforme en lien de référence entre un (ou plusieurs) objet(s) de F_1 et un (ou plusieurs) objet(s) de F_2 . On note :

F_1 ref F_2 è (Obj) $_{F_1}$ ref (Obj) $_{F_2}$

Ainsi, par exemple, le document de redistribution (F_3) fait référence au document Demande de redistribution (F_1), l'objet dérivé de F_3 , «Redistribution», fera référence à l'objet dérivé de F_1 «Demande de redistribution ».

* **Le lien de composition** : Le lien de composition entre deux documents F_1 et F_2 se transforme en un lien de composition entre un (ou plusieurs) objet(s) de F_1 et un (ou plusieurs) objet(s) de F_2 . On note :

F_2 comp F_1 è (Obj) $_{F_2}$ comp (Obj) $_{F_1}$

Ainsi, par exemple, si le document de redistribution (F_3) était composé de deux sous documents " redistribution des articles pièces de rechanges " (F_{31}), et " redistribution es articles équipements informatiques " (F_{32}). l'objet " redistribution " sera composé de deux objets dérivés respectivement de F_{31} et F_{32} , " redistribution pièces de rechange " et " redistribution équipements informatiques ".

* **Le lien de généralisation** : Le lien de généralisation entre deux documents F_1 et F_2 , se transforme en un lien de généralisation entre un (ou plusieurs) objet(s) de F_1 et un (ou plusieurs) objet(s) de F_2 . On note :

F_2 is-a F_1 è (Obj) $_{F_2}$ is-a (Obj) $_{F_1}$

Ainsi, par exemple, si on considère un Etat de planification des transports qui se spécialise en deux états : Etat des transports région parisienne et Etat des transports région lyonnaise, il sous-entend une spécialisation de l'objet " planification " en deux objets " planification région parisienne " et " planification région lyonnaise ".

* **Le lien d'utilisation** : Le transfert d'informations entre deux documents F_1 et F_2 , permet lorsqu'on remonte vers le document source de l'information, d'éviter la redondance d'objets ou de propriétés. En fait ce lien, comme dans le domaine de la conception orientée objet, établit un chemin de

communication entre deux documents. Il permet en fait d'établir des messages entre documents. Ce lien pourrait être utilisé pour définir, par le biais de messages entre objets, des opérations d'accès. On note :

F₁ utilise F₂ è (Obj)_{F₁} accède (Obj)_{F₂}

Ainsi, le document de redistribution F₃ utilise le nom Entrepôt de F₂ (état de localisation des éléments).

Le passage d'une étape à une autre permet le raffinement du SSI, jusqu'à obtenir un SSI final par itération du processus. Il y aura autant d'itérations que d'instances de fragments de modèles. La figure ci-après traite de l'intégration statique des fragments 1, 2, et 3 de l'exemple de la figure .

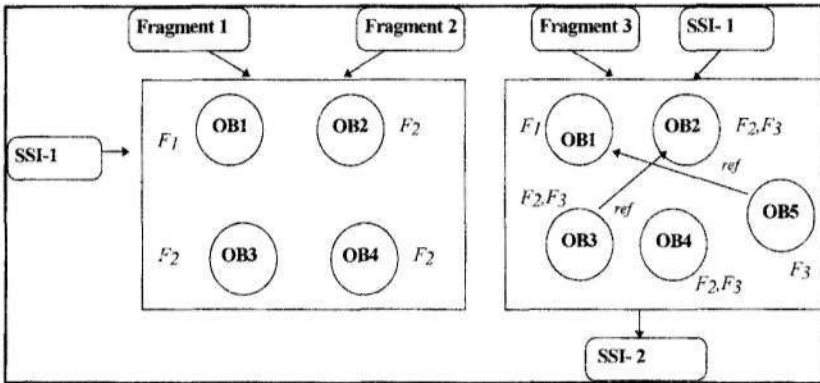


Figure 14: Exemple d'intégration statique

4.3.3. L'intégration dynamique

Le schéma suivant résume cette phase :

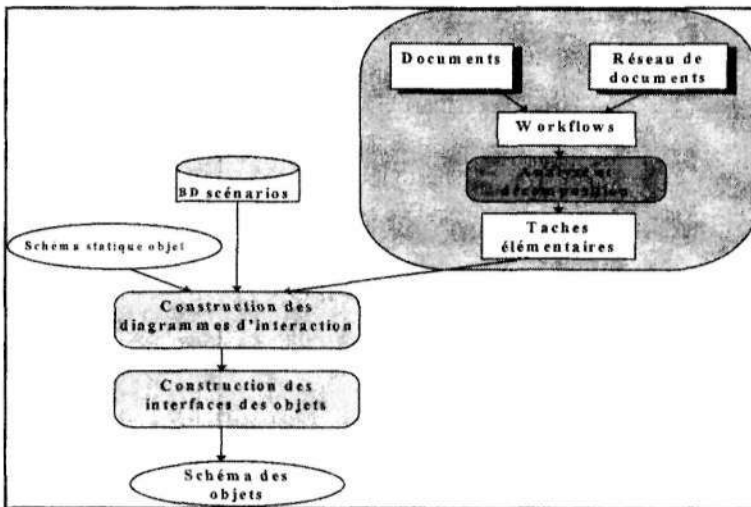


Figure 15: architecture générale de la phase d'intégration

Les étapes de l'intégration dynamique sont présentées succinctement dans les sections suivantes.

4.3.3.1. Phase de construction des diagrammes d'interaction

Le diagramme d'interaction est une représentation tabulaire de toutes les interactions entre les différents objets résultats de l'étape d'analyse de documents. A la description des opérations sera associée une représentation graphique permettant d'associer l'opération à l'objet impliqué.

Exemple: nous traitons le sous-scénario d'initialisation de la redistribution manuelle des éléments. Le diagramme d'interaction du sous-scénario planification est donné en annexe 1.

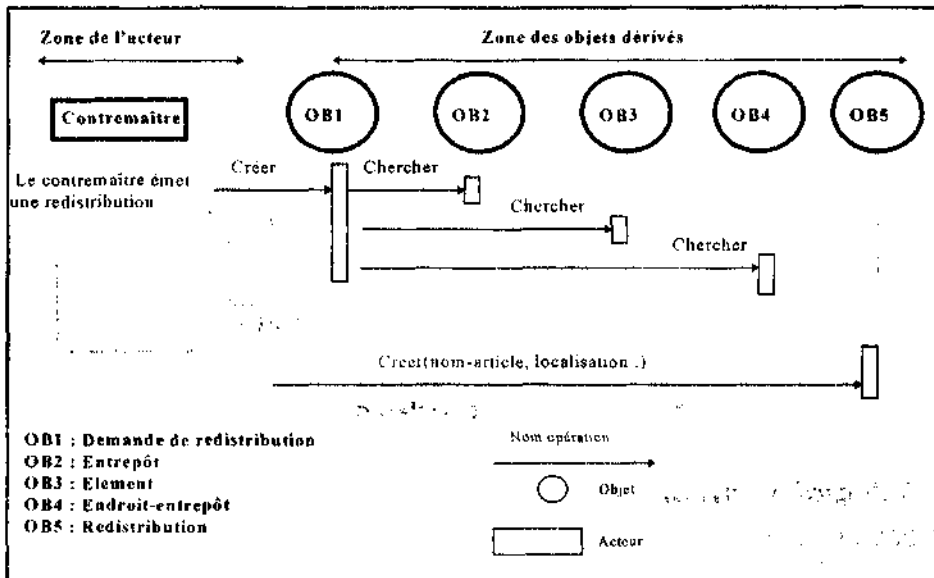


Figure 16: Représentation structurée d'un diagramme d'interaction

La zone de l'acteur va contenir le nom de l'acteur, ainsi que toutes les opérations qu'il va effectuer dans le scénario utilisé. La zone des objets contient tous les objets dérivés par les documents impliqués par le scénario. Une flèche est étiquetée par le nom de l'opération, et cible un objet.

4.3.3.2. La construction des interfaces des objets

Le processus de construction des interfaces des objets se compose des trois étapes suivantes.

- Le processus de construction des diagrammes d'interaction

Il est la séquence d'opérations suivantes :

1. parcourir une table du cas d'utilisation considéré en suivant l'ordre hiérarchique afin de lister les documents ciblés .

2. Analyser le schéma statique objet et sélectionner les objets dérivés de cet ensemble de documents.
3. Présenter à l'utilisateur la structure pré-définie du diagramme d'interaction. La structure pré-définie est le nom de l'acteur suivie par la liste des objets dérivés (en gras dans le diagramme de la figure 16).
4. Construction de l'instance de diagramme. La représentation structurée du scénario est interprétée et associée à une analyse de tâches afin de dégager les interactions sur les objets (étiquetage des flèches).
5. Les opérations 1 à 4 sont répétées jusqu'à épuisement des tables composant le cas d'utilisation.

- Le processus d'identification des opérations

Il est la séquence d'opérations suivantes :

1. Sélection du jeu d'instances des diagrammes d'interaction.
2. Sélection d'un objet .
3. Parcourir toutes les instances de diagramme sélectionnées, et sélectionner les opérations qui ciblent cet objet afin de compléter sa description.
4. Répéter les opérations 1 à 3, pour tous les objets d'une instance et de toutes les instances de diagrammes d'interaction.

- Le processus d'intégration des matrices d'opérations

5. Conclusion

Nous avons développé une méthodologie qui permet d'extraire des connaissances à partir d'une analyse de l'existant non classique. Nous sommes partis d'éléments familiers aux gestionnaires (documents et scénarios), afin de définir un processus incrémental de génération d'objets. Le processus se base sur un modèle abstrait de documents, arborescence d'unités topologiques et logiques, où chaque noeud est représenté par le concept d'objet. Dans un premier temps, seules des unités topologiques pertinentes pour le SI ont été prises en compte. Nous avons exploité la circulation des documents dans les différents postes de travail afin de construire un réseau de documents basé sur la définition de liens sémantiques et d'induction. L'analyse de ces liens, couplée avec les entités dérivées des documents, permet d'enrichir les objets par adjonction des liens sémantiques.

Nous avons utilisé les relations scénarios-documents afin de capter la dynamique des objets. Pour cela, nous passons par une représentation structurée d'un scénario, puis de sa traduction à un modèle objet grâce au modèle abstrait de documents.

Le schéma statique est dérivé à partir d'un ensemble non exhaustif de règles et d'heuristiques. Les connaissances que nous extrayons, même si elles ne représentent pas complètement le SI, peuvent servir de base d'exploration pour les développeurs. Le résultat de ce travail pourra être complété par des méthodologies plus performantes et permettre ainsi de construire le SI futur. Il peut aussi être exploité dans un cadre de réutilisation puisque le processus autorise la modification ou l'extension du schéma objet par ajout d'instances de documents ou de scénarios.

Un outil de guidage des processus d'analyse a été développé sur la méthode utilisant un méta-modèle de processus[ALIM OO]. Par ailleurs, le modèle de document ainsi que le processus d'abstraction ont été utilisés pour préciser une méthode de développement de SI géographiques [ABD 02].

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques. Le processus d'abstraction est un processus itératif qui permet de passer d'un document à un schéma objet. Ce processus est basé sur des règles et des heuristiques.

BIBLIOGRAPHIE

[ABD 02] N. Abdat, " GOMT, une méthode de développement orientée objet de Systèmes d'Information géographique, thèse de Magister, USTHB 2002

[ALI 95] Z. Alimazighi, K.Akli, L. Ouahrani, " Un atelier intelligent de développement de SI " Second International Symposium on Programming and Systems, Algiers 1995.

[ALI 96] Z. Alimazighi, " Une Méthodologie de Génération de Schémas Conceptuels orienté objet à partir d'une analyse de formulaires " Rapport de recherche CRI 96-10.

[ALI 97a] Z.Alimazighi, C. Rolland, " Génération d'Objets par une Analyse couplant les Formulaires et les Scénarios ", Congrès INFORSID'97, Toulouse 1997.

[ALI 97b] Z. Alimazighi, " Processus d'induction de la dynamique de SI orientés-objets à partir de réseaux de documents ", 1st Workshop on the Many Facets of Process Engineering, Gammarth, Tunisie 1997.

[ALI 99] Z.Alimazighi, " Environnement de développement de systèmes d'information : une approche méthodologique basée sur l'analyse de documents et scénarios ". Thèse de doctorat, USTHB Mars 1999.

[ALI 00] Alimazighi Z., Ferrad K., Lazizi F., " Un outil de guidage des processus d'analyse des systèmes d'information ", 3^{ième} séminaire national en informatique de Biskra , Mai 2000

[BAN 92] Bandeira M, Bouzeghoub M, Metais E, Saleh M, "Extraction de Connaissances à partir de formulaires pour la modélisation de bases de données ", Rapport MASI 92.19.

[BAT 88] C.Batini, G.Di Battista, "A Methodology For Conceptual Documentation and Maintenance ", Information Systems; Vol13 1988.

[BAT 92] C.Batini, Ceri, Navathe, " Conceptual Database Design: An Entity relationship approach ", Benjamin/Cummings Publishing 1992.

[BEN 93] Benner K.m, Feather M.S, Johnson W.L, Zorman L.A, " Utilizing scenarios in the software development process ", Information System Development Process, IFIP 1993.

- [BEN 01] Belahcene D.M. " Une démarche de développement de Systèmes d'Information par fragments de méthodes " thèse de Magister Octobre 2001
- [BOO 91] G. Booch, " Object Oriented Design With Applications ", Benjamin Cumming Ed, 1991.
- [BOU 85] M. Bouzeghoub, G.Gardarin, E.Métais, " Database Design Tool : An expert system Approach " Proceedings of the 11th VLDB conference, Stockholm Sweden 85.
- [BOU 97] M. Bouzeghoub, " Unification des Méthodes Objets : fusion ou confusion ", conférence invitée, congrès INFORSID'97, Toulouse 1997.
- [BRO 82] M.L. Brodie, E. Silva, " Active and Passive Component Modeling : ACM/PCM " IFIP 1982.
- [BRU 93] J. Brunet, " Analyse conceptuelle orientée - objet " Doctorat de l'université de Paris VI - 1993.
- [CAM 93] G. Cambolive, M. Khemaja, " D'un système de conception et d'exploitation de documents vers un système à base de connaissances. (application à l'Audit bancaire) ", Congrès INFORSID'93.
- [CAU 88] C. Cauvet, " Un Modèle et un Outil d'aide à la conception de systèmes d'information ", Doctorat de l'université de Paris 6, 1988.
- [CBB 99] C. Ben Achour , " Extraction des besoins par analyse de scénarios textuels, Thèse de l'université de Paris 6, 1999.
- [CHO 88] J. Choobineh, M.V. Mannino, J.F. Nunamaker, B.R. Konsynski , " An expert Database design system based on analyzed of forms ", IEE Transactions on Software Engineering, Vol. 14, 1988.
- [CHO 92] J. Choobineh, M.V. Mannino, V.P. Tseng, " The role of form analysis in computer-aided software engineering ", Conceptual modeling databases and case: An integrated view of Information Systems development: 1992.
- [COL 92] C. Collet, E. Brunel, "Définition et manipulation de formulaires avec FO2 ", TSI- Vol 10, 1992.
- [COL 94] C.Collet, T.Coupaye, T.Svensen, " Efficient and modular reactive capabilities in an Object-Oriented Database System ", BDA' 94.

- [DAN 97] B. Dano, H.Briand, F.Barbier, " Producing object-oriented dynamic specifications : an approach based on the concept of use case ", RE'97.
- [GOU 95] P.A Gough, F.T Fodemski, S.A Higgins, S.J Ray, " Scenarios- An industrial case Study and Hypermedia Enhancements ", 2nd IEEE International Symposium on requirements Engineering York, England 1995.
- [GRO 94] C. Rolland, G. Grosz, " A General Framework for describing the Requirements Engineering Process ", IEEE conference on systems Man and Cybernetics, CSMC'94, San Antonio, Texas 1994.
- [HOL 90] Holbrook H, " A Scenario-based Methodology for conducting requirements elicitation ", ACM SIGSOFT, vol. 15 n° 1 Jan 90.
- [HSI 94] P. Hsia, J . Samuel, J .Gao, D . Kung, Y. Toyoshima, C. Che, " Formal Approach to Scenario Analysis ", IEEE software 1994.
- [ISO 86] ISO, Information Processing- Text and office systems- Standart Generalized Markup Language (SGML), ISO 8879, 1986.
- [ISO 89] ISO, Information Processing - Text and office systems- Office Document Architecture (ODA) and Interchange format, ISO 8613, 1989.
- [JAC 92] Jacobson, I. et al. " Object - Oriented Software Engineering, A Use Case Driven Approach ", Addison Wesley, 1992.
- [JAC 95a] I. Jacobson, M.Ericsson, A.Jacobson, " The Object Advantage: Business Process Reengineering with Object Technology " ACM Press Books 1995.
- [JAC 95b] I. Jacobson, " The use-case construct in object-oriented software engineering ", in J.M. Carroll editor, Scenario-Based Design Envisioning Work and Technology in System Development, pages 309-336. John Wiley and Sons, 1995.
- [KHO 98] S. Khoshafian, M. Buckiewicz, "Groupware et Workflow" Masson 1998
- [LEE 94] Lee S.P., " Formalization and Automatic support for Conceptual modeling " Ph.D. thesis, university of Paris1, 1994.
- [LUB 93] Lubars M, Potts C, Richter C, " Developing Initial OOA Models ", Proceedings of the 15th Conference on Software Engineering 1993.

- [MAR 96] P.Maret, L.Poullet, J.M. Pinon, " Des modèles conceptuels pour capitaliser la connaissance au sein d'une organisation ", Ingénierie des SI vol.4 n°4/1996
- [MIZ 91] M. Mizi, " Outil interactif de conception et de manipulation de formulaires " INFORSID91.
- [MUL 01] P.A Muller, " Modélisation Objet avec UML " Eyrolles 2001
- [OLL 89] T.W. Olle & al., " A framework for the comparative evaluation of information systems methodologies ", Addison-Wesley, 1989.
- [POT 94] C.Potts, K. Takahashi, A.I Anton, "Inquiry based requirements analysis" . IEEE Software 1994.
- [POZ 93] S.Pozzi, A. Celentano, "Knowledge-based document filing ", IEEE Software 1993.
- [REG 95] B. Regnell, K. Kimbler, A. Wesslen, " Improving the use case driven approach to requirements engineering ", 2nd IEEE international symposium on requirement engineering, York, England 1995.
- [RUB 92] K.S. Rubin, A.Goldberg, " Object Behavior Analysis " Communications of the ACM, Vol.35, N° 9, 1992.
- [RUM 95] J.Rumbaugh et al. " OMT, Modélisation et Conception Orientées Objet ", Paris Masson, 1995.
- [RUM 96] J. Rumbaugh, G. Booch, " Unified Method ", Notation Summary , version 0.8, Rational Software Corporation.
- [ROL 86] C.Rolland, C.Proix, " An expert system approach to information system design ", IFIP world congress Dublin , Irlande 1986.
- [ROL 87] C. Rolland, O. Foucaut, G. Benci, " Conception des Systèmes d'Information : la méthode REMORA ", Eyrolles 1987.
- [ROL 93] C. Rolland, " Modeling the Requirements Engineering Process ", Information Modeling and Knowledge Bases, IOS Press, 1993.
- [ROL 97] C. Rolland, " Ingénierie des Méthodes ", conférence invitée, congrès

INFORSID'97, Toulouse 1997.

[SAA 96] M. Saadoun " Le projet Groupware " Eyrolles 1996.

[SCH 88] S. Schlaer, S.J. Mellor, " Object Oriented System Analysis : Modeling the World in Data ", Yourdon Press, 1988.

[SIS 96] S. Si-Said, C. Rolland, G. Grosz , " MENTOR : A Computer Aided Requirements Engineering Environment ", CAISE'96.

[SOU 96] C.Souveyet, R.Denecker, Manuel de référence de la méthode TOOM, Projet ESPRIT, " TOOBIS ", 1996

[SPA 89] S. Spaccapietra, C. Parent, " About Entities , Complex Objects and Object Oriented Data Models ", Proc. of the Working Conference on Information System Concepts, Namur, Belgium, October 1989.

[TAL 90] B. Talldal, B.Wangler, "Extracting a Conceptual model from examples of filled in forms" Data Management Current Trends: Editor Naven Prakash.

[TAR 91] H. Tardieu, A. Rochfeld, R. Colletti, "La Méthode Merise : Principes et Outils ", Paris, Editions d'organisation, 2^{ème} édition, 1991.

ANNEXE

Exemple d'application

Nous avons adapté l'exemple de gestion d'entrepôt traité dans [JAC 92]. Nous n'avons considéré que deux sous-scénarios, celui de l'initialisation de la distribution et celui de la planification que nous présentons en langage naturel simplifié ci-après.

Description du sous-scénario 1 " Initialisation "

- Le contremaître commande (F₁) une redistribution entre les entrepôts.
- Il consulte l'état de localisation (F₂) des éléments.
- Il sélectionne les éléments à déplacer et vers quels entrepôts et en quelles quantités qu'il place dans le formulaire de redistribution (F₃).

Description du sous-scénario 2 " Planification "

- Le contremaître va consulter la planification.
- Le contremaître va modifier les demandes de transports déjà planifiées.
- Les demandes de transports sont associées à un plan de transport par poids lourd donné.

Nous considérons le sous-scénario 1. Trois formulaires ont été définis, F₂ ayant déjà été présenté dans le corps de l'article les autres formulaires sont les suivants (cf. figure 13 et figure 14) :

DEMANDE DE REDISTRIBUTION	
Date : 12/09/02	Numéro : 456

Figure 17 Description du formulaire de demande de redistribution

FORMULAIRE DE REDISTRIBUTION			
			Redistribution n°: 123
			Date: 13/09/02
Nom-article	Localisation	Nom-entrepôt-destinataire	Quantité
Ordinateurs	PD32	Hydra	10
Vis 12mm	PA12	Hydra	100
Bananes	PC12	Ain Bénian	tous

Figure 18 Description du formulaire de redistribution

Les liens entre les formulaires F₁, F₂, F₃ sont représentés dans la figure 15 suivante :

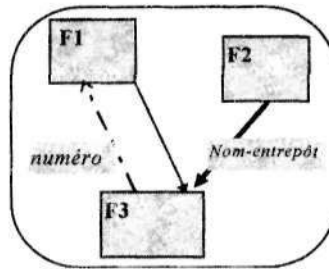


Figure 19 Description du réseau de formulaires

Identification : sous-scénario 1		
Acteur : Contremaître		
1	Créer demande de redistribution	F ₁
2	Rechercher état de localisation	F ₂
3	Chercher entrepôt	F ₂
4	Chercher élément	F ₂
5	Définir quantité	F ₂
6	Chercher endroit-entrepôt	F ₂
7	Créer redistribution	F ₃

Figure 20 Description structurée du sous-scénario 1

Application de l'étape de construction des fragments:

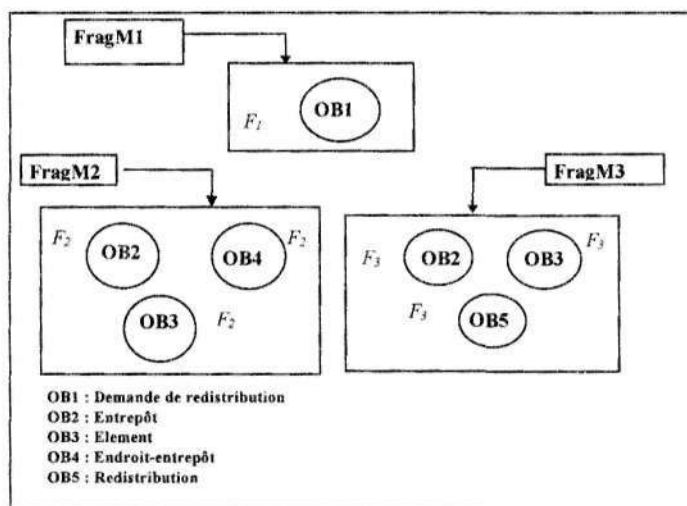


Figure 21 Les fragments de modèle associés aux formulaires F₁, F₂, F₃