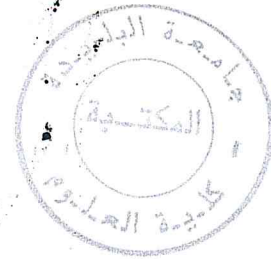


République Algérienne Démocratique et Populaire.  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saad Dahlab, Blida  
USDB.

Faculté des sciences.  
Département informatique.



**Mémoire pour l'obtention  
d'un diplôme d'ingénieur d'état en informatique.**

Option : IA

Sujet :

Réalisation d'un outil qui permet la modélisation graphique d'un  
processus Workflow et qui génère la spécification associée dans un  
schéma XML

**Présenté par :**  
BERRABAH BENAÏSSA  
BENTOUT RACHID

**Promoteur :** M<sup>me</sup> OUKID-KHOUS  
**Encadreur :** M<sup>me</sup> RAHAL-BOUMAHD

**Organisme d'accueil :** Université de Blida

**Soutenue le:** 29/06/2005, devant le jury composé de :

Mme ABED

Mme OUAHRANI

Mr BALA

**Président**

**Examineur**

**Examineur**

- /2004/2005-



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# Remerciements

*Au terme de ce travail nous remercions :*

*En premier lieu le bon Dieu qui nous a données la force à achever cette réalisation.*

*Nos familles pour ses compréhensions et ses soutiens.*

*Les membres du jury pour nous avoir fait l'honneur de juger notre travail*

*Notre promotrice Mme OUKID-KHOUAS pour son aide, sa patience et sa compréhensibilité.*

*Notre encadreuse Mme RAHAL-BOUMAHDHI pour son aide, sa disponibilité et sa patience.*

*Tout les professeurs qui ont contribués à notre formation durant les années d'étude, on particulier le chef du département Mme BENSTITI pour sa compréhensibilité.*

*Mr Djamel, Le responsable de la bibliothèque de la faculté des sciences et son équipe.*

*En fin toute autre personne qui a contribué a ce travail de loin ou de prés.*

# *Dédicace*

*Je dédie ce travail :*

*A tous les membre de ma famille, commençant par mes très chers parents MOKHTAR et SAFIA pour leur soutien durant toute ma carrière et pour leur encouragement.*

*A mes frères MOUHAMED, ABDEL RAHMEN, ABDELLAH  
A mes seours ZAHRA, SAMIRA et son mari RACHID et leurs fils  
ABDEL SALEM.*

*A tous mes amis sans exception particulièrement ISMAIL, HALIM,  
HAMZA, SAMIR et SALAH.*

*A tous les membres de groupe de SMA-AFAK OUELED MENDILE  
A mon amis NASSIM et à toute sa famille en particulier  
Leur père Mr Kamel.*

*A mon binôme BENAÏSSA et à toute sa famille  
A toute ma PROMO*

**BENTOUT RACHID**

## *Dédicace*

*Je dédie ce travail :*

*A tous les membre de ma famille, commençant par mes très chers  
parents ABDEL MALEK et KHADOUDJA pour leur soutien durant  
toute ma carrière et pour leur encouragement.*

*A mon grand père BENAISSA.*

*A mes frères MOUHAMED, MOUSTAPHA, BOUELEM*

*A mes seours AICHA, NESSRINE, ASMAE, NAWOL et MOUNIA  
et sa fille IKRAME.*

*A tous mes amis sans exception particulièrement ISMAIL, HALIM,  
HAMZA, SAMIR et SALAH.*

*A mon amis NOUR EDINE et à toute sa famille  
en particulier ses parents.*

*A mon binôme RACHID et à toute sa famille*

*A toute ma PROMO*

**BERRABAH BENAISSA**

**TABLES DES MATIERS**

<b>Introduction Générale</b> .....	1
<b>Chapitre I. Le groupware et workflow</b> .....	4
1. Introduction.....	4
2. Le groupware .....	4
2.1 Les concepts.....	5
2.1.1. La communication.....	5
2.1.2. La collaboration .....	5
2.1.3. La coordination .....	6
2.2 Les règles d'or du groupware.....	6
2.2.1. Etre conscient que le groupware n'est pas qu'un produit logiciel .....	6
2.2.2. Un projet groupware est un véritable projet de management .....	7
2.2.3. Un projet groupware et un véritable processus de changement .....	7
2.3 Typologie des applications groupware .....	7
2.3.1. La famille des applications orientées "Mémoire" .....	8
2.3.2. La famille des applications orientées "Routage" .....	8
2.3.3. Applications orientées "Echange" .....	8
3. Le workflow .....	9
3.1. Définition.....	10
3.2 Notion de processus .....	10
3.3. Les trois R du workflow.....	10
3.3.1. Les routes .....	11
3.3.2. Les règles .....	11
3.3.3. Les rôles .....	11
3.4. Typologie du workflow.....	11
3.4.1. Le workflow de production.....	12
3.4.2. Le workflow coopératif (collaboratif).....	12
3.4.3. Le workflow ad-hoc .....	12
3.4.4. Le workflow administratif .....	12
3.5. La workflow management coalition (WFMC) .....	13
3.5.1. Système de gestion de workflow (SGWF) .....	14
3.5.1.1. Modèle de référence d'un système de gestion de workflow .....	14
3.5.2 Les phases de développement d'un SGWF.....	17
3.5.2.1. La phase d'analyse .....	17
3.5.2.2. La phase de construction.....	18
3.5.2.3. La phase d'exécution .....	18
3.6 Implantation de workflow.....	18
3.6.1 Modalisation des processus.....	18
3.7. Avantages des applications du workflow.....	19
4. Conclusion .....	21
<b>Chapitre II. Le langage XML</b> .....	22
1. Introduction.....	22
2. Origine et objectifs de XML .....	22
3. Les règles .....	23
4. Les avantages .....	23
5. Concepts de base.....	24
5.1. Document structuré.....	24
5.2. Document XML .....	25
5.3. Documents bien formés et documents valides.....	26



5.3.1. Documents bien formés .....	26
5.3.2. Documents valide.....	26
5.4. Document XML minimal.....	26
6. Terminologie et syntaxe d'un document XML.....	26
6.1 Le prologue .....	27
6.1.1 La déclaration XML.....	27
6.1.2 Les instructions de traitement .....	28
6.1.3 La déclaration de type de document .....	28
6.1.3.1 La déclaration d'une DTD externe.....	29
6.1.3.2 La déclaration d'une DTD interne .....	30
6.2. L'arbre des éléments XML .....	30
6.2.1 Les éléments.....	31
6.2.1.3 Contenu d'un élément .....	32
6.2.1.4. Les éléments vides .....	32
6.2.2. Les attributs.....	32
6.2.3. Les Commentaires.....	33
7. Conclusion .....	33
Chapitre III. Le cycle de développement de définition.....	34
1. Introduction.....	34
2. La démarche suivi .....	35
3. Spécification des besoins : Etape 1 .....	36
3.1. Les cas d'utilisations .....	36
3.1.1. Les acteurs.....	36
3.1.2. Descriptions des cas d'utilisation.....	36
3.1.3. Diagramme de cas d'utilisation.....	37
3.1.4. Diagramme de séquence .....	39
3.1.5 Diagramme de collaboration.....	41
3.1.6. Diagramme d'activité.....	42
4. Etape 2 : L'analyse.....	44
4.1 Les concepts d'un processus workflow .....	44
4.1.1. Une activité .....	44
4.1.2. Un acteur .....	44
4.1.3. Un rôle.....	45
4.1.4. Une tâche.....	45
4.2. Les composants fonctionnelles d'un processus workflow .....	45
4.3. Les composants organisationnels d'un processus workflow .....	46
4.4. Le diagramme de classe.....	47
4.5. Diagramme d'etats-transition.....	49
5. La conception.....	52
5.1. Conception globale (générale).....	52
5.2. La conception détaillée .....	53
5.2.1. Module 01 : La modélisation graphique .....	53
5.2.2. Module 3 : La génération de définition.....	55
5.2.2.1. Le passage d'UML vers XML .....	56
5.2.2.2. La description d'un processus workflow en XML.....	57
5.2.2.3. La description d'une tâche.....	58
5.2.2.4. La description d'une transition.....	59
5.2.2.5. La description d'un connecteur.....	59
5.2.2.6. La description d'un rôle.....	60
5.2.2.7. La description des ressources .....	61

5.2.2.8. Traitement d'un document XML .....	63
5.2.3. Module 02 : Echange .....	66
5.2.3.1. Connexion de l'application avec les API DOM.....	66
5.2.3.2. Les interfaces DOM .....	66
6. L'implémentation.....	69
6.1. Choix du langage de programmation.....	69
6.2. Création d'un fichier XML .....	69
6.3. Les fonctions de l'outil de définition .....	71
6.3.1. La modélisation graphique : Etape 1.....	71
6.3.1.1. L'espace de travail .....	72
6.3.1.2. La barre à outil .....	73
6.3.1.3. L'exploiteur de projet.....	74
6.3.2. La génération de la définition .....	74
6.4. Le rapport du processus workflow.....	76
7. Test et validation.....	76
7.1. Le test.....	76
7.1.1. Exemple d'un cas.....	77
7.1.2. Le modèle de processus .....	77
7.1.3. La spécification du modèle .....	78
7.2. La validation des cas d'utilisation.....	78
7.2.1. Validation du cas définition des processus workflow.....	79
7.2.2. Validation du cas définition des acteurs .....	79
7.2.2.1. Validation de la définition des rôles .....	80
7.2.3. Définition des activités.....	80
7.2.4. Définition des tâches.....	81
7.2.5. Définition des transitions .....	82
7.2.5. Définition des ressources .....	83
8. Conclusion .....	84
<b>Conclusion générale .....</b>	<b>85</b>
<b>Bibliographies.....</b>	<b>86</b>



## Introduction générale

L'entreprise, depuis plusieurs générations est organisée de manière fonctionnelle, c'est à dire qu'il existe des services relativement cloisonnés comme les bureaux d'étude, les méthodes, l'ordonnancement, les achats, la production, les commerciaux, l'administratif,....etc. Même si ces fonctions prises individuellement seront toujours, elles ne s'auto-suffisent plus [Dan, 00]. L'entreprise cherche plus d'innovation, d'efficience et de réactivité. Pour ce faire il est nécessaire de décroisonner cette organisation en déterminant les interactions (processus) entre ces services pour des objectifs propres à chaque entreprise en fonction de sa nature des services, et pour apporter plus de valeur ajoutée à ses clients. En étant assez simpliste, nous pouvons dire que de ce constat est née l'organisation par processus de l'entreprise.

Toutes les applications que l'on met en oeuvre dans les intranets collaboratifs font partie d'un domaine de l'informatique qui existe maintenant depuis plus d'une dizaine d'années sous le vocable de *groupware* (parfois traduit par collecticiels en français). Les outils appartenant à cette grande famille de solutions permettent d'améliorer grandement le travail coopératif entre les différents membres d'une même équipe où entre les membres d'équipes différentes ayant des informations à partager, parmi ces solutions nous nous intéressons sur la technique workflow qui sert à automatiser les processus métier.

Le travail que nous présentons rentre dans un travail de projet de recherche au niveau de LRDSI (laboratoire de recherche pour le développement des systèmes informatisés) dans l'université de blida. Ce travail consiste à développ  un outil de d finition d'un processus workflow a base d'un langage XML.

Notre rôle dans ce projet est de réaliser la première phase qui consiste à développer un outil de définition permettant une représentation graphique d'un processus workflow par un expert avant la définition textuelle de ce processus, et la définition générée est considérée comme un cas de la base de cas.

## **1. Problématique**

Le travail présenté consiste à résoudre les problèmes suivants :

- La modélisation graphique d'un processus workflow : il faut que l'expert à une bonne connaissance informatique.
- L'automatisation des processus sert a utilisé un outil puissant, il s'agit du workflow.
- Il n'existe pas une notation standard se qui produit un problème d'échange des processus.
- La difficulté de l'interprétation d'un processus a cause de les notation existant qui sont pas formelles.

## **2. Les objectifs**

Le but de notre travail est de réaliser un outil de définition permettant de résoudre les problèmes décrits précédemment.

Notre projet sert a développé un outil de génération qui est utilisé pour générer la représentation du processus sous une forme graphique. Ce projet vise à étudier et à réaliser un outil qui permet la modélisation graphique d'un processus workflow, et générer la définition associée. La génération du contenu de processus sera réalisée à l'aide d'un fichier texte : schéma (XML). On formalisera les aspects du Workflow comme les rôle, les tâches et leurs organisation, les données utilisées, ...etc. à l'aide du langage XML (Extensible Markup Language).

L'outil qu'on va développé permet aux utilisateur de :

- L'utilisation des objets graphique correspondre aux éléments métier.
- Construire des nouveau processus workflow, modifier et consulter des processus workflow existant.
- La génération du représentation textuelle après la modélisation graphique d'un processus workflow permettre au moteur de workflow de l'exécuter.
- La facilité de modéliser un processus workflow.

Nous avons organisé notre mémoire comme suit :

Dans le premier chapitre nous présentons la technologie groupware et un de ces type d'application le workflow on précisons leur concepts et leur types.

Dans le deuxième chapitre nous étudions des concepts fondamentaux et la grammaire du langage XML qui sera utilisé pour l'interprétation des processus workflow.

Le dernier chapitre sera consacré pour la représentation de notre travail, dans ce chapitre nous avons décrit la démarche suivit depuis l'analyse des besoin jusqu'à l'implémentation et test, en détaillons chaque étape de cycle de vie de la méthode en cascade, et utilisons le langage de modélisation UML.

A la fin de cette mémoire nous avons faire un conclusion résume l'importance et les perspectives de notre travail.

# CHAPITRE I.

## GROUPWARE ET WORKFLOW

### 1. Introduction

Le rôle du système d'information dans les entreprises est en train de changer. La décentralisation de puissance et de mémoire induite par la dispersion des micro-ordinateurs les oriente vers de nouvelles architectures et vers un partage différent des responsabilités entre utilisateurs du système d'information et informaticiens.

Dans l'une de ces variables de changement qui s'imposent aujourd'hui se trouvent les nouvelles technologies de l'information et de la communication. Nous nous arrêterons sur la technologie du *groupware* de façon générale, et à celle du *workflow* en particulier.

Nous avons organisé ce chapitre comme suit :

Dans la première section nous présentons la technologie du groupware, la section suivante est consacrée à la représentation du technique workflow, à la fin nous illustrons cette présentation par les avantages de cette technique.

### 2. Le groupware

Le groupware est souvent considéré comme synonyme de travail coopératif assisté par ordinateur (*Computer –Supported Cooperative Work* ou *CSWC* en anglais) [Marek, 99].

Le groupware fait partie de ces nouvelles technologies de l'information, tout comme internet. Bien que sa première définition précise remonte à ...1978. Peter et Trudy Johnson-Lenz<sup>4</sup> le définissaient ainsi:

*"Groupware is intentional group processes and procedures to achieve specific*

*purpose plus software tools designed to support and facilitate the group's work".*  
[Frey, 99].

Cette définition originale montre les trois dimensions que devrait couvrir tout groupware: la dimension humaine, organisationnelle et la dimension technologique.

Le groupware est une technologie qui recouvre des domaines aussi vastes que la coopération, l'interaction homme-machine et l'interaction interpersonnelle via des techniques numériques. Ceci avec l'objectif d'améliorer et de transformer les entreprises de manière radicale. En effet, l'adoption des outils de groupware est basée sur la communication et le partage du savoir, ce qui nécessite une organisation sociale aplatée, démocratique et coopérative [Fred, 00].

## 2.1. Les concepts

Le Groupware est un concept qui porte sur la distribution et le partage d'information, l'échange d'idée, le partage de connaissances pour un travail commun et la synchronisation des tâches de travail. Il utilise les logiciels basés sur une architecture réseau.

Cette technique couvre les fonctions suivantes :

**2.1.1. La communication :** Permettre aux participants d'échanger des informations, par l'écrit, par la parole ou le geste. Parmi les outils qui permettent la communication on trouve :

- **La messagerie électronique :** Qui représente un outil puissant et indispensable pour informatiser un groupe de travail, il forme le cœur de tout système groupware [Frey, 99]. Les messages sont conservés et forment une base de données permanentes de la connaissance de l'entreprise, son carnet d'adresse est utilisé pour organiser les réunions de groupe, router des documents, accorder les droits d'accès aux documents.

- **Les forums électroniques :** Il permet de partager des connaissances, donner une même base d'information au début de travail en commun aux participants. Ceux-ci facilitent les prises de décisions en groupe et la création d'idées nouvelles.

### 2.1.2. La collaboration

- **Les base de données partagées :** Utilisent le Système « pull », ou chacun va y

chercher l'information dont il a besoin au moment désire. Mais comment savoir qu'une information a été modifiée ou non ? car ce qui est important, c'est de posséder la dernière mise à jour de cette information, et d'être averti que l'information que l'on détenait déjà, vent d'être reliée a de nouveaux documents [Frey,99].

### 2.1.3. La coordination

Les Workflow Permettent le routage des documents entre plusieurs personnes. Ils utilisent la messagerie pour notifier tout changement et une base de donnée partagée pour stoker les éléments avec leur état d'avancement.

## 2.2. Les règles d'or du groupware

Pour situer l'importance d'un système groupware au sein de l'entreprise, il existe 3 règles :

**2.2.1. Etre conscient que le groupware n'est pas qu'un produit logiciel :** Il entraîne des changements sur :

- **Changements sur le management :** Pour que le groupware doit être productif, il faudra apprendre aux participant à travailler en équipe et de partager leurs connaissances et leurs informations avec d'autre.
- **Changements sur l'organisation :** D'une part, parce qu'il faudra apprendre aux employés à travailler en équipe et à accepter de partager leurs connaissances et leurs informations avec d'autres. Si les individus ne le font pas, le groupware ne sert à rien, et pourra même être contre-productif.

D'autre part, les centres de responsabilité sont déplacés sur les groupes, les frontières internes sont supprimées, et la hiérarchie aplanie. En somme, suppression de la centralisation du contrôle. De ce fait, un changement dans la manière de manager et d'accompagner les hommes est nécessaire [Frey, 99].

- **Changement sur l'informatique :** L'architecture du groupware doit s'appuyer sur le système d'information de l'entreprise. Cependant, si les systèmes de communication ne sont pas adaptés pour supporter un système groupware, il faudra penser à changer l'architecture réseau de l'entreprise. De plus, il se peut que

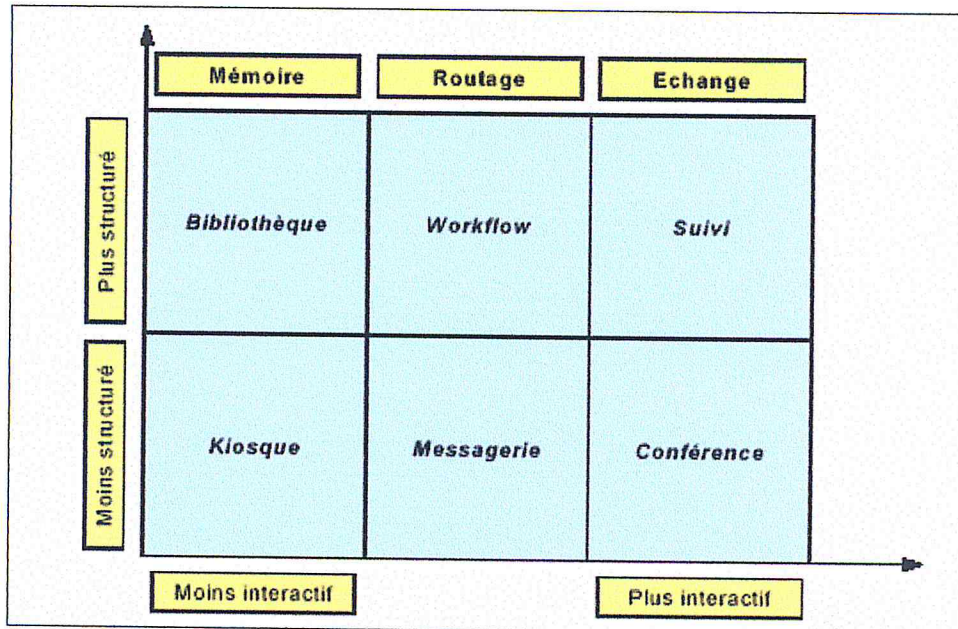


Tableau I.1 : Classification des outils groupware [Fred, 00].

Ce tableau fait ressortir trois grandes familles d'applications :

- La famille des applications orientées "Mémoire"
- La famille des applications orientées "Routage"
- La famille des applications orientées "Echange"

### 2.3.1. La famille des applications orientées "Mémoire"

Le but principal de ces applications de groupware est de mettre en commun des informations, voire des connaissances, capitalisées par et pour les activités de différents groupes et individus. Cette mise en commun constitue une **mémoire collective partagée**.

La classe générique des applications de groupware "Mémoire" se décompose en deux sous-classes d'applications : "les applications de **bibliothèque**" et celles dites "applications **kiosque**". Les premières facilitent l'acquisition, la conservation organisée et la restitution de la mémoire sémantique du groupe, alors que les secondes s'attachent à sa mémoire volatile.

### 2.3.2. La famille des applications orientées "Routage"

Le but principal de cette famille des applications est d'organiser dans le temps et dans l'espace des flux d'informations suivant des schémas de circulation généralement prédéfinis entre les acteurs. La caractéristique principale de ce type

d'application de groupware est le rôle de transporteur d'information. Chaque application de routage assure un transport, en temps réel ou différé, d'objets électroniques d'un individu à l'autre, d'un individu à une application ou d'une application à une autre. La transmission des informations s'effectue ici sur le mode asynchrone.

Deux sous-classes se distinguent des applications de routage. Celles dites "applications de **workflow**" et celles des "applications de **messagerie**". Une application de messagerie est en quelque sorte la première et la plus simple des applications de workflow. En effet certains produits dits de workflow ne sont qu'un moteur de messagerie couplé d'un éditeur de formulaires électronique.

### 2.3.3. Applications orientées "Echange"

Cette classe d'application est sans aucun doute celle qui justifie le plus l'existence du groupware et de ses applications dans l'entreprise. En effet, les processus de travail en groupe sont dominés par des échanges intenses. Dans les environnements complexes où la division du travail fait largement appel aux mécanismes de coordination, les besoins de communication et d'échange entre les individus où entre les groupes sont fondamentaux. Ces échanges peuvent être plus ou moins structurés selon les besoins de coordination ou de coopération dominant.

Le soutien aux échanges par la sous-classe d'applications dites "**applications de suivi**". Des exemples types de ce genre d'application sont l'agenda et le planning de groupe. Ils permettent d'optimiser la gestion du temps individuel et collectif d'une ou de plusieurs équipes, de gérer les tâches quotidiennes personnelles ou déléguées et les étapes d'un projet.

D'autres interactions plus ouvertes et moins structurées déterminent la sous-classe d'applications dites "**applications de conférence**". Celles-ci peuvent se présenter sous la forme de conférence électronique (forum de discussion) qui permet des discussions entre plusieurs personnes intervenant à des moments différents et décalés par rapport à l'unité de temps d'une réunion traditionnelle en face à face.

## 3. Le workflow

Les logiciels de workflow servent à assister, automatiser ou contrôler le traitement des tâches. Ils jouent un rôle central dans la restructuration des processus de travail et l'amélioration de la qualité des transactions commerciales.



Le workflow cherche à automatiser les règles formelles en vue de restructurer les procédures métier de l'entreprise, le groupware essaie de faciliter les interactions informelles entre les groupes en renforçant les aspects communication, coordination et coopération du travail en équipe [Mark, 98].

### 3.1. Définition

Le workflow désigne tout simplement l'automatisation des processus que nous utilisons chaque jour pour faire notre métier. Une application de workflow automatise la séquence des actions, activités ou tâches que nécessite un processus de travail. Elle suit aussi l'état de chacune des instances du processus et gère le processus lui-même [Marek, 98].

### 3.2. Notion de processus métier (Entreprise)

Lorsque les activités sont liées au sein d'un réseau, il en résulte une définition de processus qui intègre la schématique du workflow et les liens qui relient les activités. Le *Workflow Management Coalition* (WMC) définit les activités d'un processus comme un *réseau d'activités* [Marek, 98].

Le processus relie donc des activités multiples effectuées par des acteurs différents utilisant des outils variés et accédant à des sources d'informations diverses.

Formaliser un processus conduit à l'équiper d'un " workflow ", c'est-à-dire d'une documentation explicite des tâches et de leurs relations :

- Préciser les interfaces nécessaires à chaque activité : on regroupe sur le même écran les plages de consultation et de saisie nécessaires à une activité, ce qui évite à l'utilisateur la connexion à d'autres applications ainsi que la navigation dans des codes et touches de fonction diverses.
- Programmer les tables d'adressage permettant de router automatiquement les messages à l'issue d'une tâche, lorsque l'utilisateur tape sur la touche " valider " qui marque la fin de son travail, il n'a pas à chercher à qui envoyer le résultat .

### 3.3. Les trois R du workflow

Le workflow est un concept qui associe des technologies et outils capable d'acheminer automatiquement des événements ou des utilisateurs.

L'automatisation d'un workflow nécessite sa représentation électronique.

### 3.3.1. Les routes

Les routes déterminent des chaînes d'activités plus ou moins complexes (parallélisme, séquence ment...) qui forment des cheminements le long des quels se ce déplacent les objets de gestion [Levan, 99b].

Le routage des documents, des informations ou des taches, représente la première grande fonction du workflow. La route est le chemin suivi par l'ensemble des objets constituant le workflow. Ces routes sont linéaires, circulaires ou parallèles. La route peut être prédéfinie préalablement ou bien choisi au moment de l'exécution.

### 3.3.2. Les règles

Les règles regroupent les informations concernant les tâches à réaliser pour accomplir une activité [levan, 99b]

Les règles déterminent la route à suivre, permettront d'affecter les taches aux participants, ainsi que les définir les conditions auxquelles doit satisfaire le workflow pour passer à l'étape suivante et le traitement des exceptions.

### 3.3.3. Les rôles

Après la définition de la route à suivre suivant certaines règles, il reste à gérer les personnes qui vont accomplir les taches et qui vont communiquer entre elles. Les workflow gèrent des rôles, c'est-à-dire des fonctions investies d'une mission dans l'accomplissement d'un processus, et détenant des responsabilités définies pour l'exécution de tache.

## 3.4. Typologie du workflow

Quoique les workflow mettent en œuvre les mêmes fonctionnalités mais ils ont des architectures techniques différentes en fonction de leurs besoins dans l'entreprise.

Il existe plusieurs typologies d'application workflow, toutes utiles pour comprendre ce qu'est le workflow et pour disposer d'élément de réflexion et de décision par rapport au besoin et problématique d'une entreprise.

Le principale critère de différenciation repose sur la mission fondamentale de l'application workflow : l'application workflow s'attache à automatiser des procédures de production dont il est possible de définir à l'avance toutes les règles.

### 3.4.1. Le workflow de production

Qui correspond à la gestion des processus de base de l'entreprise. Les procédures supportent peu de changements dans le temps, et les transactions sont répétitives. On peut y trouver par exemple la production de contrats d'assurance, la gestion de litiges, la gestion de réclamations clients [MAY].

Un workflow de production est caractérisé par un cadre procédural formel qui s'applique à toutes les activités et à tous les rôles impliqués dans l'accomplissement d'un processus en question.

### 3.4.2. Le workflow coopératif (collaboratif)

Gérant des procédures évoluant assez fréquemment, et liées à un groupe de travail restreint dans l'entreprise [MAY].

Les workflow collaboratifs concentrent sur le travail d'équipe en vue d'atteindre des objectifs communs. L'utilisation des workflow collaboratifs dans le but de faciliter les communications intergroupes est actuellement reconnue comme un élément vital dans le succès d'une entreprise, tout comme l'usage d'Internet et du Word Wide Web.

### 3.4.3. Le workflow ad-hoc

Ce type de workflow est utilisé pour la gestion des procédures non déterminées, ou mouvantes [MAY].

Les workflow ad-hoc automatisent des procédures d'exception, c'est-à-dire occasionnelles voire uniques. Ces processus pourraient dans certains cas représenter des enjeux critiques pour la performance de l'entreprise mais ils sont le plus souvent liés à des routines administratives.

### 3.4.4. Le workflow administratif

Qui correspond à tout ce qui est routage de formulaires, basé en général sur une infrastructure de messagerie [MAY].

Le workflow administratif correspond alors à des processus de routages de formulaires simple. Il s'agit alors d'automatiser la manipulation de formulaires électroniques, en remplacement des imprimés traditionnels, entre plusieurs personnes intervenant sur leur traitement et suivant des procédures pré définies. Les

applications de workflow administratif associent généralement un gestionnaire de formulaires (conception et utilisation) à un moteur de messagerie. Par exemple la gestion d'autorisation de dépenses, la gestion de frais de mission [Frey, 99].

La matrice suivante résume de manière graphique la typologie fonctionnelle des applications de workflow :

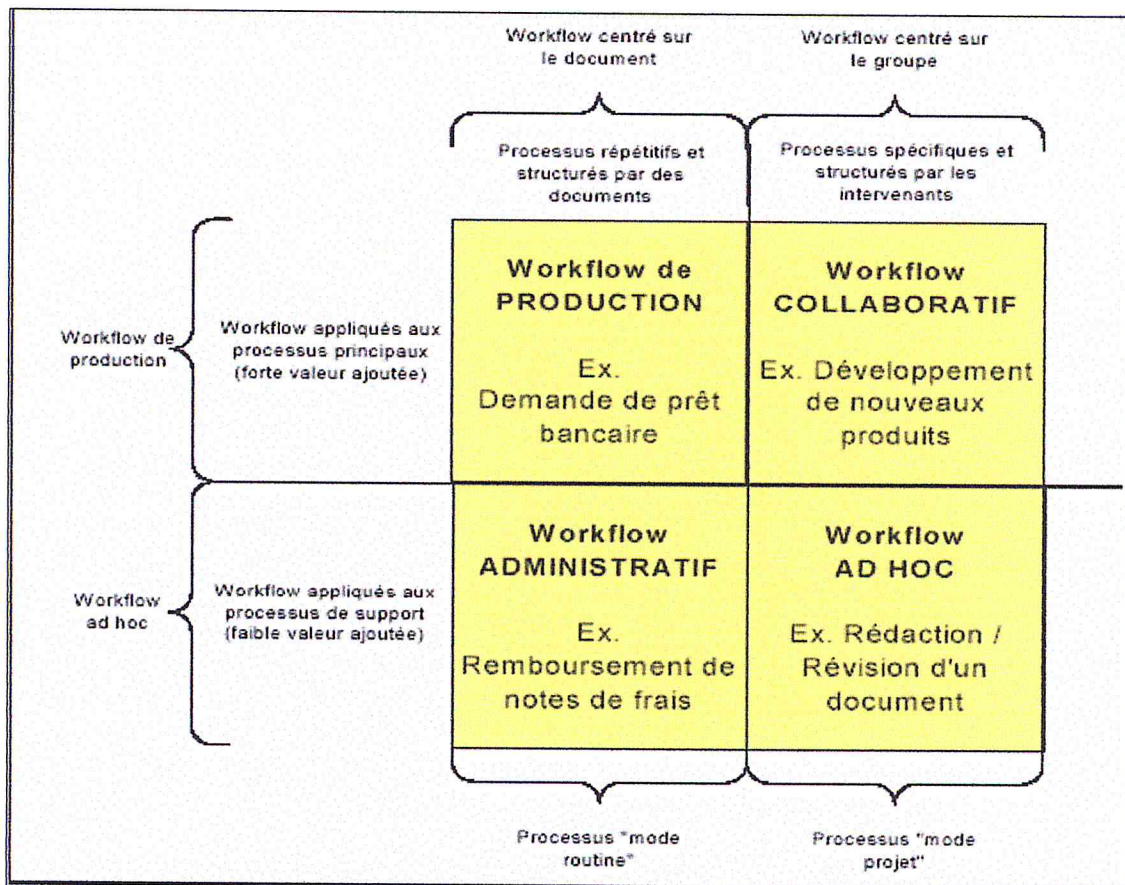


Figure I.1 : Typologie fonctionnelle des applications de workflow [Levan, a99].

### 3.5. La workflow management coalition (WFMC)

Fondée en 1993 [Pont, 00], la Workflow Management Coalition (WfMC) est une organisation internationale dédiée au marché du workflow. La terminologie, les interfaces logicielles, les méthodes et les techniques d'implémentation variaient d'un produit à l'autre. Si cette disparité des solutions permettait à certains éditeurs de se démarquer sur le marché du workflow, elle décourageait aussi les investissements qui étaient rapidement menacés d'obsolescence. L'absence de standard laisse libre court aux solutions propriétaires qui sont de réelles menaces pour la flexibilité des systèmes d'information. Cela a toujours été un souci pour les responsables informatiques qui veillent sur les données de l'entreprise, mais la menace était

encore plus forte lorsqu'il s'agissait de la connaissance même des processus [Mélissa, 02].

Le rôle de la WfMC est donc fondamental dans la maturation du marché des produits workflow, comme dans la maturation des concepts et des usages du workflow dans les entreprises [Mélissa, 02].

Sa mission est de promouvoir et de développer l'usage du workflow par l'établissement de standards sur la terminologie, la connectivité et l'interopérabilité entre les produits de workflow [Pont].

### **3.5.1. Système de gestion de workflow (SGWF)**

Le système de workflow est un système complet qui sert à définir, gérer et exécuter des procédures (workflow) en exécutant des programmes dont l'ordre d'exécution est prédéfini dans une présentation informatique de la logique de ces procédures [Levan, 99b].

#### **3.5.1.1. Modèle de référence d'un système de gestion de workflow**

La Coalition a proposé un cadre de référence pour les standards workflow. Ce cadre couvre cinq catégories de standards relatifs à l'interopérabilité et la connectivité des outils, autorisant la cohabitation de plusieurs solutions dans un environnement donné.

Ce modèle de référence exprime à lui seul tout le travail de la Coalition. Celui-ci est fondé sur une hypothèse : tous les systèmes de gestion de workflow reposent sur les mêmes composantes génériques qui interagissent selon diverses modalités.

Pour obtenir l'interopérabilité entre plusieurs produits de workflow, il faut définir des standards d'interface et d'échange de données. Le modèle de référence de la Coalition définit ainsi un cadre de référence pour la définition et l'implémentation de ces standards. Mais comme un système workflow est constitué d'un regroupement de plusieurs technologies, la Coalition a défini cinq composantes faisant elles-mêmes l'objet de standards [Mélissa, 02].

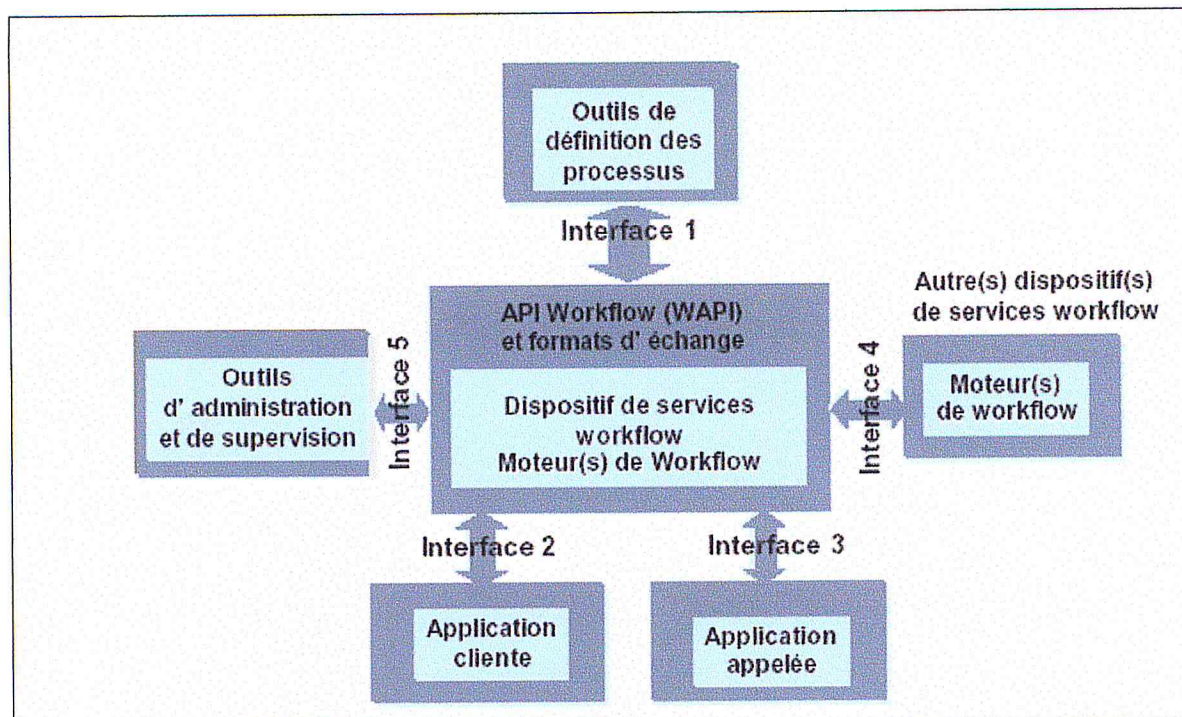


Figure I.2 : Modèle de référence de la WfMC [Van, 02].

Le modèle de référence de la Coalition identifie cinq composants et interfaces :

**1) L'outil de définition de processus (interface 1) :** De nombreux outils peuvent servir à l'analyse, à la modélisation et à la description de processus d'entreprise. Le modèle de référence n'est pas particulièrement concerné avec la nature particulière de tels outils et généralement ceux-ci sont conçus en fonction du produit de workflow avec lequel ils sont couplés. L'interface proposée par la Coalition vise à garantir le maximum de souplesse et d'ouverture dans ce domaine.

Avec la WfMC, d'autres acteurs de l'industrie informatique travaillent dans ce domaine de la modélisation de processus et des outils CASE.

**2) Le moteur de services workflow (serveur workflow) :** Le moteur de services workflow correspond à un environnement run-time capable d'exécuter un ou plusieurs workflow. Cet environnement peut impliquer un ou plusieurs moteurs de workflow, c'est-à-dire plusieurs produits workflow différents. Le moteur de services workflow est distinct des applications et des outils orientés utilisateurs mis en œuvre dans l'accomplissement des tâches et des activités du processus. Une grande variété de standards applicatifs peut cependant être intégrés dans le moteur de services workflow pour fournir un système de gestion de workflow complet. Cette intégration prend deux formes:

- L'interface d'applications appelées qui permet au moteur de workflow d'activer une application spécifique pour une activité donnée. Cette interface sera intégrée au module serveur et ne demandera pas d'action particulière de la part de l'utilisateur (par exemple l'appel à une messagerie électronique ou l'exportation de données vers une base de données sur mainframe).

- L'interface du client workflow à travers laquelle le moteur de workflow interagit avec une application workflow cliente chargée d'organiser le travail pour le compte d'un utilisateur.

**3) L'application cliente workflow (interface 2) :** L'application cliente workflow est le module logiciel qui présente les bons de travail à l'utilisateur et peut appeler des applications et des outils logiciels nécessaires à l'accomplissement des tâches.

L'utilisateur rend ensuite la main au moteur de services workflow pour poursuivre le déroulement du processus. Le client workflow peut faire partie intégrante d'un système de gestion de workflow comme il peut être un produit tiers, comme une messagerie par exemple) ou bien encore une application spécifique.

Il est donc indispensable de déterminer des modalités de communication ouvertes entre un moteur de services workflow et un client workflow qui puissent apporter un ensemble de fonctions de connexion aux services workflow et d'exécution des tâches et activités du processus.

**4) L'application appelée par le workflow (interface 3) :** Les systèmes de gestion de workflow doivent communiquer avec toutes les applications externes nécessaires à l'accomplissement des tâches : appeler un service de messagerie X.400, envoyer une télécopie, utiliser des fonctions de gestion de documents, des outils bureautiques, des applications de production, etc.

La Coalition attache beaucoup d'importance au développement de standards relatifs à l'appel de telles applications en fournissant une interface dédiée à cette liaison entre applications. Par ailleurs il a été démontré qu'il serait intéressant de mettre à disposition un jeu d'API qui permettraient aux développeurs de construire des applications intégrant des fonctionnalités workflow grâce à l'appel direct de fonctions propres au moteur de workflow.

**5) Les autres moteurs de services workflow (interface 4) :** Un des objectifs fondamentaux de la Coalition est de définir des standards permettant à différents

systèmes de gestion de workflow conçus et produits par différents éditeurs de travailler ensemble sur les mêmes bons de travail.

Les produits workflow sont très variés dans leur nature, depuis le simple outil de routage de données (workflow ad hoc) à la gestion de processus complexes et évolutifs (workflow coopératif). Tous ces produits présentent des avantages par rapport à des besoins particuliers. Les standards ne visent pas à appauvrir la richesse des solutions workflow proposées par les éditeurs.

Ces standards d'interopérabilité peuvent agir à différents niveaux : du simple transition de tâches d'un produit workflow à l'autre jusqu'à l'échange intégral de définitions de processus avec des données d'ordonnement de processus.

**6) L'outil d'administration et de pilotage du système workflow (interface 5) :** Il s'agit de définir un standard d'interface permettant à un outil d'administration et de pilotage de travailler avec n'importe quel moteur de services workflow. Tout d'abord cela permettra d'obtenir une vision complète de l'état d'un workflow cheminant à travers toute une organisation, indépendamment des systèmes workflow mis en œuvre.

Ensuite, cela permettra de choisir le meilleur outil d'administration et de pilotage en fonction des ses besoins et objectifs.

### **3.5.2 Les phases de développement d'un SGWF**

Les différentes phases par lesquelles doit passer un projet de WorkFlow. Elles sont au nombre de trois [MAY]:

#### **3.5.2.1. La phase d'analyse**

C'est la phase de modélisation des procédures, sous la responsabilité des organisateurs de l'entreprise. Rares sont les produits de WorkFlow qui intègrent cette dimension [MAY]. Elle est issue plutôt des méthodes de BPR (Business Process Reengineering), dont l'objectif est de remettre à plat tous les processus de l'entreprise pour en rebâtir de nouveaux, plus efficaces et mieux adaptés à la compétitivité ambiante. A ces phases de BPR sont toujours associés les outils informatiques. C'est dans ce cadre que ce sont essentiellement développés les outils de workflow, puisque parfaitement optimisés pour le traitement automatisé des procédures de l'entreprise [MAY].



### 3.5.2.2. La phase de construction

Elle consiste, à partir des modélisations de processus issus de la phase précédente, à formaliser les procédures résultantes au sein d'un outil informatique, et à définir l'ensemble des conditions nécessaires à son bon fonctionnement, et à son intégration dans l'informatique existante. Tous les produits de WorkFlow possèdent un module gérant cette phase, mais de manière plus ou moins évoluée. Les produits complets doivent offrir un mode de représentation graphique des procédures [MAY].

### 3.5.2.3. La phase d'exécution

C'est la phase pendant laquelle les procédures sont exécutées et les tâches traitées, c'est également pendant cette phase que les statistiques, fondamentales pour le suivi de tout processus, sont générées. Des outils d'administration doivent également exister afin de pouvoir intervenir à tout moment sur les procédures elles-mêmes en cas de problème. Bien entendu tous les produits de WorkFlow intègrent ce module [MAY].

## 3.6 Implantation de workflow

Le workflow doit être considéré comme un outil, que ce soit pour améliorer les infrastructures existantes, ou pour restructurer des processus métiers. Parce qu'il n'est qu'un outil, la technologie workflow n'a pas de solution aux problèmes liés au métier de l'entreprise.

L'introduction du workflow dans une organisation nécessite une phase de conception, tout comme pour le groupware. L'accent doit être mis sur les conditions de l'intégration du workflow dans le système d'information de l'entreprise, et moins sur la technique. Encore une fois, le système "vivant" doit être respecté en tenant compte de tous les aspects de l'entreprise.

Comme tout outil, le workflow ne traite que ce qu'il reçoit. On a trop souvent tendance à l'oublier. Donc si les données en entrées sont mauvaises ou inadaptées, le résultat de leur traitement ne sera pas bon.

### 3.6.1 Modélisation des processus

Modéliser un processus, c'est décrire la succession des tâches qui concourent à une production de valeur ajoutée : ce que fait chaque acteur, les données qu'il manipule, les traitements qu'il ordonne, les délais dans lesquels son travail doit être

exécuté, le routage de ses messages vers d'autres acteurs, les compteurs qui permettent au responsable du processus de contrôler la qualité.

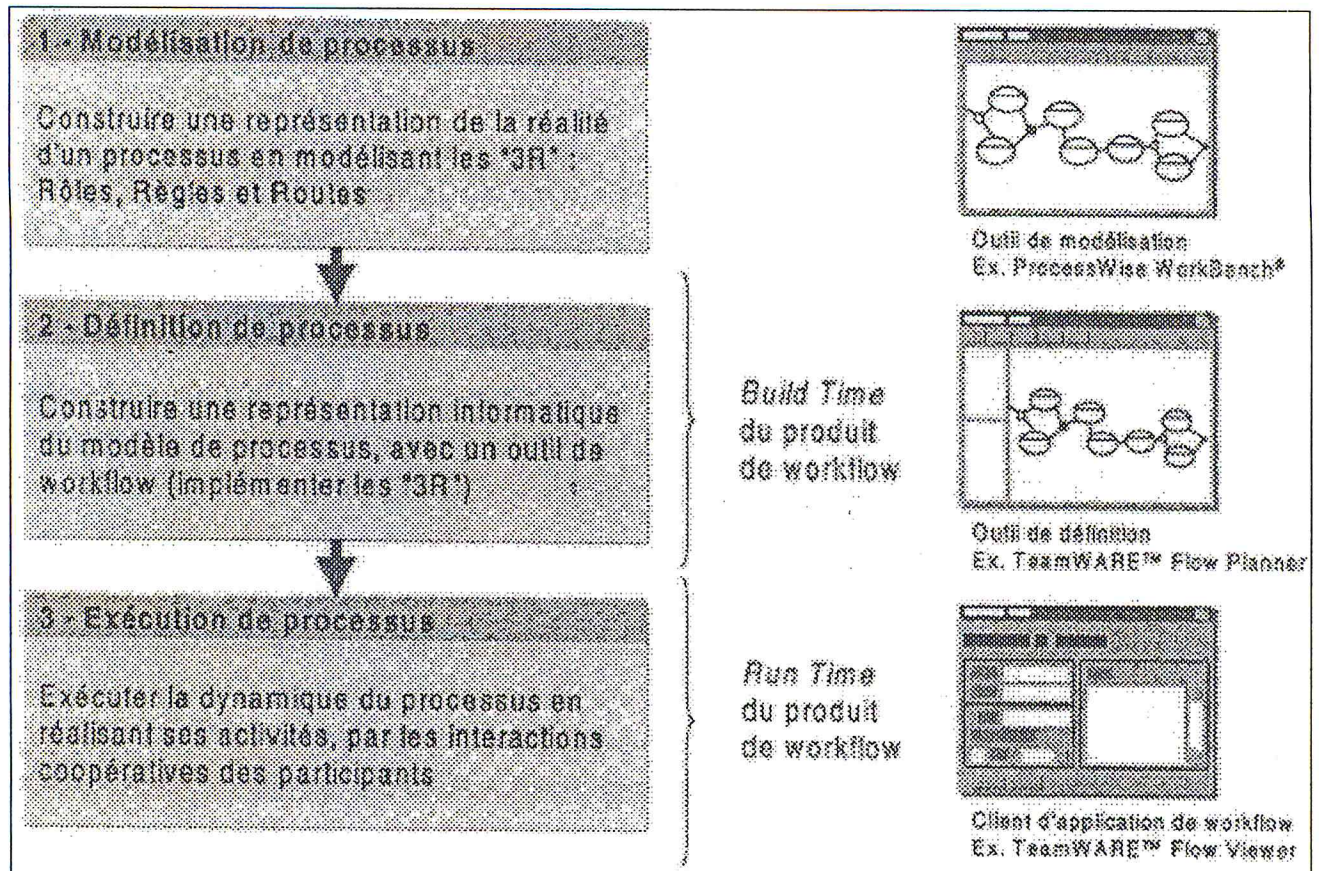


Figure I.3 : Etapes d'exécution d'un processus [Frey, 99].

De manière générale, quels que soit les produits workflow utilisés, que le référentiel méthodologique soit distinct ou non du référentiel technique, l'implantation du workflow commence par modéliser les trois R du workflow:

- Les rôles, qui doivent être définis indépendamment des personnes
- Les règles relatives aux conditions d'exécution des activités, tout en tenant compte de l'intégration des applications externes au workflow (traitement de texte, tableur,...).
- Les routes assurant la liaison entre les activités et leurs règles, et les acteurs dans leurs rôles.

### 3.7. Avantages des applications du workflow

Les différents avantages et bénéfices rencontrés lors de l'introduction d'un système de workflow peuvent être de deux natures. Soit ils sont mesurables donc

tangibles, soit ils sont moins "palpables", mais contribuent tout autant à l'amélioration significative de la qualité du travail effectué.

Du côté des gains tangibles nous retrouvons les éléments suivants :

- **Réduction des coûts opérationnels**

Les organisations utilisant des systèmes de workflow constatent une diminution des coûts de transaction. L'exemple d'une banque ayant mis en place un système de workflow pour gérer ses demandes de prêts bancaires, relève une diminution de ces coûts de plus de 33%.

- **Amélioration de la productivité**

Les opérations routinières et répétitives peuvent être automatisées réduisant ainsi significativement le temps d'exécution du processus. De plus, le travail peut être effectué 24h/24, ceci étant un facteur vital pour les multinationales et les entreprises effectuant des transactions commerciales par le biais d'Internet.

- **Processus plus rapides**

Deux facteurs expliquent le gain de temps des processus gérés par des systèmes de workflow. Le premier, nous l'avons vu plus haut est dû à l'automatisation des opérations routinières. Le deuxième concerne les activités "manuelles" ou nécessitant une intervention humaine. Celles-ci, peuvent souvent être effectuées parallèlement (en tous cas pour une partie d'entre elles). Le workflow permet dans ce cas, grâce à une coordination efficace et une attribution des activités à plusieurs acteurs, de faire progresser le processus nettement plus rapidement.

Les gains intangibles sont les suivants :

- **Service amélioré**

Grâce à la rapidité de gestion des demandes de la clientèle ainsi qu'à une meilleure information sur l'état d'avancement de celles-ci, le service rendu aux clients s'en trouve amélioré.

- **Amélioration des conditions de travail des employés**

Les tâches répétitives et peu gratifiantes peuvent être automatisées, libérant de cette façon le personnel pour des activités plus intéressantes.



- **Facilitation du changement**

Les entreprises peuvent constamment, grâce aux systèmes de workflow, redéfinir et automatiser leurs processus.

- **Augmentation de la qualité**

Suite aux automatisations des tâches répétitives, ainsi qu'à une meilleure coordination et compréhension du travail, les erreurs sont plus rares.

- **Communication facilitée**

Grâce aux informations disponibles concernant les tâches à effectuer et l'état d'avancement des processus, la communication et la transparence du travail sont améliorés.

- **Aide à la prise de décision**

Etant informé du déroulement des processus et des activités, il est plus facile de prendre les bonnes décisions.

- **Amélioration du planning**

Les informations disponibles concernant l'organisation, son business et ses processus améliorent les facultés de planning.

- **Communications inter-entreprise**

La gestion de processus inter entreprise augmente considérablement la productivité et la transparence du marché.

#### **4. Conclusion**

Dans ce chapitre nous avons présenté la technologie groupware, et une de ces applications, le workflow.

Afin de réaliser un outil de définition de workflow, nous devons définir un formalisme qui nous permet de définir le processus à exécuter. Le prochain chapitre représente le formalisme utiliser.

# **CHAPITRE II :**

## **LE LANGAGE XML**

### **1. Introduction**

Le Langage de balisage extensible (en anglais Extensible Markup Language ,abrégé XML) décrit une classe d'objets de données appelés documents XML et décrit partiellement le comportement des programmes qui les traitent [Patr, 98].

Dans ce chapitre nous présentons XML, le langage que nous utilisons pour le développement de l'outil de définition de workflow, pour cela nous présentons dans la première section l'origine et les objectifs de XML, ensuite les règles et avantages de XML. La seconde partie consiste à écrire les concepts de base et la syntaxe d'un document XML.

### **2. Origine et objectifs de XML**

XML a été développé par un groupe de travail (XML Working Group initialement connu sous le nom de comité d'examen éditorial SGML, SGML Editorial Review Board) constitué sous les auspices du Consortium du World Wide Web (W3C) en 1996 [Patr, 98]. Le GT était présidé par Jon Bosak de Sun Microsystems avec la participation active d'un groupe d'intérêt XML (auparavant connu sous le nom de groupe de travail de SGML (SGML Working Group)) également organisé par le W3C. Les objectifs de conception de XML sont les suivants :

1. XML devrait pouvoir être utilisé sans difficulté sur Internet.
2. XML devrait soutenir une grande variété d'applications.
3. XML devra être compatible avec SGML.
4. Il devrait être facile à écrire des programmes traitant les documents XML.

5. Le nombre d'options dans XML doit être réduit au minimum, idéalement à aucune.
6. Les documents XML devraient être lisibles par l'homme et raisonnablement clairs.
7. La conception de XML devrait être préparée rapidement.
8. La conception de XML sera formelle et concise.
9. Il devrait être facile de créer des documents XML.
10. La concision dans le balisage de XML est de peu d'importance.

### 3. Les règles

XML impose des règles de syntaxe très spécifiques par rapport aux autres langages de balisage:

- Il permet de définir ses propres balises et ses propres attributs. Il est donc plus flexible que HTML qui, lui, ne possède qu'un nombre limité de balise.
- Un document XML peut être validé par des règles strictes, contenues par des DTD (*Document type définition*) ou des Schéma, décrivant sa structure et la hiérarchie de ses données.
- Les informations ainsi que le traitement de la mise en forme sont rigoureusement séparés de la structure du document XML.
- XML est un format standardisé ouvert ne nécessitant aucune licence, intégralement basé texte et qui peut être associé à n'importe quel jeu de caractère.
- XML est un document portable, il peut être lu sur n'importe quelle plate forme car c'est du texte et n'importe quel outil pouvant lire un fichier texte peut lire un document XML.
- De plus en plus d'application utilise le format XML ; C'est le cas de certains SGBD (Système de Gestion de Base de Données) mais aussi d'outils de bureautique comme Microsoft Office 2003 ou Sun Open Office. XML est également au cœur de la nouvelle plate-forme de développement de Microsoft.Net.

### 4. Les avantages

Nous présentons Les principaux avantages de XML [CCM]:

- **La lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML.
- **Une structure arborescente** : permettant de modéliser la majorité des problèmes informatiques.
- **Universalité et portabilité** : les différents jeux de caractères sont pris en compte.
- **Déployable** : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme HTTP.
- **Intégrabilité** : un document XML est utilisable par toute application pour vue d'un parseur (c'est-à-dire un logiciel permettant d'analyser un code XML).
- **Extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

Le langage XML est appliqué dans plusieurs domaines, nous donnons quelque exemple :

- Les services en ligne et les recherches plus efficaces sur le Web.
- La personnalisation de la relation client.
- La présentation d'un document sur des supports différents.
- Le transfert de données entre différentes bases de données.
- L'intégration des systèmes d'information (interfaçage avec des applications de gestion et avec la bureautique).
- La gestion des documents.
- Le graphisme et le multimédia.

## 5. Concepts de base

### 5.1. Document structuré

Les documents XML est définis par leur nature hiérarchique mais aussi par la normalisation et le mécanisme de validation, peuvent constituer de très bons documents structurés. Les documents structurés sont des documents qui contiennent de l'information à propos de leur structure logique, sémantique et intellectuelle [Sevic, 02]. L'exemple suivant montre la différence entre un document structuré et un document non structuré.

**Exemple 1 :**

```
<?xml version="1.0"?>
<document>
  <p><font size="20pt"><b>Introduction</b></font></p>
  <p><i>George Washington</i>
    <b>Washington</b>
    <b>Washington</b>.</p>
</document>
```

**Exemple 2 :**

```
<?xml version="1.0"?>
<document>
  <section>
    <titre>Introduction</titre>
    <p><personne>George Washington</personne> n'a jamais
      gouverné le <etat>Washington</etat> mais a résidé à
      <ville>Washington</ville>.</p>
  </section>
</document>
```

La différence entre ces documents est très grande. Dans le premier cas, il s'agit d'un document formaté pour une application particulière. Dans le deuxième cas, il s'agit d'une structure d'information.

La norme XML est appropriée pour définir des documents structurés pour deux raisons principales:

- 1) Le modèle de données est hiérarchique, ce qui est également le cas des documents textuels, hypermédias, et de bon nombre de structures de données.
- 2) Les documents XML peuvent être validés selon une grammaire, permettant ainsi de définir, et par le fait même limiter, les structures possibles est donc les traitements à effectuer.

**5.2. Document XML**

XML est un langage de description et d'échange de documents structurés. Comme SGML dont il est issu, il décrit la structure d'un document à l'aide d'un système de balises marquant le début et la fin des éléments qui le compose [Steph, 00].

La norme XML en tant que telle doit être vue comme un outil permettant de définir un langage (on dit alors qu'il s'agit d'un métalangage), permettant de créer des



documents structurés à l'aide de balises [CCM]. Une balise est une chaîne de caractère du type: <balise> possédant éventuellement des attributs.

### 5.3. Un documents bien formés et documents valides

**5.3.1. Un document bien formé :** c'est la condition minimale pour l'exploitation d'une instance de document, définie par un ensemble de règles qui doivent être vérifiées, indépendamment de la DTD ; par exemple, en XML, tout balisage descriptif au sein d'une instance de document doit être constitué d'une balise de début et d'une balise de fin [Sard, 99].

Un document est bien formé s'il obéit aux règles syntaxiques du langage XML. Un tel document sera correctement traité par un programme adéquat, comme un parseur XML. Si le document n'est pas bien formé son traitement provoquera un message d'erreur ou un arrêt de l'application qui le traite.

**5.3.2. Un document valide :** cette notion est commune à XML et SGML, et signifie que le document est bien formé et qu'il se conforme totalement à une DTD [Sard, 99].

Notons que [Steph, 00] :

- Un document valide peut être distribué dans un système d'information sans sa DTD ou sans référence à cette DTD, sous réserve de procéder au remplacement des références à des entités par leur valeur. Le document apparaît dans ce cas comme un document bien formé.

### 5.4. Un document XML minimal

Un document XML minimal contient un élément vide sans attribut, et ne contient pas de prologue.

## 6. Terminologie et syntaxe d'un document XML

Tout document XML se compose :

- D'un prologue, dont la présence est facultative mais conseillée. Il contiendra un certain nombre de déclarations.
- D'un arbre d'éléments. Il forme le contenu du document.
- De commentaires et d'instructions de traitement, dont la présence est facultative.

Ils pourront, moyennant certaines restrictions, apparaître aussi bien dans le prologue que dans l'arbre d'éléments.

## 6.1. Le prologue

Le prologue des documents XML joue trois rôles importants:

- 1) Préciser qu'il s'agit d'un document XML.
- 2) Identifier le jeu de caractères utilisé.
- 3) Identifier la grammaire (DTD) utilisée.

Les trois éléments sont facultatifs, mais il est en général préférable d'inclure la déclaration XML qui contient les deux premiers éléments d'information [Sevig, 02].

### 6.1.1 La déclaration XML

Elle indique au programme qui va traiter le document :

- La version du langage XML utilisée.
- Le codage de caractères utilisé dans le document ; par défaut ISO 10646 mais le programme doit pouvoir traiter aussi l'UTF-8 et l'UTF-16.
- L'existence ou non de déclarations extérieures au document qui doivent être prise en compte pour le traitement de celui-ci.

**Syntaxe:** `<? xml version='1.0' encoding='ISO-8859-1' standalone='yes'?>`

La déclaration est facultative mais fortement conseillée [Steph, 00]. Chacune des trois informations est elle aussi facultative mais si elles apparaissent c'est obligatoirement dans cet ordre.

Les constituants de la balise sont :

<	?xml	version= 'numéro'	encoding= 'encodage'	standalone= 'yes no'	>
Ouverture de la balise	La déclaration du document	Le numéro de version d'XML	Le codage des caractères	spécifie le type de la DTD	Fermeture de la Balise

**Figure II.1 :** Les constituants d'une balise.

### 6.1.2 Les instructions de traitement

Une instruction de traitement est toute chaîne de caractère délimitée par `<? Et ? >`. La seule règle pour le contenu de l'instruction est qu'elle doit commencer par un nom XML valide suivi d'un espace suivi de texte [XMLG, 00].

Leur syntaxe est la suivante :

`<? Cible [Données de l'instruction de traitement]?>`

`<? :` Ouverture de la balise

**Cible :** Un nom XML

Les données de l'instruction de traitement : représentons une chaîne de caractères que l'analyseur XML passera inchangé à l'application identifiée.

`?> :` Fermeture de la Balise.

### 6.1.3 La déclaration de type de document

La déclaration de type de document indique, le cas échéant, la DTD à laquelle se conforme le document. Elle permet aussi de spécifier certaines déclarations propres au document [Steph, 00].

Toute instance de document XML s'appuie sur les règles définies par une DTD. Ces règles sont équivalentes aux règles d'une grammaire formelle, de sorte que toute instance de document peut être représentée sous la forme d'un arbre caractérisant la structure logique, ou structure de balisage, du document. Le mot-clé **DOCTYPE**, placé en tête de la DTD, définit quel est le non-terminal qui est la racine de la grammaire et donc la racine de cet arbre [Sard, 99].

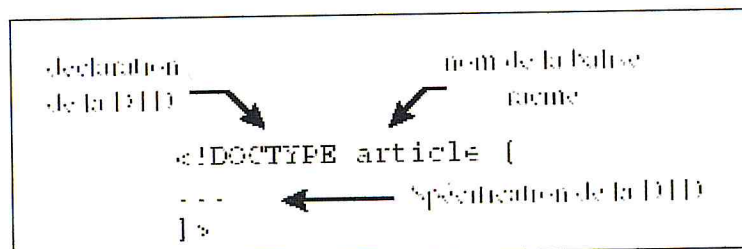


Figure II.2 : La déclaration d'un document XML.

Il existe trois types de déclaration :

- La déclaration d'une DTD externe.
- La déclaration d'une DTD interne.

- La déclaration d'une DTD mixte.

### 6.1.3.1 La déclaration d'une DTD externe

Un document XML peut préciser qu'il respecte une DTD définie dans un document à part, on parle alors de DTD externe. Cette information doit être insérée dans l'entête du document [Girard, 01]:

Pour une référence externe, deux façons existent [Rusty, 04]:

1) l'identifiant **SYSTEM** :

Syntaxe : **<!DOCTYPE Nom-racine SYSTEM "chemin-DTD-externe">**

Les constituants de la balise sont :

**<** : Ouverture de la balise.

**! DOCTYPE** : Instruction XML.

**Nom-racine**: Le nom de l'élément racine du document. C'est un nom XML.

**SYSTEM** : On fait appel à une DTD externe.

**chemin-DTD-externe**: Le chemin de la DTD.

**>** : Fermeture de la Balise.

**Exemple :**

```
<? xml version="1.0" encoding="ISO-8859-1"?>
<! DOCTYPE annuaire SYSTEM "http://abu.cnam.fr/annuaire.dtd">
<annuaire>
...
</annuaire>
```

2) L'identifiant **PUBLIC** :

**Syntaxe :**

**<!DOCTYPE Nom-racine PUBLIC "Non-identifiant-public" " FPI-DTD-externe">**

**<** : Ouverture de la balise

**! DOCTYPE** : Instruction XML

**Nom-racine**: Le nom de l'élément racine du document. C'est un nom XML.

**PUBLIC** : La DTD est publiée et accessible au plus grand nombre d'utilisateurs. C'est une ressource disponible pour tous sur un serveur Web distant.

**Non-identifiant-public:** Le Nom de la DTD (identifiant public).

**FPI-DTD-externe:** Le chemin de la DTD externe.

> : Fermeture de la Balise.

**Exemple :**

```
<! DOCTYPE SEASON PUBLIC
"-//Elliott Rusty Harold//DTD Greetings and salutations//EN"
"http://www.cafeconleche.org/dtds/greeting.dtd">
```

### 6.1.3.2 La déclaration d'une DTD interne

Un document valide peut inclure directement sa DTD dans sa déclaration de type. L'attribut standalone doit prendre la valeur yes, indiquant ainsi que le document est complet et qu'aucune déclaration externe ne doit être recherchée [Steph, 00].

La syntaxe est la suivante [Rusty, 04]:

**<!DOCTYPE Nom-racine [Déclarations]>**

Les constituants de la balise sont :

< : Ouverture de la balise.

! DOCTYPE : Instruction XML.

**Nom-racine :** Le nom de l'élément racine du document. C'est un nom XML.

**Déclarations :** Les règles de la DTD interne.

> : Fermeture de la Balise.

**Exemple :**

```
<? xml version="1.0" standalone="yes" ?>
<!DOCTYPE carte [
    <!ELEMENT carte (#PCDATA)>
]>
<carte>As de pique</carte>
```

## 6.2. L'arbre des éléments XML

Un élément peut contenir un ou plusieurs autres éléments. On parlera d'élément père et d'élément fils. En effet la partie essentielle d'un document XML sera toujours formée d'une hiérarchie d'éléments qui dénote la sémantique de son contenu : c'est l'arbre d'élément [Steph, 00]. Dans un document XML il n'existe qu'un et un seul élément père, l'élément racine, qui contient tous les autres.

**Exemple :**

```
<personne>
  <prenom>Anne</prenom>
  <nom>Martin</nom>
  <physique>
    <yeux>bleu</yeux>
    <cheveux>chatain</cheveux>
  </physique>
  <adresse>
    <rue>rue de Mandur</rue>
    <code_postal>76000</code_postal>
    <ville>Rouen</ville>
  </adresse>
</personne>
```

**6.2.1 Les éléments**

Les éléments sont les objets les plus importants des documents XML. En effet, fondamentalement, les documents XML sont des hiérarchies strictes d'éléments. Ainsi, il existe toujours un (et un seul) élément supérieur qui contient tous les autres. De plus, un élément peut contenir d'autres éléments ou du texte (un type particulier d'élément), et se situe à l'intérieur d'un seul élément [Sevig, 02].

**Syntaxe :**

```
<nom>contenu_élément</nom>
```

Voici un document XML bien formé et complet:

```
<? xml version="1.0"?>
<livre>
  <titre>Introduction à XML</titre>
  <édition>Deuxième édition</édition>
</livre>
```

Trois éléments composent ce document:

- 1) Un élément supérieur **livre**
- 2) Un élément **titre**
- 3) Un élément **édition**

Les éléments se reconnaissent par l'utilisation d'une syntaxe particulière, soit une balise d'ouverture constituée du caractère "<" suivi du nom de l'élément suivi du caractère ">", et enfin d'une balise de fermeture qui reprend la même syntaxe mais cette fois en ajoutant "/" entre le "<" et le nom de l'élément.

L'élément **livre** contient deux sous-éléments, alors que les éléments **titre** et **édition** ne contiennent que du texte. Cet exemple illustre également que les caractères accentués peuvent être utilisés sans problèmes dans les noms d'éléments.

### 6.2.1.1. Contenu d'un élément

Un élément peut contenir d'autres éléments, des données, des références à des entités, des sections littérales et des instructions de traitement.

Un élément peut avoir un contenu récursif, c'est à dire qu'il peut contenir une instance du même type d'élément que lui-même.

Les données des éléments peuvent contenir tous les caractères autorisés sauf le *et commercial* & et le *inférieur* <.

### 6.2.1.2. Les éléments vides

Nous définissons parfois des éléments sans contenu. En HTML, ce sont notamment les balises pour les images, les files horizontaux et les sauts de ligne (<IMG>.<HR> et <BR>). En XML, ces éléments s'écrivent avec une syntaxe spécifique, comme <IMG/>,<HR> et <BR/>.

Un document XML valide doit déclarer les éléments vides comme les non vides. Par définition, ils n'ont pas de sous-éléments, ce qui rend leur déclaration très simple. Il suffit d'utiliser la balise <! ELEMENT> mais en utilisant le mot réservé **EMPTY** à la suite du nom de l'élément (attention les majuscules sont distinguées des minuscules) [XMLG, 00].

#### Exemple:

```
<! ELEMENT BR EMPTY>  
<! ELEMENT IMG EMPTY>  
<! ELEMENT HR EMPTY>
```

### 6.2.2. Les attributs

Nous avons vu qu'un élément XML se définit entre des balises : <element>Valeur de l'element</element>.

En plus de sa valeur, il est possible d'associer des informations supplémentaires à un élément : un attribut.

**Notation** : <element attribut="valeur de l attribut">Valeur de l'élément</attribut>

**Exemple :** `<personne sexe="masculin">Mohamed</personne>`

### 6.2.3. Les Commentaires

Des commentaires peuvent être insérés dans les documents. Ils sont encadrés par les marques de début `<!--` et de fin `-->` de commentaire.

**Syntaxe :** `<!-- commentaire -->`

Le corps du commentaire peut contenir n'importe quel caractère à l'exception de la chaîne `--`. On ne peut donc pas inclure un commentaire dans un autre. Le commentaire ne peut pas non plus être inclus à l'intérieur d'une balise [Steph, 00].

## 7. Conclusion

Dans ce chapitre nous avons présenté le langage XML, les avantages et la syntaxe d'un document XML, soulignant leur manière de structure.

XML est un standard de données, sera le vecteur des transactions commerciales, de la publication de profils de préférences individuelles, de la collaboration automatisée et du partage de bases de données. Il permettra une grande variété de nouvelles utilisations, reposant toutes sur une représentation standard visant à transmettre des données structurées sur le Web aussi facilement qu'il est possible actuellement de déplacer les pages HTML.

Dans le prochain chapitre nous présentons le développement de l'outil de définition de workflow, où nous utilisons le langage XML pour atteindre notre objectif.



# **CHAPITRE III :**

## **LE CYCLE DE DEVELOPPEMENT DE L'OUTIL WORKFLOW**

### **1. Introduction**

La réalisation d'un système de gestion de workflow permet d'interpréter et de stocker la définition de processus.

Le travail présenté consiste à développer un outil de définition de workflow, pour cette objectif, dans cette partie nous décrivons les étapes de développement de notre logiciel qu'on a choisie, cette démarche permet de faciliter la compréhension de l'application et de réduire la complexité.

Ce chapitre est organisé comme suit :

Nous débuterons ce chapitre par la présentation du choix de la démarche suivi dans notre étude, dont il nous citons les fonctionnalités de notre système.

Ensuite, nous présentons les besoins identifier de manière formelle en utilisant le langage de modélisation UML en définissant les diagrammes de cas d'utilisation et les scénarios aux cas d'utilisation, enfin en arrivons au méta modèle qui sera le squelette de notre système.

L'étape suivante consiste à décrire la principale étape du développement : l'étape de conception. Après nous présentons l'implémentation du système en utilisant XML et Visual Basic 6.

Pour tester les résultats de notre outil de définition de workflow, nous prenons un exemple d'application, le processus de développement d'un nouveau produit.

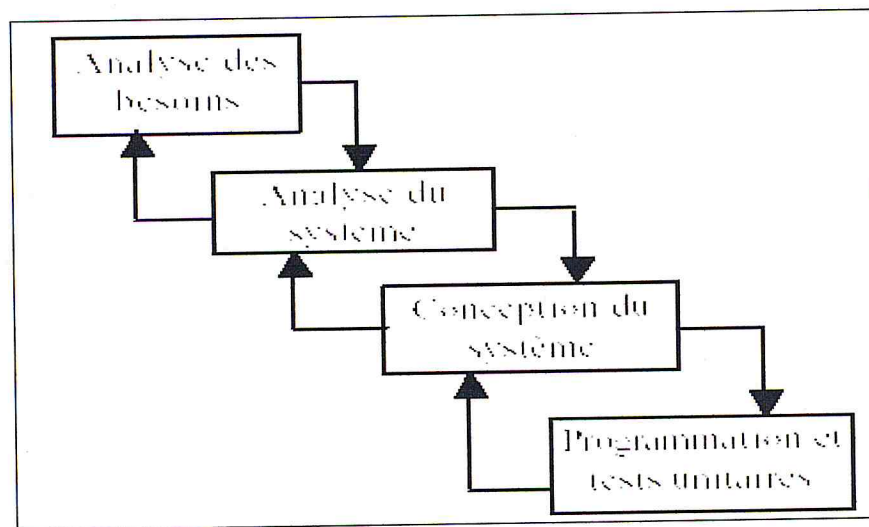
## 2. La démarche suivi

Un cycle de vie d'un logiciel est une Méthodologie de développement de logiciel qui relie les différentes étapes de conception [LAURE, 02].

A fin des années 60 [Claude, 96], le concept de développement en (ou phases) est apparu, un étape ce termine par la production de documents qui sont vérifiés et validés avant de passer à l'étape suivante, ceci permet de contrôler l'avancement des activités en cours et la validité des résultats intermédiaires.

Une Méthodologie de développement de logiciel comprend une notation dont l'objectif est d'assurer le rendu visual des éléments de modélisation.

Le modèle de développement de logiciel, le plus connu est le modèle de "cascade" [Claude, 96], introduit en 1970 (Royce) [Claude, 96]. Il décrit le cycle de vie d'un logiciel par une suite des phases s'enchaîne dans un déroulement linière, depuis l'analyse des besoins jusqu'à la maintenance [Muller, 97]. Dans notre travail nous avons suivi le modèle "cascade".



**Figure III.1** : Le cycle de vie des logiciels en cascade [BOURES, 03].

La notation UML peut être considéré comme une unification de la méthode OOSE, une approche qui offre une excellent support pour l'ingénierie des besoins, de méthode OMT-2 adéquate pour l'étape d'analyse et enfin de Booch'93, qui offre une notation riche pour la phase de conception, donc on peut dire que UML couvre tous les étape de développement, pour cette raisons nous avons choisis UML pour modéliser notre système.

### 3. Spécification des besoins : Etape 1

La spécification des besoins commence par la détermination du quoi faire, c'est-à-dire des besoins de l'utilisateur. Le travail réalisé doit répondre aux attentes des utilisateurs que l'on va exprimer dans la partie suivante à l'aide des cas d'utilisations en précisant les scénarios et les acteurs pour chaque cas d'utilisation.

#### 3.1. Les cas d'utilisations

Les cas d'utilisation (use cases) servant à saisir le comportement attendu d'un système en cours de développement sans avoir à préciser la façon dont ce comportement est réalisé [Muler, 97]. Les cas d'utilisation permettent aux développeurs d'atteindre une compréhension commune avec les utilisateurs finaux du système et les experts du domaine. De plus, leur utilisation aide à valider l'architecture et à suivre l'évolution du système en cours de son développement. Au début de cette étape, il faut définir les acteurs (utilisateurs) de chaque cas d'utilisation.

##### 3.1.1. Les acteurs

Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation. C'est pour eux que le système est fabriqué. Les acteurs qui peuvent utiliser notre logiciel « l'outil de définition » sont :

- **Un expert** : c'est un travailleur dans l'entreprise qui est chargé de faire la définition des processus métier, qui définit un modèle graphique du processus, ce modèle graphique est utilisé par notre système pour générer la définition du workflow (fichier XML), l'utilisateur (l'expert) doit fournir des mise à jour de modèle.
- **Un consultant** : qui veut consulter les modèles, il peut consulter les modèles graphiques et les définitions du workflow (fichier XML) seulement, il ne peut pas définir ou modifier un modèle.

##### 3.1.2. Descriptions des cas d'utilisation

Les cas d'utilisations de notre outil de définition sont :

- 1) **La génération de définition** : qui utilise le cas d'utilisation « Modélisation graphique » cette relation exprimé par « uses ». Pour générer la définition d'un workflow par un fichier XML il faut que le modèle graphique d'un processus soit

définit, la définition d'un workflow est réalisée par la détermination des tâches et leurs propriétés, les transitions et les connecteurs qui les relient, les rôles qui les réalisent.

**2) La consultation des processus :** elle inclut deux autres cas d'utilisation, la consultation des modèles graphiques et la consultation des définitions (code XML), cette relation est exprimée par «include».

**3) La mise à jour de processus :** inclut trois autres cas d'utilisation : « la mise à jour des tâches » qui inclut l'ajoute, la suppression, la modification des tâches, « la mise à jour des transitions » qui inclut l'ajoute de transition, la suppression et la modification des transitions, « la mise à jour des propriétés » qui inclut l'ajoute, la suppression et la modification des propriétés.

**4) La modélisation graphique :** qui inclut cinq cas d'utilisation, la définition des tâches, la définition des acteurs, la définition des connecteurs, la définition des rôles, la définition des transitions, les activités.

### 3.1.3. Diagramme de cas d'utilisation

Dans la figure ci-dessus nous présentons le diagramme de cas d'utilisation d'UML de notre système qui permet de structurer la description des cas d'utilisation (use case).

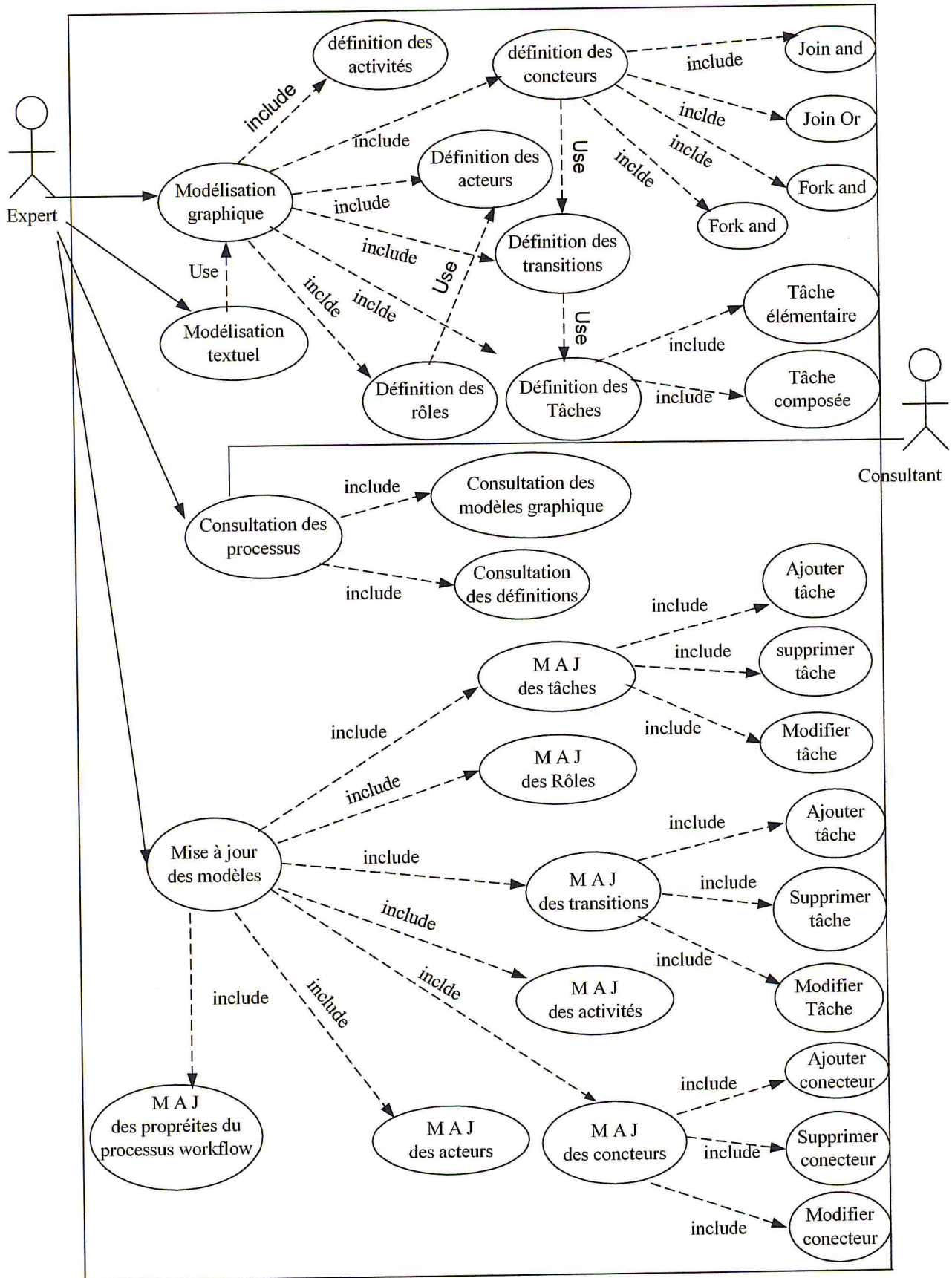


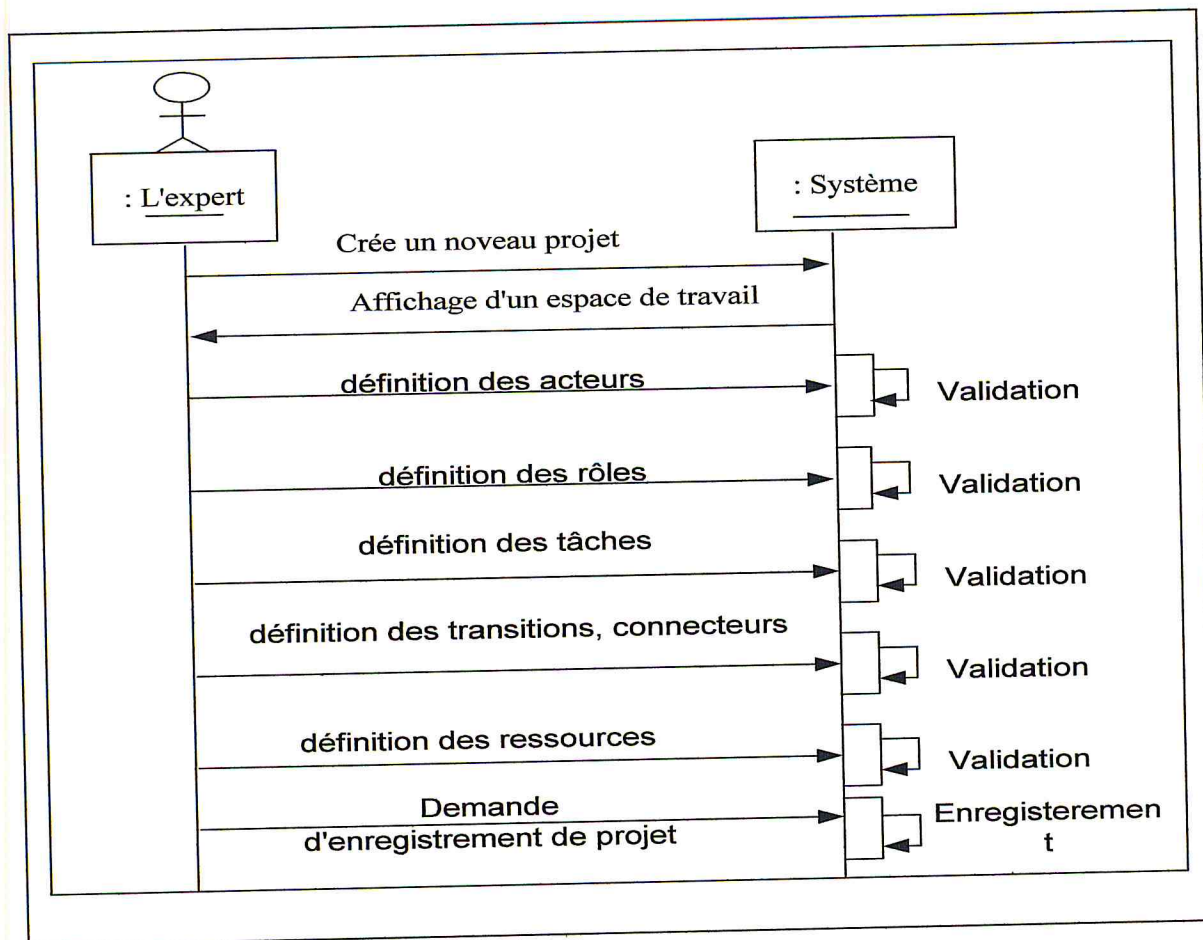
Figure III.2 : Le diagramme de cas d'utilisation de L'outil de définition.

### 3.1.4. Diagramme de séquence

Le diagramme de séquence permet de représenter des collaborations entre les objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Il permet de préciser la réalisation du comportement des cas d'utilisations. Nous illustrons l'étape de "spécifications des besoins" par l'utilisation de diagramme de séquence. La figure ci-dessus contient le diagramme de séquence qui représente le cas d'utilisation « la modélisation graphique ».

#### **Scénario 1 : « la modélisation graphique »**

- 1- L'expert crée un nouveau projet.
- 2- Le système lui affiche un formulaire de propriétés du processus.
- 3- L'expert définit les propriétés du processus.
- 4- Le système valide la définition des propriétés du processus.
- 5- Le système lui affiche un espace de travail.
- 6- L'expert définit les concepts de processus.
- 7- Le système valide la définition des concepts.
- 8- L'expert lance une demande d'enregistrement.
- 9- Le système enregistre le modèle graphique.



**Figure III.3 :** Le diagramme de séquence de « la modélisation graphique ».

### Scénario 2 : « la modélisation textuelle »

- 1- L'expert demande au système d'enregistrer le modèle graphique du processus, après la modélisation graphique.
- 2- Le système doit lancer une commande de génération au parseur, puis le parseur contrôle et génère la définition du processus. (L'intérêt et les travaux d'un parseur sont présentés dans la section 5.2.2.8).
- 3- Le parseur envoie la réponse de génération au système.
- 4- S'il y a erreur, le système envoie un message d'erreur à l'expert.

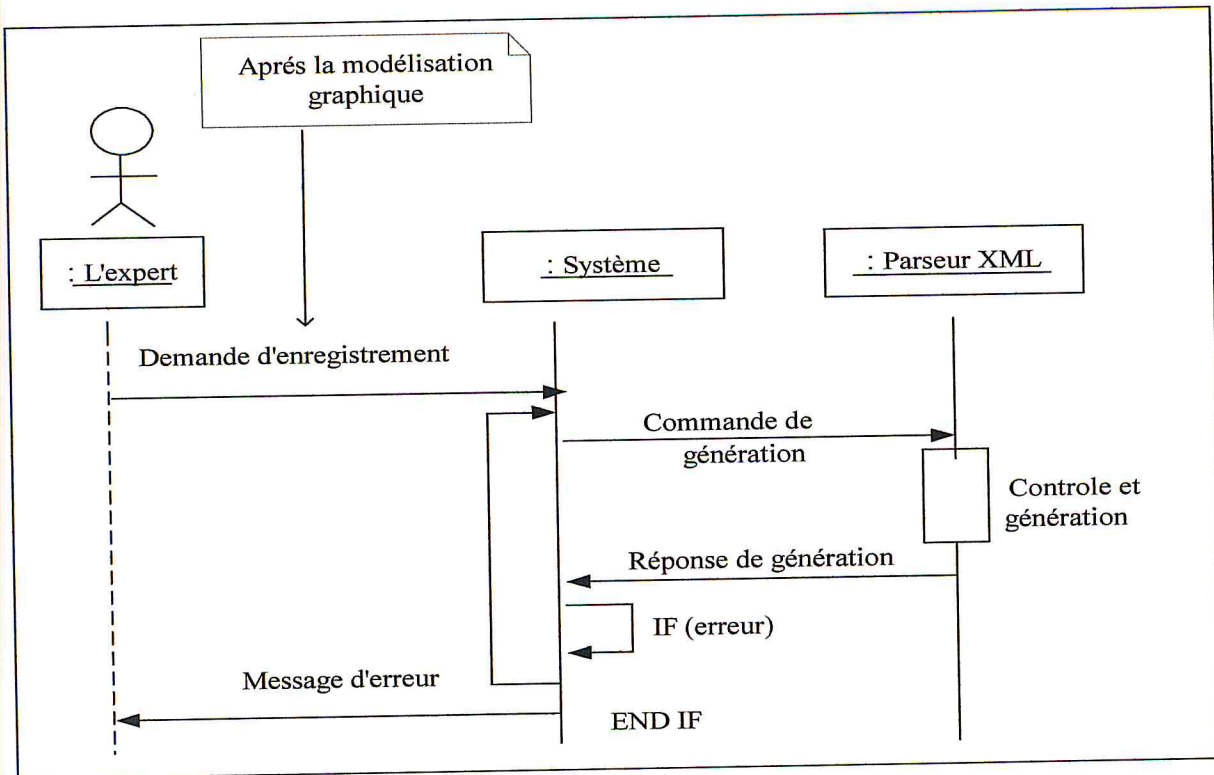


Figure III.4 : Le diagramme de séquence de la modélisation textuelle.

### 3.1.5 Diagramme de collaboration

Ce diagramme laisse apparaître l'organisation des objets qui envoient et reçoivent des messages pour un cas d'utilisation spécifique [VIG, 03]. Ce diagramme permet de décrire les interactions entre les objets intervenant dans la réalisation d'un scénario d'un cas d'utilisation.

Le diagramme de collaboration ci-dessus représente le cas d'utilisation « la génération de définition » :

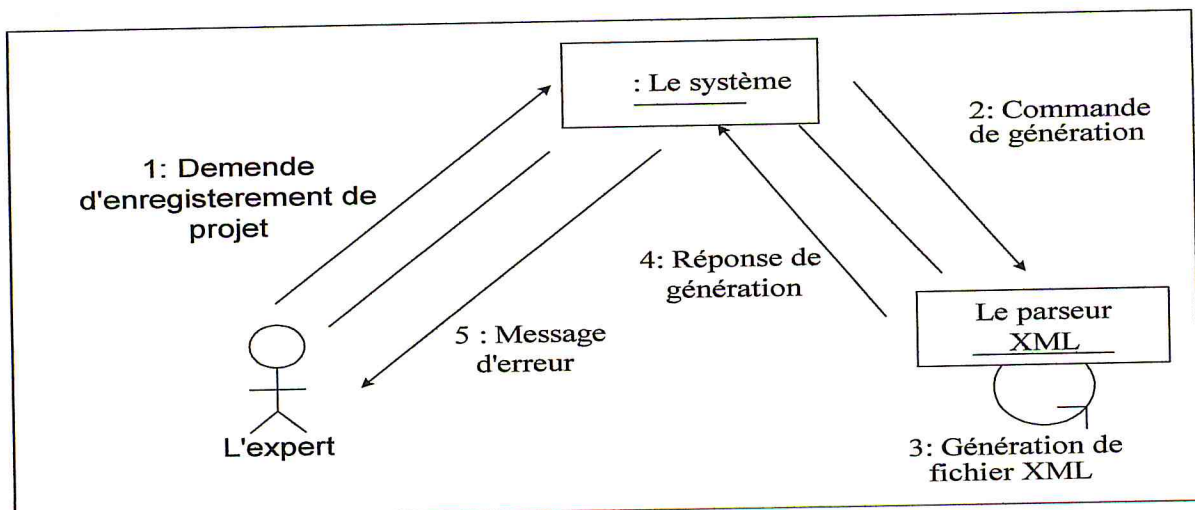
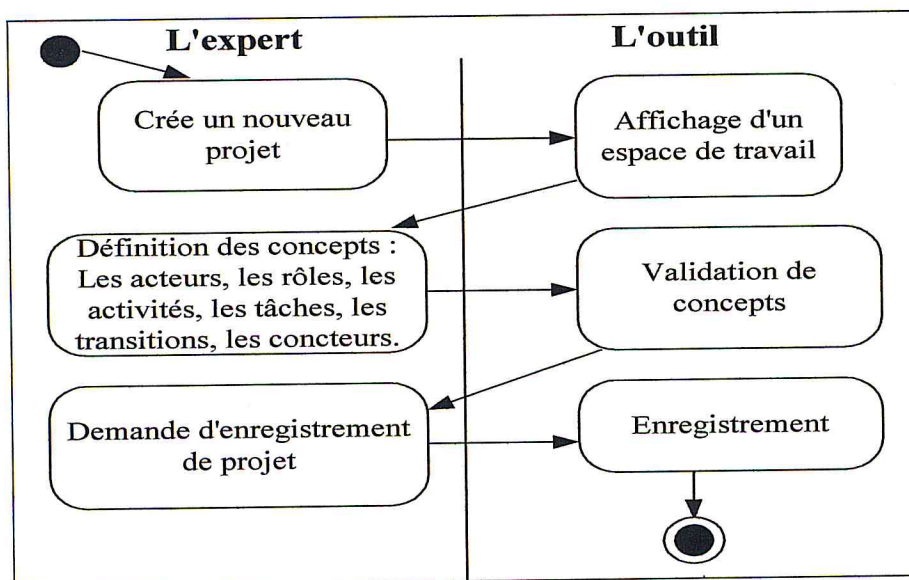


Figure III.5: Le diagramme de collaboration de génération de fichier XML.



### 3.1.6. Diagramme d'activité

UML permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation, à l'aide de diagramme d'activité [Muller, 97], la transition entre les activités se fait automatiquement dès l'activité s'achève. Le diagramme d'activité peut être découpés en couloirs d'activités pour montrer les différentes responsabilités au sein d'une organisation. Nous utilisons ce diagramme afin mieux spécifier les besoins d'utilisateur. Le diagramme d'activité ci-dessus représente le cas d'utilisation «la modélisation graphique ».



**Figure III.6 :** Le diagramme d'activité de cas d'utilisation « la modélisation graphique ».

Le diagramme d'activité suivant représente le cas d'utilisation « la modélisation textuelle » :

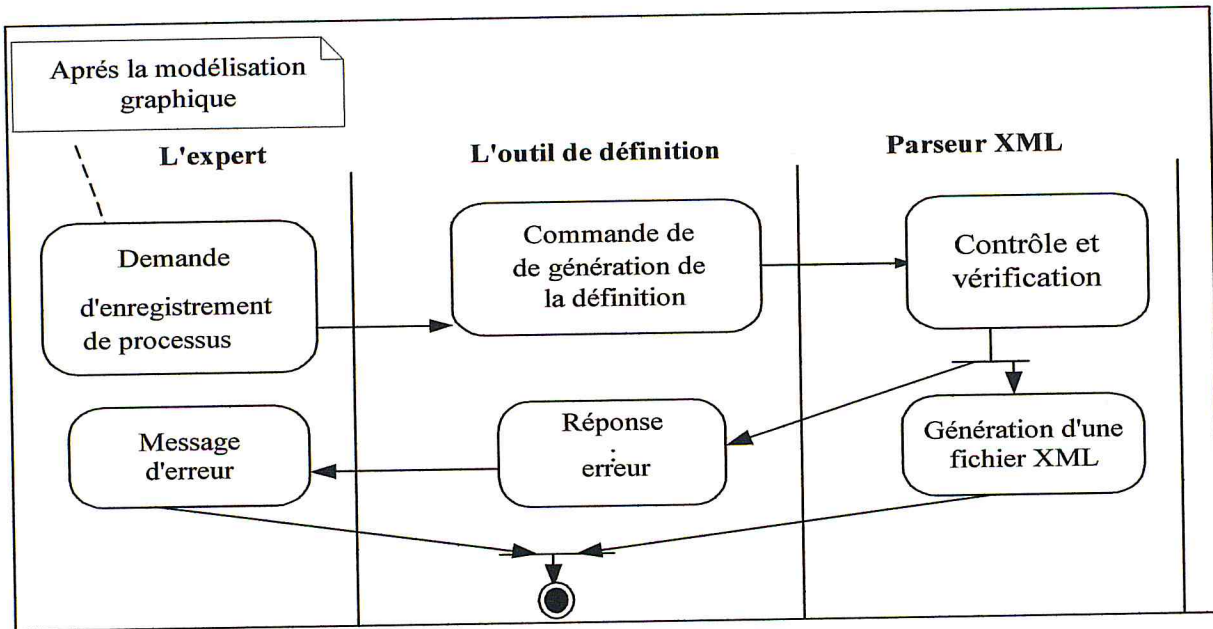


Figure III.7 : Le diagramme d'activité de cas d'utilisation «la modélisation textuelle».

## 4. Etape 2 : L'analyse

L'analyse permet de mettre en évidence les entités du système. En particulier, on fait apparaître des objets qui prennent en charge une partie de travail. De ces objets et de leurs liens, on extrait le diagramme des classes, qu'on enrichit avec la connaissance du domaine du problème [Vaill, 01].

L'analyse de système permet de définir un modèle conceptuel (ou spécification conceptuelle) [BOURES, 03].

L'objectif de cette partie est d'accéder à une compréhension plus aiguë des besoins et des exigences, pour permettre de choisir la conception de la solution et d'en livrer une description facile à entretenir, favorisant la structuration de l'ensemble du système y compris de son architecture. Il s'agit du raffinement des exigences.

Dans cette phase nous déterminons les éléments intervenant dans le système à construire, ainsi que leur structure et leurs relations.

Le premier objectif de notre travail est la modélisation graphique d'un processus workflow. Un processus workflow relie des tâches multiples et une tâche est effectuée par un rôle. Afin d'atteindre cet objectif il nous faut un méta modèle regroupe l'ensemble des concepts d'un processus workflow.

### 4.1 Les concepts d'un processus workflow

Les principaux concepts d'un processus workflow sont :

#### 4.1.1. Une activité

Un processus métier est une séquence d'actions sur les données, chaque action étant aussi nommée activité. Le diagramme d'activités d'UML est l'exemple parfait de la représentation de la séquence d'activités d'un processus métier [Rivard, 03].

#### 4.1.2. Un acteur

La séquence d'action, automatisée ou non, est remplie par un nombre variable d'acteurs. Pour comprendre le processus, la première activité consiste à définir le rôle de chacun des acteurs, son ordre d'apparition et d'intervention dans la séquence. Au préalable on définit les acteurs eux-mêmes, en tant que personnes uniques clairement identifiées, ou en tant qu'appartenant à un groupe de collaborateurs présentant le même profil [Rivard, 03].

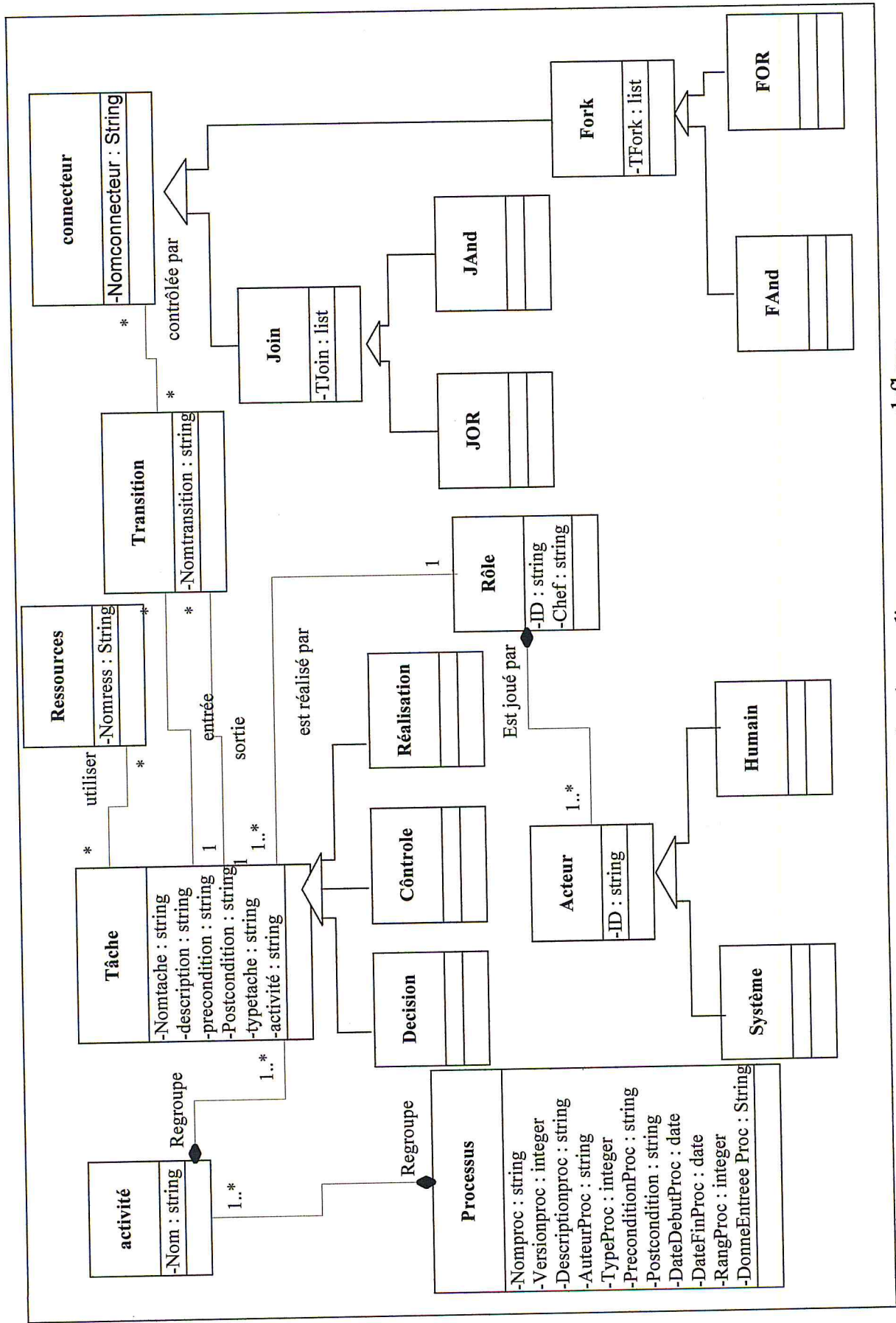


Figure III.11 :le diagramme de classe d'un processus workflow

Le méta modèle que nous avons d'écrit, permet de représenter la structure d'un processus workflow qu'est composé d'un ensemble d'activités et chaque activité est un ensemble de tâches.

Une tâche dérive trois sous classe : tâche de type décision, contrôle et réalisation, la relation entre les tâches est exprimée par la classe transition. Une tâche est réalisée par un rôle qui peut être joué par un ou plusieurs acteurs dont un est élu comme chef. Un acteur peut être soit humain ou système. Une tâche utilise des ressources pour l'exécution, ces ressources peuvent être des documents ou des applications.

#### 4.5. Diagramme d'états-transition

Le diagramme d'états-transition visualisent des automates d'états finis du point de vue des états des transitions [Muller, 97]. Ce diagramme devrait être crée pour tous les objets qui passent par multiples d'états, et des actions spécifiques doivent être effectuées entre les états.

Nous utilisons le diagramme d'état de transition pour représenter les différents états que peuvent prendre un rôle et un acteur.

La figure III.12 représente le diagramme d'état de transition de la classe rôle, les états qui doivent gérer un rôle sont d'écrit ci-dessus :

- **Création** : dans cet état ce fait la construction d'un rôle.
- **Définition** : dans cet état les propriétés d'un rôle doit être définit.
- **Enregistrement** : après la définition des propriétés d'un rôle, il faut l'enregistrer dans la mémoire.
- **Modification** : Cet état permet de modifier les propriétés d'un rôle existe déjà.
- **Supprimer** : cet état permet de supprimer un rôle.

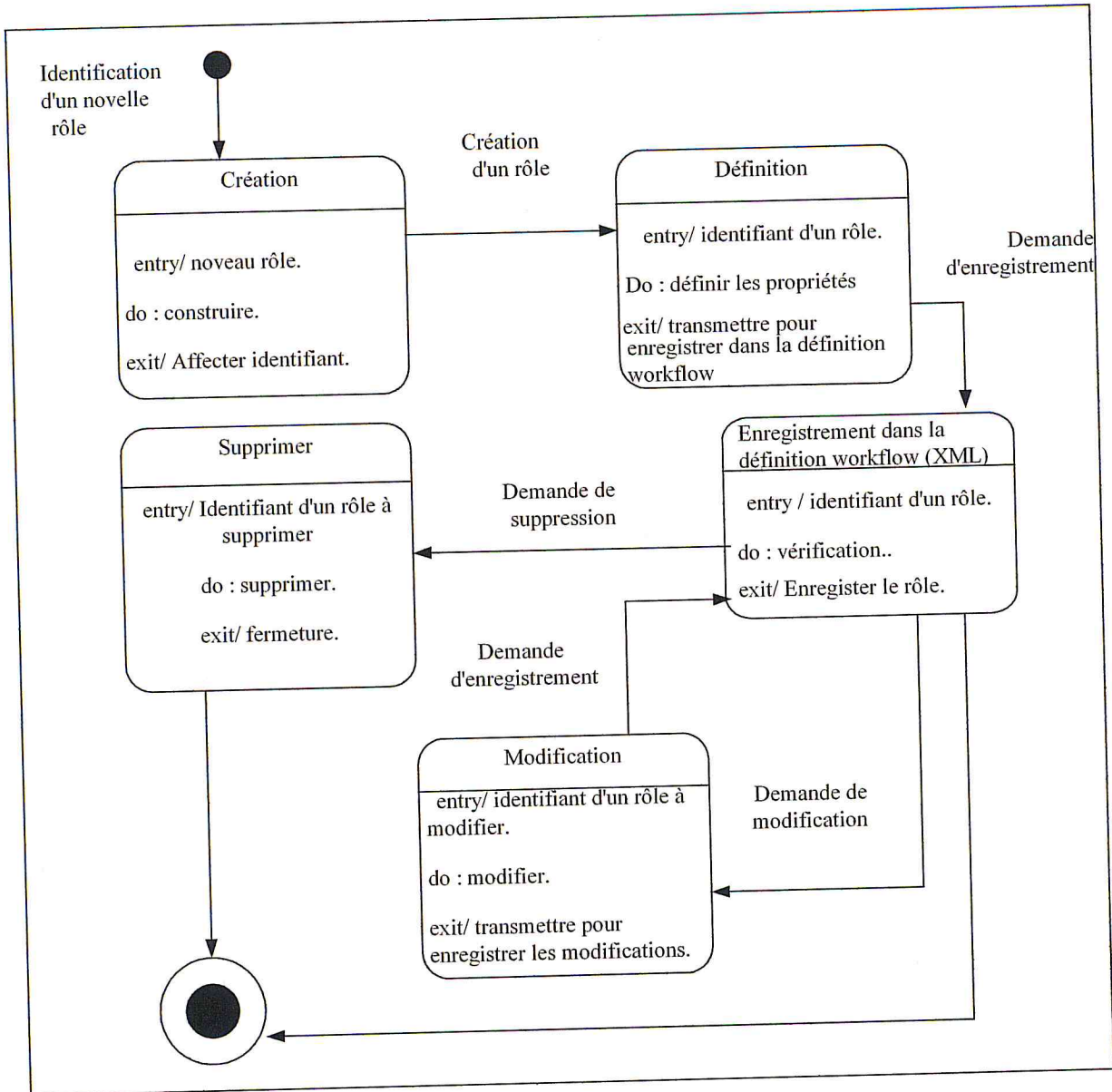


Figure III.12 : Le diagramme d'états-transitions pour la classe rôle.

Le diagramme d'état transitions (Figure III.13) représente la classe acteur. Les états d'un acteur sont :

- **Création** : dans cet état ce fait la construction d'un acteur.
- **Définition** : dans cet état les propriétés d'un acteur doit être définit.
- **Enregistrement** : après la définition des propriétés d'un acteur, il faut l'enregistrer dans la base de données.
- **Modification** : Cet état permet de modifier les propriétés d'un acteur existe déjà.
- **Supprimer** : cet état permet de supprimer un acteur.

### 4.1.3. Un rôle

Un rôle est une collection d'acteurs, collaborant afin d'atteindre un objectif commun.

Un rôle est l'entité chargée de l'exécution des tâches

### 4.1.4. Une tâche

Une Tâche est une unité de travail, le délai de réalisation d'une tâche est surveillé par un "timer ". La tâche correspond à un niveau de décomposition de l'activité.

Dans un workflow, on peut distinguer des tâches manuelles et des tâches automatisées. Ces deux catégories de tâches sont prises en compte dans la conception et la réalisation d'un workflow. Les tâches dites « manuelles » ne sont pas contrôlables par l'application de workflow. Par exemple, l'appel d'un fournisseur au téléphone pour négocier un prix qui sera ensuite enregistré dans l'édition d'une commande, sera considéré comme une tâche « manuelle » alors que l'édition de la commande sera considérée comme une tâche automatisée.

## 4.2. Les composants fonctionnelles d'un processus workflow

La décomposition progressive d'un processus permet d'identifier les tâches. Une tâche est de type décision, contrôle ou réalisation. La relation entre les tâches est exprimée par la classe *transition*, les transitions sont contrôlées par des connecteurs fork et join. Le diagramme de classe suivant représente les composants constituant un processus workflow, indépendamment des moyens et des ressources mis œuvre pour les réaliser (rôles, ressources).

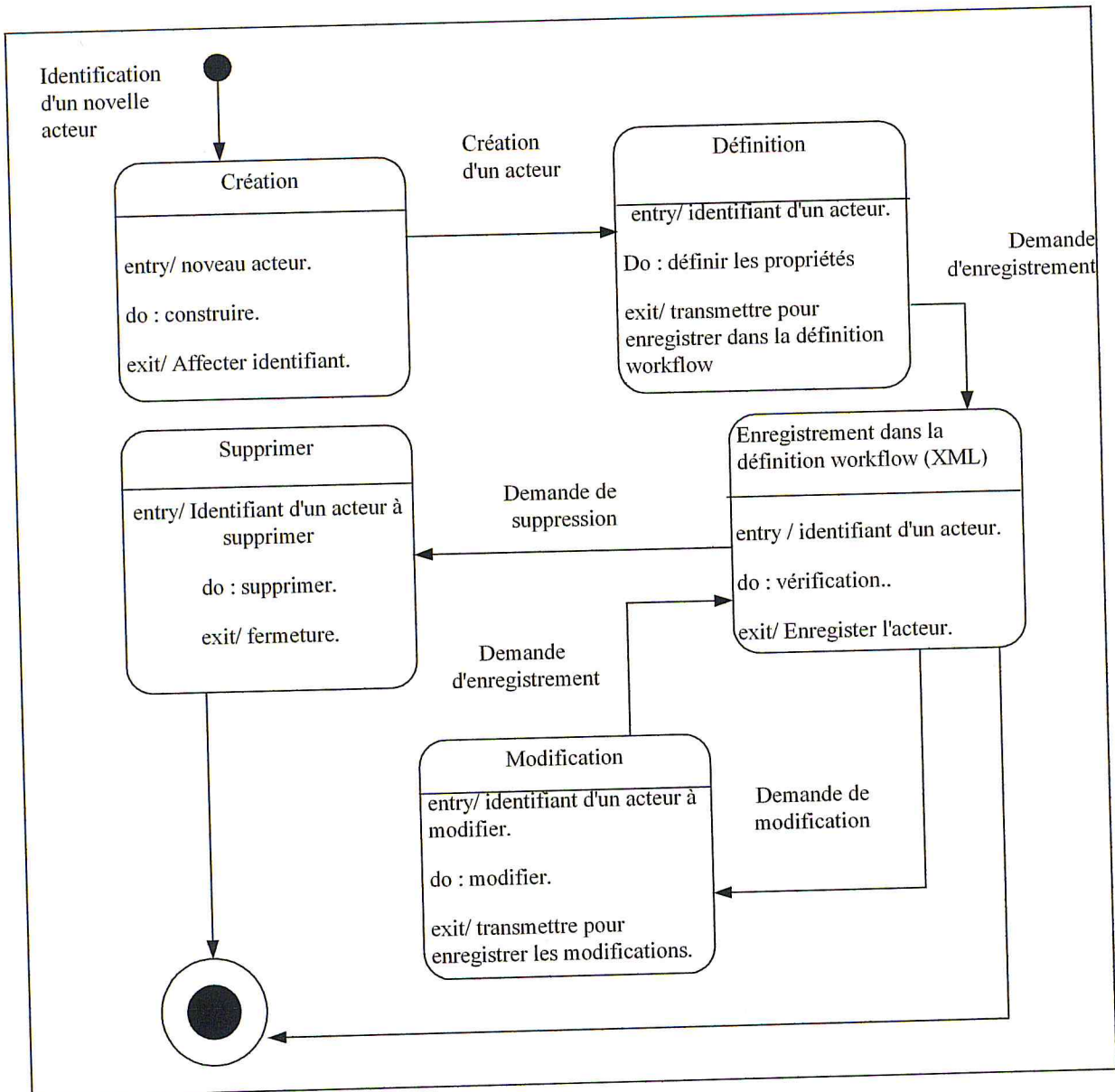


Figure III.13 : Le diagramme d'états-transition pour la classe acteur.



### 5. La conception

Dans l'analyse, nous avons mis en évidence une première structure logique du système. L'analyse sert à définir l'architecture du système informatique et l'organisation des composants logiciels [Vill, 01]. L'objectif de la conception est d'acquérir une compréhension approfondie des questions concernant les exigences fonctionnelles et contraintes liées.

Dans cette phase nous décrivons l'architecture du système de manière générale, après nous détaillons la conception, on utilisant les diagrammes d'UML.

#### 5.1. Conception globale (générale)

Dans cette partie nous présentons les modules de notre système, en déterminons les interfaces entre ces modules comme nous montre la figure III.14. Nous avons représenté le module « Echange » comme un lien entre le module « modélisation » et le module « génération ».

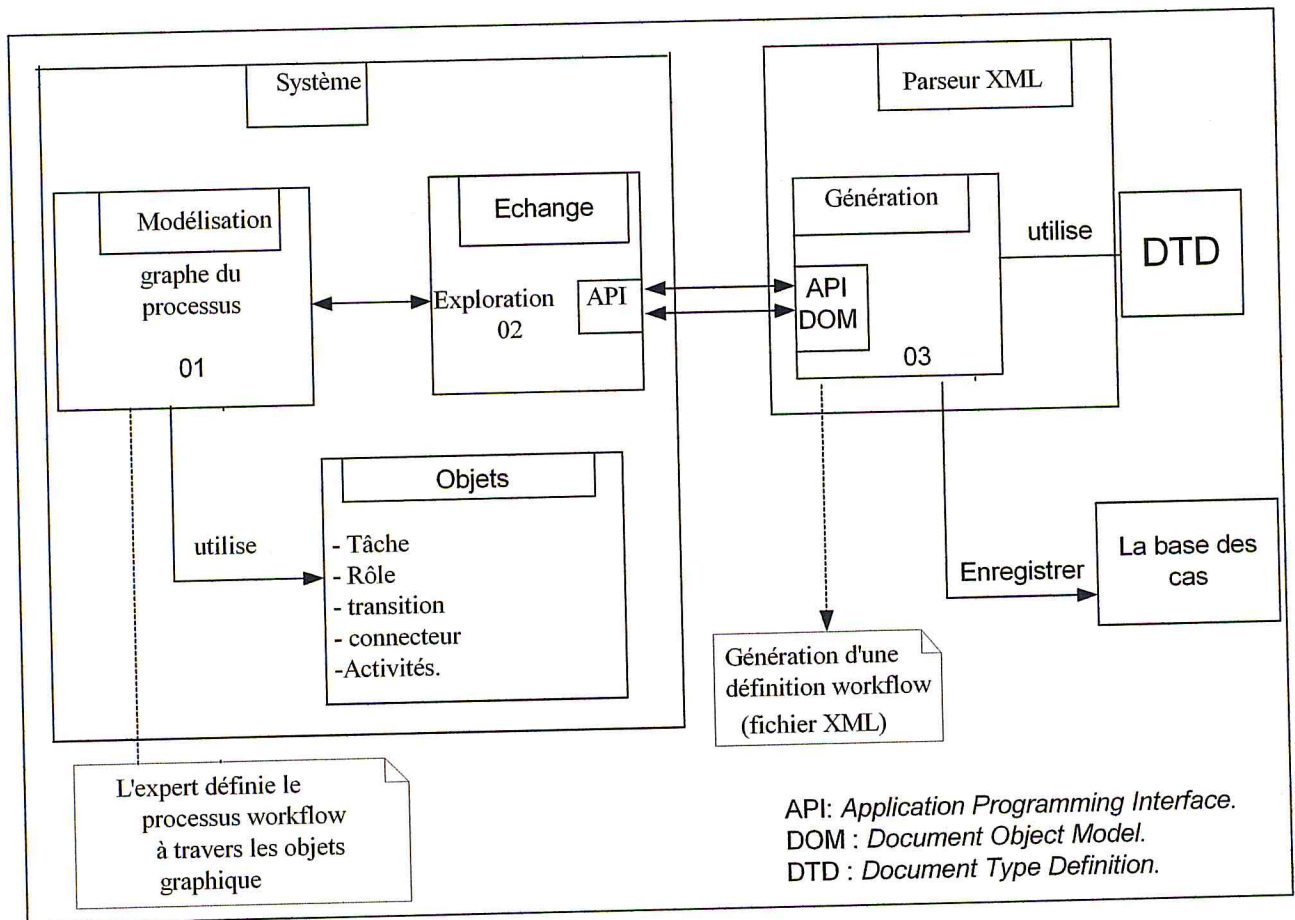


Figure III.14 : L'architecture générale de l'outil de définition de wokflow.

- **Module 1** : La modélisation graphique du processus.
- **Module 2** : Echange.
- **Module 3** : La génération de définition.

L'interface de l'échange permet au module « modélisation » d'envoyer les modèles au parseur, d'accéder à la base des cas et de récupérer des données de parseur.

L'architecture de notre système est composée de trois modules :

Le module «la modélisation graphique» utilise la classe "des objet" pour réaliser la modélisation d'un processus, le résultat de cette module est un graphe indique la représentation graphique d'un processus workflow, l'extension de graphe est (.PWF).

La génération de définition est réalisée à travers le module «la génération de définition», le résultat de ce module est un fichier textuel d'extension (.XML).

Le changement d'information est assuré par le module «Echange». Après la modélisation graphique d'un processus workflow au niveau du module «modélisation graphique», les objets du processus seront envoyés au module «génération de définition» à travers le module «Echange» , le parseur fournit des interfaces qui permettent au module «Echange» de manipuler les données.

## 5.2. La conception détaillée

La conception détaillée précise le contenu des composants (classes, autres composants, bibliothèques, programmes) [Vill, 01]. Elle fournit une description détaillée pour chaque module, dans cette partie nous précisons comment les fonctions de composants sont réalisées, et la manière de représenter les données.

### 5.2.1. Module 01 : La modélisation graphique

Un processus se décrit sous la forme d'un graphe, les nœuds représentent les tâches élémentaires et composée, les arcs représentent les transitions d'une tâche vers une autre. Ce module utilise la classe objet pour représenter les éléments métier d'un processus. Pour la construction d'un modèle de processus nous avons utilisé les éléments représentés dans la figure suivant :

### 4.1.3. Un rôle

Un rôle est une collection d'acteurs, collaborant afin d'atteindre un objectif commun.  
Un rôle est l'entité chargée de l'exécution des tâches

### 4.1.4. Une tâche

Une Tâche est une unité de travail, le délai de réalisation d'une tâche est surveillé par un "timer ". La tâche correspond à un niveau de décomposition de l'activité.

Dans un workflow, on peut distinguer des tâches manuelles et des tâches automatisées. Ces deux catégories de tâches sont prises en compte dans la conception et la réalisation d'un workflow. Les tâches dites « manuelles » ne sont pas contrôlables par l'application de workflow. Par exemple, l'appel d'un fournisseur au téléphone pour négocier un prix qui sera ensuite enregistré dans l'édition d'une commande, sera considéré comme une tâche « manuelle » alors que l'édition de la commande sera considérée comme une tâche automatisée.

## 4.2. Les composants fonctionnelles d'un processus workflow

La décomposition progressive d'un processus permet d'identifier les tâches. Une tâche est de type décision, contrôle ou réalisation. La relation entre les tâches est exprimée par la classe *transition*, les transitions sont contrôlées par des connecteurs fork et join. Le diagramme de classe suivant représente les composants constituant un processus workflow, indépendamment des moyens et des ressources mis œuvre pour les réaliser (rôles, ressources).

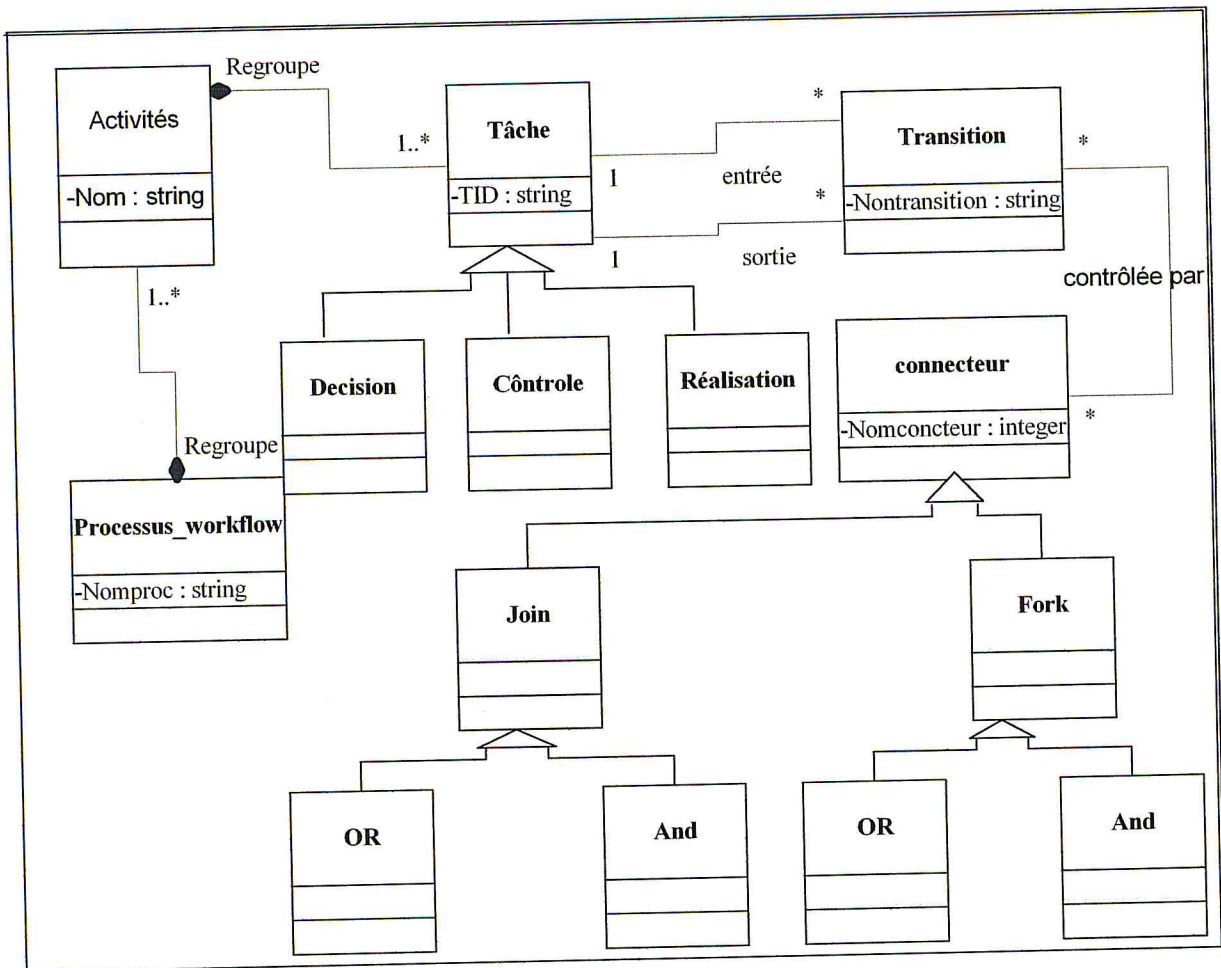


Figure III.8 : Le diagramme de classe de composants fonctionnels d'un processus.

### 4.3. Les composants organisationnels d'un processus workflow

L'exécution de chaque tâche est réalisée par un rôle. Dans cette partie, nous décrivons les moyens humains, techniques et organisationnels utilisés pour accomplir les objectifs attendus du processus, ils ont réalisés de manière collaborative par un ensemble d'acteurs auxquels sont assignés des rôles. Un acteur exerce plusieurs rôles, et un rôle est attribué à plusieurs acteurs, un acteur est de type humain ou système.

Le diagramme de classe suivant représente les composants organisationnels d'un processus :

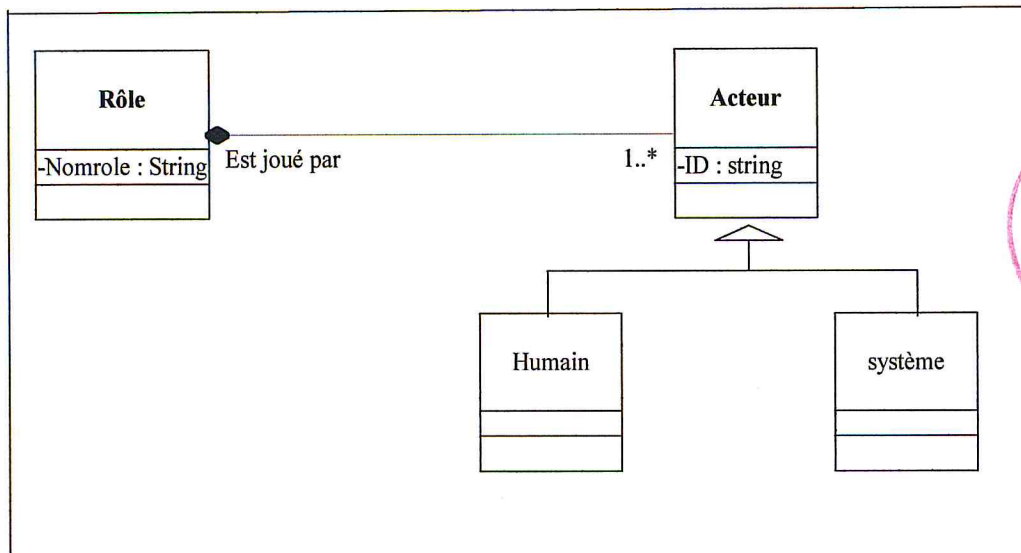


Figure III.9 : Le diagramme de classe de composants organisationnels.

#### 4.4. Le diagramme de classe

Le diagramme de classe exprime la structure statique d'un système, en terme de classes et de relations entre ces classes [Muller, 97]. Le diagramme de classe (Figure III.11) contient toutes les entités de notre système, et les relations entre eux. Nous utilisons ce diagramme pour représenter le méta modèle, qui représente les éléments constituant un processus workflow (définition méta modèle).

Le diagramme de classe suivant représente les ressources utilisées par processus, pour exécuter les tâches :

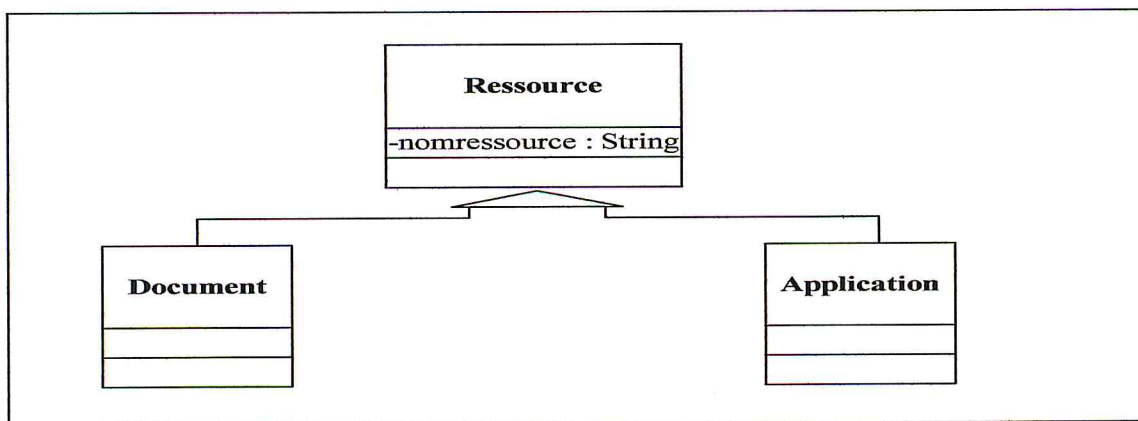


Figure III.10 : Diagramme de classe des ressources.

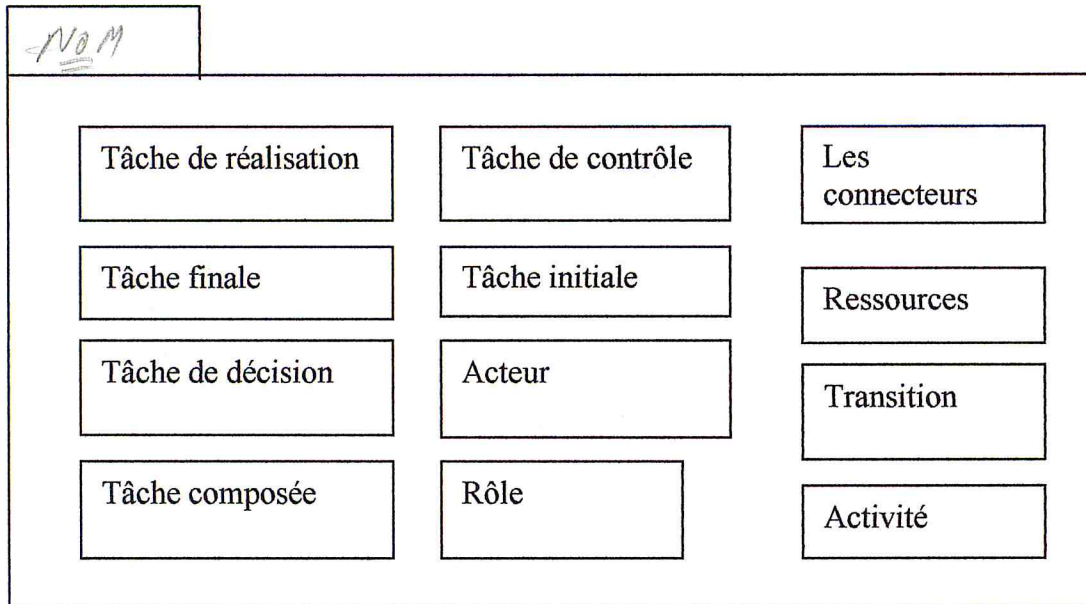


Figure III.15 : Les éléments de modélisation.

La première fonction de notre outil de définition de workflow est la modélisation graphique d'un processus workflow, ce graphe est représenté sous forme d'un diagramme d'activité, ce dernier inclue les informations relatives aux condition de début et de fin, les tâches qui le composent, les règles de navigation, les rôles des participants.

La modélisation graphique d'un processus workflow est réalisée par l'utilisation d'une notation graphique pour représenter les composants d'un processus.

Les notations graphiques utilisées sont représentées dans la figure suivant :

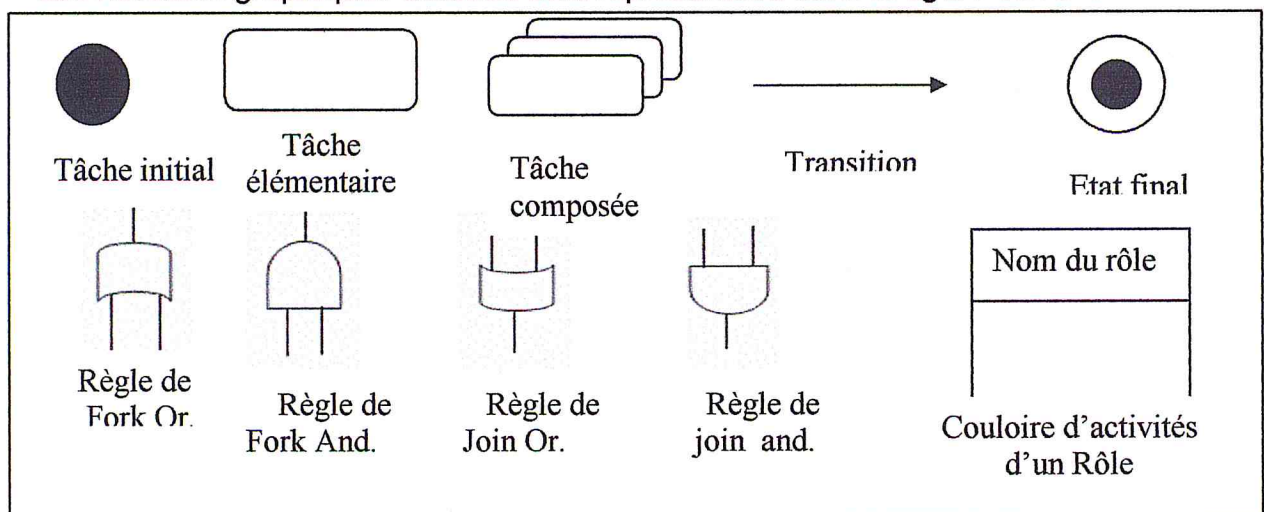


Figure III.16 : Les éléments de modèle de processus.

- **Tâche initial** : Cette tâche représente l'objet début d'un modèle de processus, n'a pas de prédécesseur et à un ou plusieurs successeurs, représenté par une cercle.
- **La tâche élémentaire**: pour représenter une tâche élémentaire, nous avons utilisé un rectangle étiquetée par le type de tâche, la tâche de type *Décision*, de type *Contrôle* et de type *Réalisation*.
- **Tâche final** : cette tâche représente la fin d'un processus workflow, il existe une et une seule fois dans un modèle de processus. L'objet fin à au moins un prédécesseur et n'a pas de successeur.
- **Transition** : La transition d'une tâche à une autre est représentée par une flèche reliant les tâches suivant une règle d'enchaînement.
- **Rôle** : Un rôle est un ensemble d'acteur, il est responsable de réaliser des tâches, pour représenter un rôle dans le graphe nous avons utilisé le dessin définit dans la figure III.16.
- **La règle Fork** : C'est une expression booléenne, décrivant le type de la relation entre deux tâche. Elle permet de représenter deux types de relation OR et AND.
- **La règle de jointure** : C'est une règle pré condition permet de jointure plusieurs tâches pour enchaîner avec une nouvelle tâche.
- **La tâche composée** : une tâche composée est composée de plusieurs tâche élémentaire, pour représenter la tâche composée nous avons utilisé la notation graphique définit dans la figure III.16.

### 5.2.2. Module 3 : La génération de définition

Au niveau de ce module se fait la représentation textuelle d'un processus workflow, l'interface 1 de modèle de référence de la *workflow management coalision* (VMFC) est désignée sous le terme d'interface d'import/export de définition de processus, qui devrait fournir un format d'échange commun pour les types d'informations suivantes:

- Conditions de déclenchement et de terminaison de processus.
- Identification d'activités dans le processus, incluant les applications externes associées et les données d'ordonnancement de processus.
- Identification des types de données et des chemins d'accès.

- Définition des conditions de transition et des règles de routage.
- Information relative aux décisions d'allocation de ressources.

Pour générer la représentation textuelle d'un processus workflow associée à une représentation graphique, nous avons utilisé le langage XML pour interpréter le méta modèle présenté dans la phase précédente, le fichier généré doit être conforme au méta modèle. Pour cette raison nous proposons de passer par deux étapes :

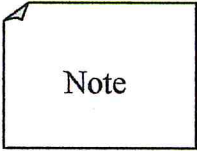
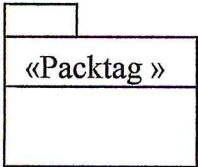
- Le passage** : dans cette étape, il faut faire un passage d'UML vers l'XML, pour présenter le DTD qui inclut la structure d'un fichier XML conforme au méta modèle.
- La validation** : après l'étape de passage il faut valider le document généré, pour cette raison nous avons utilisé un parseur XML.

Un analyseur (parseur) XML lit un document XML et vérifie que son contenu est bien formulé, il peut également vérifier que le document est valide (analyseur validant) [XMLG, 00]

#### 5.2.2.1. Le passage d'UML vers XML

Le méta modèle qui nous avons décrit dans l'étape précédente nous permet de représenter la structure d'un processus workflow. Il s'agit maintenant de lui faire correspondre une représentation concrète en XML. Dans ce but, nous présentons le modèle XML en document de définition de type (DTD), c'est la façon standard de définir des modèles XML.

Pour passer depuis UML vers XML, nous avons défini des règles de passage qui sont résumées dans le tableau suivant [Gard, 02] :

	UML	XML
Les notes		<code>&lt;!--commentaire--&gt;</code>
Paquetage		<code>&lt;!DOCTYPE nom [ ..... ..... &gt;</code>



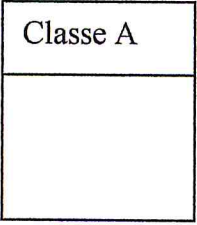
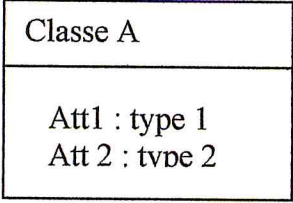
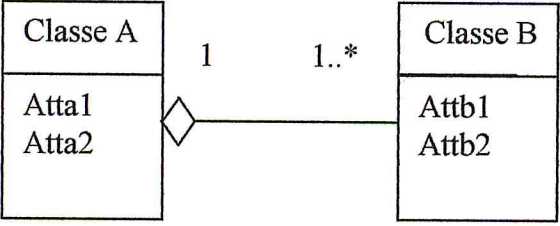
<p>Classe A</p>		<pre>&lt;! ELEMENT A ..... ..... ..... &gt;</pre>
<p>Les attributs</p>		<pre>&lt;! ELEMENT A &lt;! ELEMENT A (Att1,Att2)&gt; &lt;!ELEMENT Att1 type 1&gt; &lt;!ELEMENT Att2 type 2&gt; &gt;</pre>
<p>L'agrégation</p>		<pre>&lt;! ELEMENT B ..... &gt; &lt;! ELEMENT A &lt;!ELEMENT A (Atta1, Atta2, B+)&gt; &gt;</pre>

Tableau III.1 : Le passage d'UML vers XML.

**5.2.2.2. La description d'un processus workflow en XML**

Les schémas de workflow sont stockés en tant qu'une représentation d'un processus. Quand un schéma de workflow est modifié ou mis à jour, il est stocké comme un nouveau processus dans la base de cas. Les processus et les composants d'un processus workflow sont représentés avec les éléments syntactiques et sémantiques bien définis qui sont discutés ci-dessous.

Les processus workflow sont codés dans XML avec les étiquettes appropriées pour différents éléments structuraux. Les détails des éléments structuraux, les étiquettes correspondantes et les représentations XML sont détaillés ci-dessus.

Élément, étiquette de XML	Description	Représentation
NomProc	Chaque schéma est assigné à un nom de processus unique.	Un symbole (WAS2)
VersionProc	La version de processus	Un numéro (02)
DescriptionProc	Une annotation textuelle.	Texte libre
AuteurProc	Le responsable qui modélise ce processus.	Texte
TypeProc	Le type de workflow (Adhoc ; production; Collaboration ; Administration)	Type énumération
DonneeSortieProc	List d'arguments après la terminaison du workflow	Chaque argument est une donnée attribuée à une entité exemple : transaction terminée.
DonneEntreeProc	Liste d'arguments pour initialiser le workflow	Chaque argument est une donnée attribuée à une entité exemple : client.
PreconditionProc	Conditions qui doivent existe dans l'état d'organisation courant pour initialiser le processus courant.	Exprimer comme un prédicat dans la logique de premier ordre.
PosteconditionProc	Conditions qui doivent existe après l'état final d'un processus	Exprimer comme un prédicat dans la logique de premier ordre.
DateDebutProc	La date d'activation du processus workflow	Type date
DateFinProc	La date maximale de la validité de processus	Type date
RangProc	Rang utilisé pour les choix de filtrage pendant la récupération	Entier

**Tableau III.2 : Les éléments d'un processus workflow.**

La première colonne du Tableau identifie L'élément et son étiquette correspondante de XML, les autres colonnes décrivent la sémantique de l'élément et illustrent sa représentation informatique.

### 5.2.2.3. La description d'une tâche

Les éléments structuraux d'une tâche et leur représentation XML sont définis dans le tableau suivant :

Élément, étiquette de XML	Description	Représentation
<b>nomtache</b>	Nom unique de tâche à travers toute descriptions de domaine.	Un symbole (TI-21)
<b>description</b>	Description de la tâche	Texte
<b>precondition</b>	Conditions qui doivent existe dans l'état courant pour initialiser la tâche.	Exprimer comme un prédicat dans la logique de premier ordre.
<b>postcondition</b>	Conditions qui doivent existe après l'état final.	Exprimer comme un prédicat dans la logique de premier ordre.
<b>typetache</b>	Le type de la tâche: élémentaire, composée.	Type énuméré
<b>activite</b>	L'activité qui contient la tâche	Texte
<b>Rôle</b>	Le rôle responsable d'exécuter une tâche.	Texte
<b>Mode</b>	Catégorie d'une tâche.	énumérer

Tableau III.3 : Les éléments d'une tâche.

#### 5.2.2.4. La description d'une transition

Les éléments structuraux d'une transition et leur représentation XML sont définis dans le tableau suivant :

Élément, étiquette de XML	Description	Représentation
<b>nomtransition</b>	Chaque transition à un nom.	Un symbole (TR-1)
<b>dureetransition</b>	La durée de transition	Time
<b>typetransition</b>	Le type de transition	énumérer
<b>tachedebut</b>	La tâche qui déclanche la transition	Tâche
<b>tachefin</b>	La tâche déclanchée par la transition	Tâche

Tableau III.4 : Les éléments d'une transition.

#### 5.2.2.5. La description d'un connecteur

Les éléments structuraux d'un connecteur et leur représentation XML sont définis dans le tableau suivant :

Élément, étiquette de XML	Description	Représentation
<b>nomconnecteur</b>	Chaque connecteur é un nom.	Texte
<b>typeconnecteur</b>	Le type du connecteur.	(Fork, Join)
<b>logiqueconnecteur</b>	L'opération logique du connecteur : And ou OR	(And, OR)
<b>tachecible</b>	La tâche du connecteur.	Tâche
<b>listetransition</b>	La liste des transitions qui appartient à un connecteur.	Id tâche

**Tableau III.5 : Les éléments d'un connecteur**

#### 5.2.2.6. La description d'un rôle

L'exécution de chaque tâche est réalisée par un rôle. Dans cette partie, nous décrivons les moyens humains, techniques et organisationnels utilisés pour accomplir les objectifs attendus du processus, ils ont réalisés de manière collaborative par un ensemble d'acteurs auxquels sont assignés des rôles. Un *acteur* exerce plusieurs rôles, et un rôle est attribué à plusieurs acteurs. Les éléments structuraux d'un rôle sont énumérés dans le tableau suivant :

Élément, étiquette de XML	Description	Représentation
<b>nomrole</b>	L'identifiant d'un rôle.	Un symbole (R-4)
<b>chefrole</b>	Le chef du rôle.	Texte
<b>listeacteur</b>	Collection d'acteur	Acreur

**Tableau III .6 : Les éléments d'un rôle.**

Les éléments structuraux d'un acteur sont énumérés dans le tableau suivant :

Élément, étiquette deXML	Description	Représentation
<b>id</b>	Chaque acteur à un identifiant	Un symbole (A-45)
<b>nomacteur</b>	Le nom de l'acteur.	Texte
<b>TypeActeur</b>	Le type d'acteur : humain, système	Type énuméré

**Tableau III.7 : Les éléments d'un acteur.**

### 5.2.2.7. La description des ressources

Une tâche utilise des ressources pour l'exécution, ces ressources peuvent être un documents ou un application. Le tableau suivant représente les éléments structuraux d'une ressource.

Élément, étiquette de XML	Description	Représentation
<b>nomressource</b>	Le nom de ressource utilisée par une tâche	Texte
<b>typeressource</b>	Chaque ressource à un type	énumérer
<b>url</b>	Le chemin d'accès de la ressource	Texte
<b>listetache</b>	La liste des tâches qui utilisent le ressource.	Texte

**Tableau III.8 : Les éléments d'une ressource.**

La figure suivante illustre le DTD (grammaire XML) du méta modèle :

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT NomPorc (#PCDATA)>
<!ELEMENT VersionProc (#PCDATA)>
<!ELEMENT AuteurProc (#PCDATA)>
<!ELEMENT DescriptionProc (#PCDATA)>
<!ELEMENT TypeProc (#PCDATA)>
<!ELEMENT RangProc (#PCDATA)>
<!ELEMENT DonneEntreeProc (#PCDATA)>
<!ELEMENT DonneeSortieProc (#PCDATA)>
<!ELEMENT PreconditionProc (#PCDATA)>
<!ELEMENT PosteconditionProc (#PCDATA)>
<!ELEMENT DateDebutProc (#PCDATA)>
<!ELEMENT DateFinProc (#PCDATA)>
<!ELEMENT ProprieteProcessus
(NomPorc,VersionProc,AuteurProc,DescriptionProc,TypeProc,RangProc,Do
nneEntreeProc,DonneeSortieProc,PreconditionProc,PosteconditionProc,D
ateDebutProc,DateFinProc)>
<!ATTLIST ProprieteProcessus id CDATA #REQUIRED>
<!ELEMENT nomacteur (#PCDATA)>
<!ELEMENT typeacteur (#PCDATA)>
<!ELEMENT Acteur (nomacteur,typeacteur)>
<!ATTLIST Acteur id CDATA #REQUIRED>
<!ELEMENT ListeActeurs (Acteur)+>
<!ATTLIST ListeActeurs nbacteurs CDATA #REQUIRED>
<!ELEMENT nomactivite (#PCDATA)>
<!ELEMENT Activite (nomactivite)>
```

```

<!ATTLIST Activite id CDATA #REQUIRED>
<!ELEMENT ListeActivites (Activite)+>
<!ATTLIST ListeActivites nbactivites CDATA #REQUIRED>
<!ELEMENT nomrole (#PCDATA)>
<!ELEMENT chefrage (#PCDATA)>
<!ELEMENT acteur (#PCDATA)>
<!ELEMENT listeacteur (acteur)+>
<!ATTLIST listeacteur nbacteur CDATA #REQUIRED>
<!ELEMENT Role (nomrole,chefrage,listeacteur)>
<!ATTLIST Role id CDATA #REQUIRED>
<!ATTLIST Role X CDATA #REQUIRED>
<!ATTLIST Role Y CDATA #REQUIRED>
<!ATTLIST Role W CDATA #REQUIRED>
<!ATTLIST Role H CDATA #REQUIRED>
<!ELEMENT ListeRoles (Role)+>
<!ATTLIST ListeRoles nbroles CDATA #REQUIRED>
<!ELEMENT nomtache (#PCDATA)>
<!ELEMENT typetache (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT precondition (#PCDATA)>
<!ELEMENT postecondition (#PCDATA)>
<!ELEMENT duree (#PCDATA)>
<!ELEMENT mode (#PCDATA)>
<!ELEMENT donnee (#PCDATA)>
<!ELEMENT typedonnee (#PCDATA)>
<!ELEMENT role (#PCDATA)>
<!ELEMENT activite (#PCDATA)>
<!ELEMENT Tache
(nomtache,typetache,description,precondition,postecondition,duree,mo
de,donnee,typedonnee,role,activite)>
<!ATTLIST Tache id CDATA #REQUIRED>
<!ATTLIST Tache X CDATA #REQUIRED>
<!ATTLIST Tache Y CDATA #REQUIRED>
<!ATTLIST Tache W CDATA #REQUIRED>
<!ATTLIST Tache H CDATA #REQUIRED>
<!ELEMENT ListeTaches (Tache)+>
<!ATTLIST ListeTaches nbtaches CDATA #REQUIRED>
<!ELEMENT nomtransition (#PCDATA)>
<!ELEMENT typetransition (#PCDATA)>
<!ELEMENT dureetransition (#PCDATA)>
<!ELEMENT tachedebut (#PCDATA)>
<!ELEMENT tacheffin (#PCDATA)>
<!ELEMENT Transition
(nomtransition,typetransition,dureetransition,tachedebut,tacheffin)>
<!ATTLIST Transition id CDATA #REQUIRED>
<!ELEMENT ListeTransitions (Transition)+>
<!ATTLIST ListeTransitions nbtransitions CDATA #REQUIRED>
<!ELEMENT nomconnecteur (#PCDATA)>
<!ELEMENT typeconnecteur (#PCDATA)>
<!ELEMENT logiqueconnecteur (#PCDATA)>
<!ELEMENT tache cible (#PCDATA)>
<!ELEMENT transition (#PCDATA)>
<!ELEMENT listetransition (transition)+>
<!ATTLIST listetransition nbtransition CDATA #REQUIRED>

```

```

<!ELEMENT Connecteur
(nomconnecteur,typeconnecteur,logiqueconnecteur,tachecible,listetran
sition)>
<!ATTLIST Connecteur id CDATA #REQUIRED>
<!ATTLIST Connecteur X CDATA #REQUIRED>
<!ATTLIST Connecteur Y CDATA #REQUIRED>
<!ATTLIST Connecteur W CDATA #REQUIRED>
<!ATTLIST Connecteur H CDATA #REQUIRED>
<!ELEMENT ListeConnecteurs (Connecteur)*>
<!ATTLIST ListeConnecteurs nbconnecteurs CDATA #REQUIRED>
<!ELEMENT nomressource (#PCDATA)>
<!ELEMENT typeressource (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT tache (#PCDATA)>
<!ELEMENT listetache (tache)>
<!ATTLIST listetache nbtache CDATA #REQUIRED>
<!ELEMENT Ressource (nomressource,typeressource,url,listetache)>
<!ATTLIST Ressource id CDATA #REQUIRED>
<!ATTLIST Ressource X CDATA #REQUIRED>
<!ATTLIST Ressource Y CDATA #REQUIRED>
<!ATTLIST Ressource W CDATA #REQUIRED>
<!ATTLIST Ressource H CDATA #REQUIRED>
<!ELEMENT ListeRessource (Ressource)*>
<!ATTLIST ListeRessource nbressource CDATA #REQUIRED>
<!ELEMENT MaxLimite EMPTY>
<!ATTLIST MaxLimite macteur CDATA #REQUIRED>
<!ATTLIST MaxLimite mactivite CDATA #REQUIRED>
<!ATTLIST MaxLimite mrole CDATA #REQUIRED>
<!ATTLIST MaxLimite mtache CDATA #REQUIRED>
<!ATTLIST MaxLimite mtransition CDATA #REQUIRED>
<!ATTLIST MaxLimite mconnecteur CDATA #REQUIRED>
<!ATTLIST MaxLimite mressource CDATA #REQUIRED>
<!ELEMENT workflow
(ProprieteProcessus,ListeActeurs,ListeActivites,ListeRoles,ListeTach
es,ListeTransitions,ListeConnecteurs,ListeRessource,MaxLimite)>

```

Figure III.17 : Le DTD du méta modèle.

#### 5.2.2.8. Traitement d'un document XML

XML n'est pas un langage de programmation mais un méta langage permettant de représenter des données. Pour traiter ces données il faut disposer d'un parseur, encore appelé analyseur en bon français [Grad, 02].

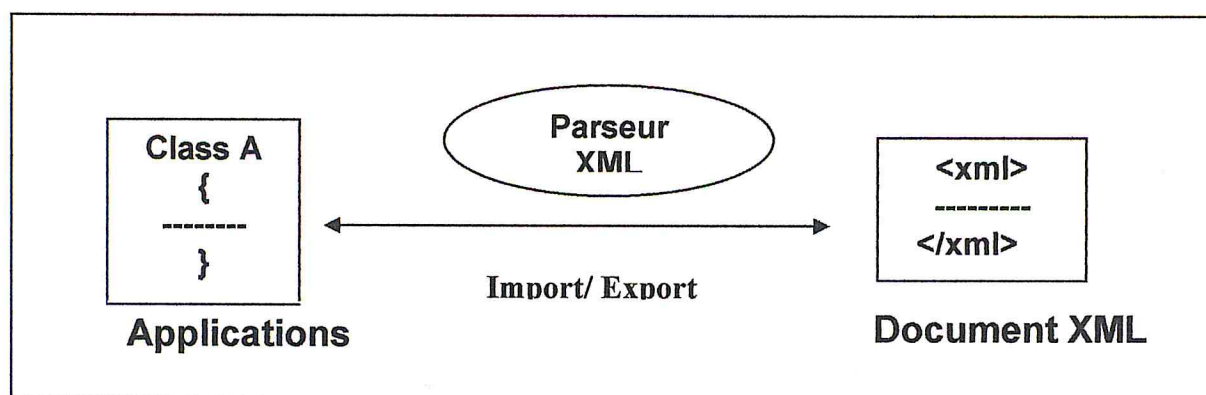
L'analyseur est un outil permettant de parcourir un document et d'en extraire des informations. Dans le cadre XML on parle de parseur. Ce dernier permet de créer une structure hiérarchique contenant les données contenues dans le document XML.

On distingue deux types de parseurs XML [BECH, 03]:

- Les parseurs validant (validating) permettant de vérifier qu'un document XML est conforme à sa DTD.
- Les parseurs non validant (non-validating) se contenant de vérifier que le document XML est bien formé (c'est-à-dire respectant la syntaxe XML de base).

La deuxième étape de notre travail consiste à générer un fichier XML, nous optons le deuxième type de parseur "validant", pour générer XML validé représente bien un workflow.

Un parseur XML est un outil (package, librairie, programme...) permettant d'accéder, de créer et de mettre à jour des documents XML. C'est le lien indispensable entre les applications et les documents XML.



**Figure III.18** : Le parseur XML entre une application et un document XML.

Pour analyser un document, les parseurs utilisent actuellement deux types d'approches [BECH, 03]:

- **Les API (Application Programming Interface) basés sur un traitement hiérarchique** : les analyseurs construisent une structure hiérarchique contenant des objets représentant les éléments du documents, et dont les méthodes permettant d'accéder aux propriétés. La principale API utilisant cette approche est DOM (Document Object Model).

DOM (Document Object Model) est une spécification du W3C (Word Wide Web consortium), définissent la structure d'un document sous forme d'une hiérarchie d'objets, afin de simplifier l'accès aux éléments constitutifs du documents [Bech, 03].



- **Les API basés sur un mode évènementiel** : permettant de réagir à des événements (comme le début d'un élément, la fin d'un élément) et de renvoyer le résultat à l'application utilisant cette API.

SAX (Simple API for XML) est la principale interface utilisant l'aspect évènementiel. SAX est une API basée sur un modèle évènementiel, cela signifie que SAX permet de déclencher des événements au cours de l'analyse du document XML, Une application utilisant SAX implémente généralement des gestionnaires d'événements, lui permettant d'effectuer des opérations selon le type d'élément rencontré [Ross, 02].

SAX n'est pas toujours facile d'emploi ou l'accès est séquentiel, l'API n'a aucune mémoire : impossible de connaître le contexte d'un évènement, donc il faut [Ross] :

- Avoir un modèle de l'analyse (par exemple sous forme d'un automate).
- Gérer à la main la mémorisation des informations importantes.

DOM <sup>permet</sup> de transformer un document XML en un arbre et fournit des API pour tout les objets (neuds) de l'arbre, ces API permettent a une application de traiter ces objets. Pour cette raison nous avons utilisé les API DOM pour traiter un document XML.

Le document XML est lu dans le parseur : celui-ci expose le document XML comme un arbre DOM. La représentation DOM reste en mémoire tant que l'application veut travailler avec ce document. Une fois que vous avez terminé avec le document, celui-ci peut persister sous une forme ou sous une autre [Will, 00].



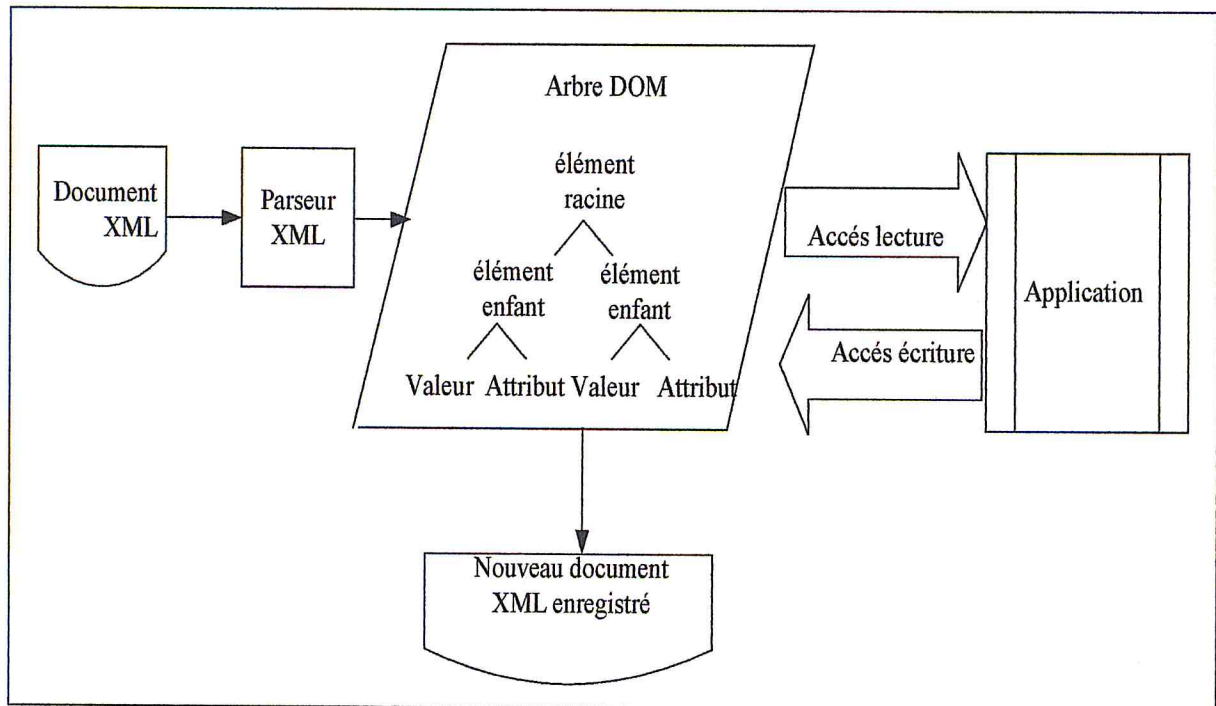


Figure III.19 : Traitement d'un document XML avec le parseur XML DOM [Will, 00].

### 5.2.3. Module 02 : Echange

Ce module assure la communication entre le module modélisation et génération, il fournit des API permettre d'accéder au API de DOM pour utiliser ses interfaces dans la génération du fichier XML.

#### 5.2.3.1. Connexion de l'application avec les API DOM

L'implémentation du DOM crée en mémoire un arbre représentant le document sous forme d'objets. Ces objets, ou noeuds, dispose un ensemble d'interfaces : la spécification DOM indique ce que sont ces interfaces et ce que vous pouvez en appelant sur celles-ci une méthode ou une propriété. Ainsi puisqu'il s'agit de programmation, vous manipulez les objets à travers les interfaces.

##### 5.2.3.1.1. Les interfaces DOM

DOM décrit un document sous la forme d'un arbre d'objets (noeud) comme nous montre la figure III.20. Pour mémoire, un arbre défini comme un ensemble d'objets interconnectés, c'est-à-dire des neouds, l'un de ces noeuds constituant la racine.

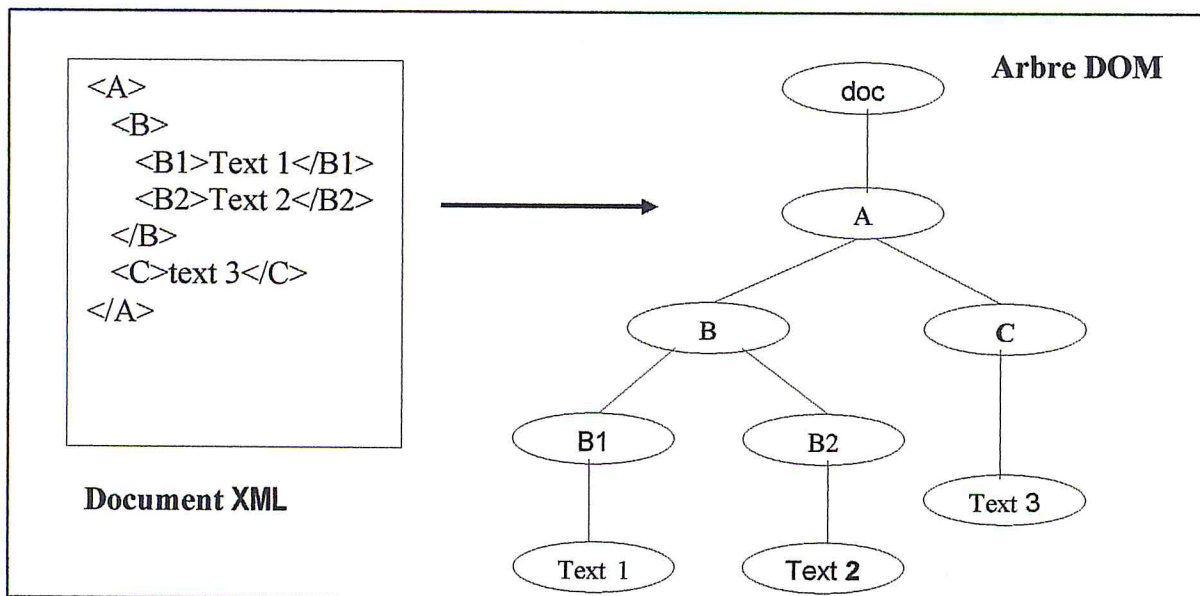


Figure III.20 : Création de l'arbre DOM.

DOM est une interface de programmation (API: Application Programming Interface), permettant à une application de parcourir la structure du document et d'agir dynamiquement sur celle-ci. Ainsi un langage de programmation utilise DOM pour naviguer au sein d'un document XML.

L'API DOM définit les interfaces suivantes :

- **Node** : offre les propriétés de base communes à tous les nœuds d'un document et relatives à sa structure d'arbre.
- **Document** : donne l'accès aux propriétés de la racine du document XML et propose les méthodes nécessaire à la création de nouveau nœud dans le document.
- **Élément** : représente un élément XML et permet d'en gérer les attributs et les contenu.
- **Att** : représente un attribut d'un élément XML.
- **NodeList** : représente un point d'accès dynamique à une collection ordonnée de nœuds XML.

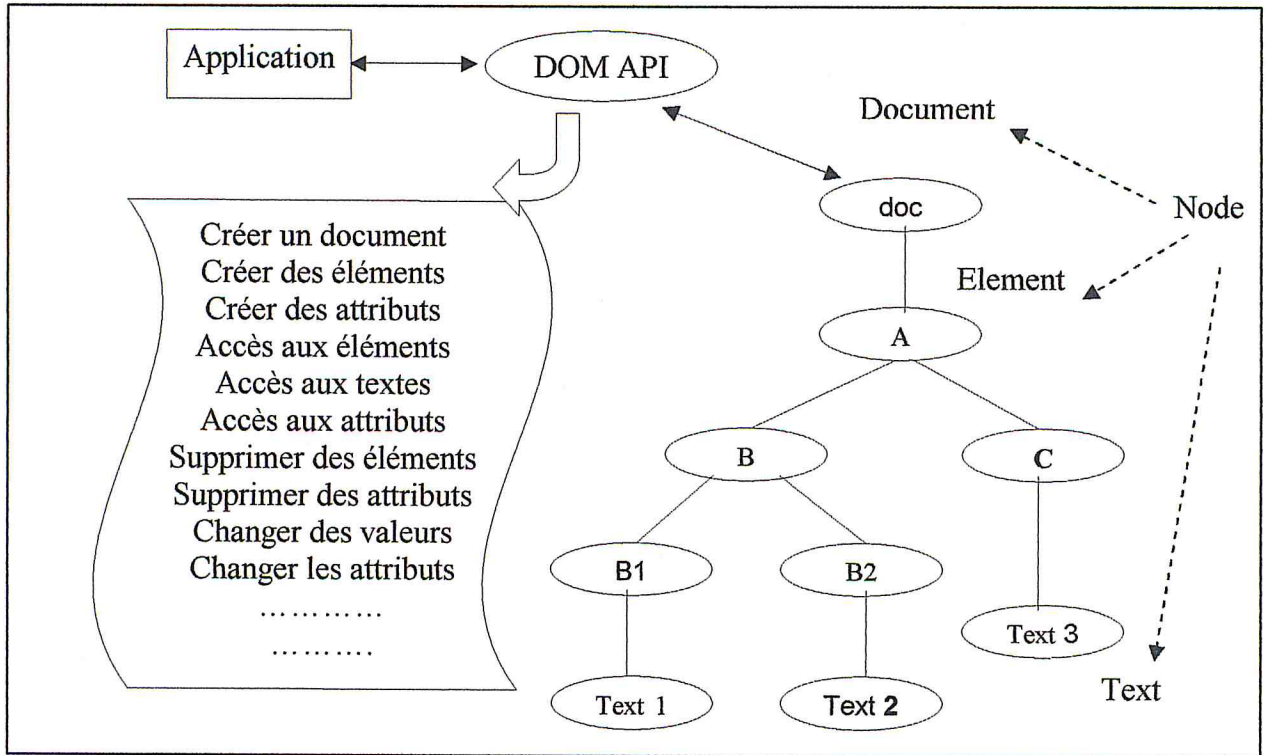


Figure III. 21 : Les interfaces DOM.

Le diagramme de composant suivant représente les différents composants qui interviennent à la communication entre les deux modules : la génération et la modélisation, en utilisant les interfaces DOM.

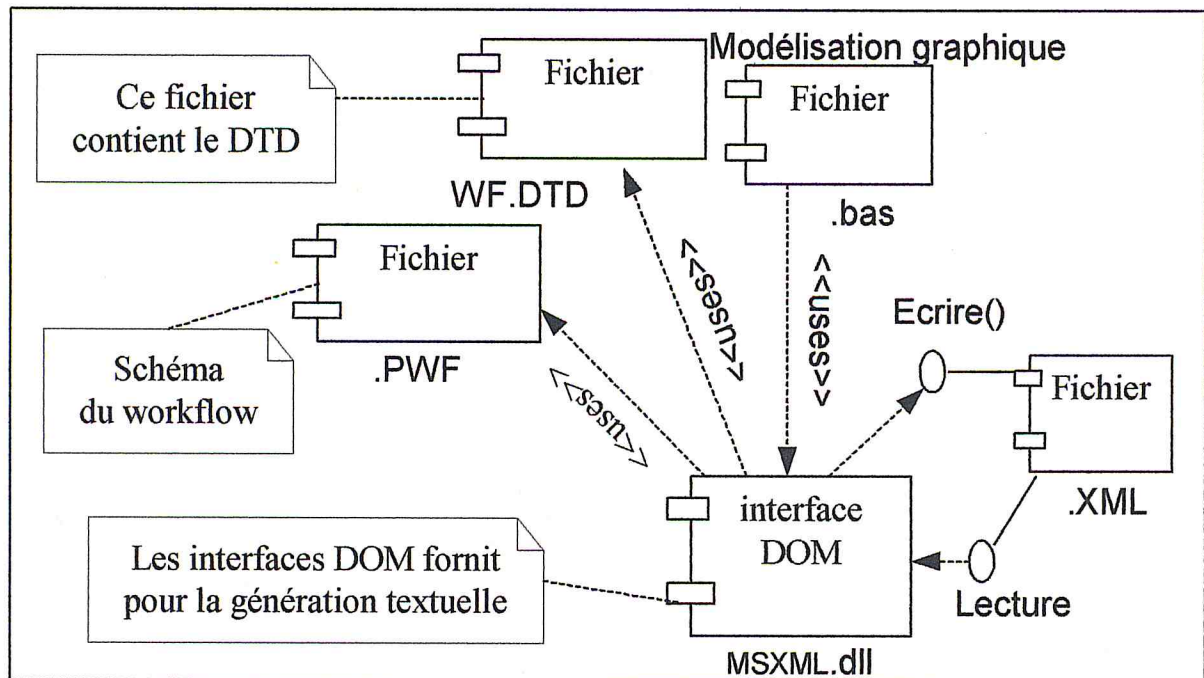


Figure III. 22 : Le diagramme de composants du système.

## 6. L'implémentation

Dans cette phase ce fait la traduction des algorithmes définis dans la partie conception dans un langage de programmation.

### 6.1. Choix du langage de programmation

La dernière étape de la réalisation d'un logiciel est le choix du langage de programmation. Après le choix du parseur XML et les API DOM nous pouvons choisir le langage de programmation pour développer notre outil de définition d'un workflow. Nous avons choisis parmi les langages de programmation le visual basic, par ce qu'il est un langage orienté objet et les API DOM sont des composants intégrés dans ce langage.

### 6.2. Création d'un fichier XML

Pour la création d'un fichier XML nous avons respecté les règles de passage de UML vers XML décrit précédemment. Dans cette partie nous spécifions quelques éléments du méta modèle par le langage XML.

- **Tâche** : une tâche est décrit par la spécification XML suivant :

```
<!ELEMENT Tache
(nomtache, typetache, description, precondition, postecondition, duree, mo
de, donnee, typedonnee, role, activite)>
<!ATTLIST Tache id CDATA #REQUIRED>
<!ELEMENT nomtache (#PCDATA)>
<!ELEMENT descreption (#PCDATA)>
<!ELEMENT precondition (#PCDATA)>
<!ELEMENT postcondition (#PCDATA)>
<!ELEMENT typetache (#PCDATA)>
<!ELEMENT activité (#PCDATA)>
<!ELEMENT role (#PCDATA)>
<!ELEMENT duree (#PCDATA)>
<!ELEMENT mode (#PCDATA)>
<!ELEMENT donnee (#PCDATA)>
<!ELEMENT typedonnee (#PCDATA)>
```

**Figure III.23** : Définition d'une tâche.

- **Rôle** : un rôle est décrit par la spécification XML suivant :

```

<!ELEMENT Role (nomrole, chefrole, listeacteur)>
<!ATTLIST Role id CDATA #REQUIRED>
<!ELEMENT nomrole (#PCDATA)>
<!ELEMENT chefrole (#PCDATA)>
<!ELEMENT listeacteur (acteur)+>

```

**Figure III.24 : Définition d'un rôle.**

Chaque rôle a un identifiant unique et un acteur qu'est le chef de rôle.

- **Acteur** : Un acteur est caractérisée par un identifiant unique et le type d'acteur, système ou humain. En XML, un acteur est défini comme suit :

```

<!ELEMENT Acteur (nomacteur, typeacteur)>
<!ATTLIST Acteur id CDATA #REQUIRED>
<!ELEMENT nomacteur (#PCDATA)>
<!ELEMENT typeacteur (#PCDATA)>

```

**Figure III.25 : Définition d'un acteur.**

- **Connecteur** : L'entité connecteur est composé des éléments utilisé pour représenter les transitions de plusieurs sorties ou plusieurs entrées. En XML, un connecteur est défini comme suit :

```

<!ELEMENT Connecteur
(nomconnecteur, typeconnecteur, logiqueconnecteur, tache cible, listetrans
ition)>
<!ATTLIST Connecteur id CDATA #REQUIRED>
<!ELEMENT nomconnecteur (#PCDATA)>
<!ELEMENT typeconnecteur (#PCDATA)>
<!ELEMENT logiqueconnecteur (#PCDATA)>
<!ELEMENT tache cible (#PCDATA)>
<!ELEMENT transition (#PCDATA)>
<!ELEMENT listetransition (transition)+>
<!ATTLIST listetransition nbtransition CDATA #REQUIRED>

```

**Figure III.26: Définition d'un connecteur.**

- **Transition** : une transition représente une lien entre les tâches et exprime l'ordre d'exécution des tâches. En XML, une transition est défini comme suit :

```

<!ELEMENT Transition
(nomtransition, typetransition, dureetransition, tachedebut, tache fin)>
<!ATTLIST Transition id CDATA #REQUIRED>
<!ELEMENT nomtransition (#PCDATA)>
<!ELEMENT typetransition (#PCDATA)>
<!ELEMENT dureetransition (#PCDATA)>
<!ELEMENT tachedebut (#PCDATA)>
<!ELEMENT tache fin (#PCDATA)>

```

**Figure III.27 : Définition d'une transition.**

### **6.3. Les fonctions de l'outil de définition**

L'outil de définition couvre trois étapes :

1. L'expert modélise un processus par l'utilisation des objets graphique fournit par L'outil de définition, le résultat de cette étape est un graphe.
2. Après la modélisation graphique du processus, la génération textuelle correspond est généré par l'outil de définition.
3. Le fichier XML généré dans l'étape deux est exécuté par un moteur de workflow.

Dans notre travail nous avons réalisé la première et la deuxième étape, la troisième étape est dépasse notre travail.

#### **6.3.1. La modélisation graphique : Etape 1**

Dans notre logiciel nous avons défini un éditeur graphique comprend tous les notations graphique correspondre aux éléments du méta modèle, ce éditeur permet aux experts de modéliser simplement un processus.

La figure III.28 représente l'interface principale de notre logiciel qu'est composé de :

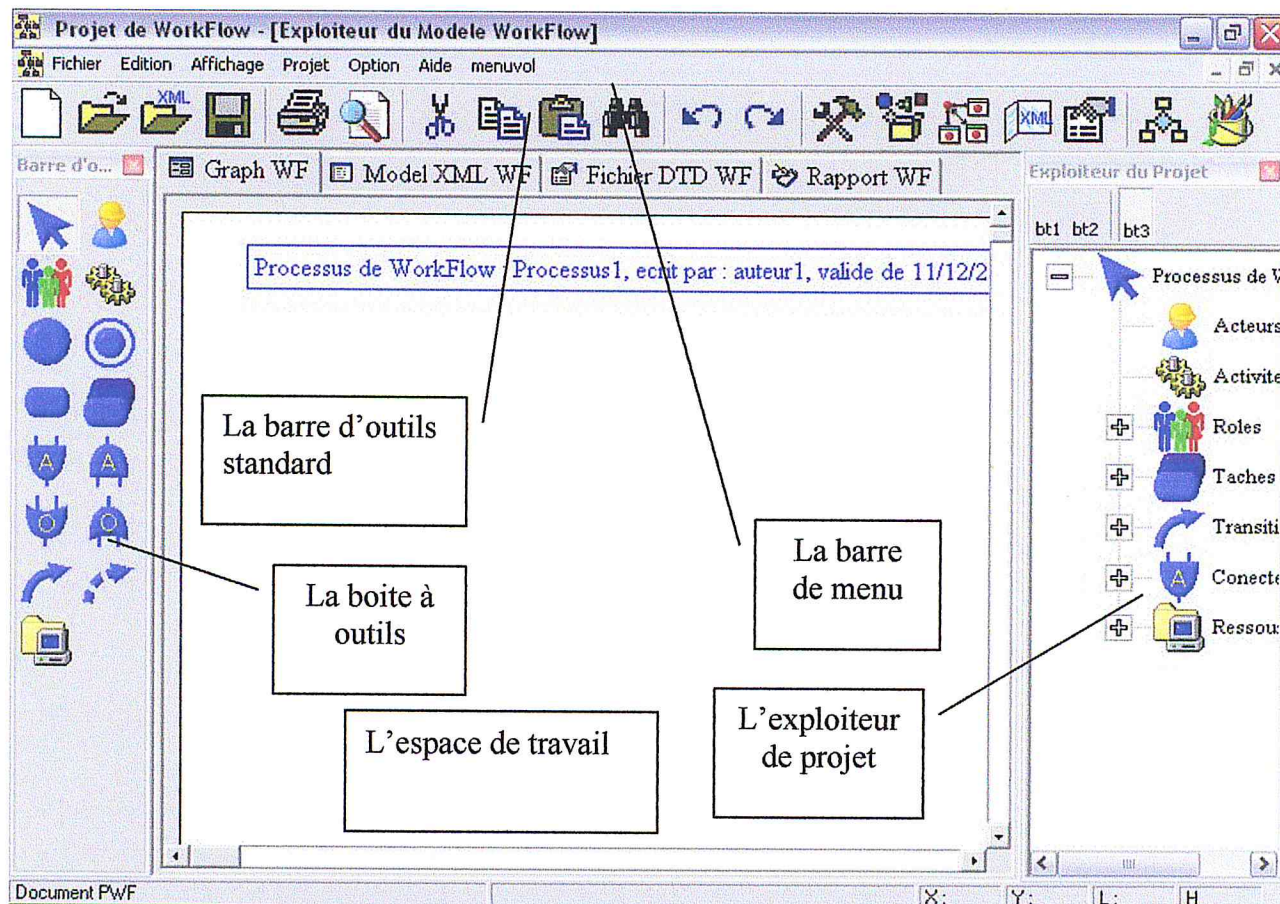


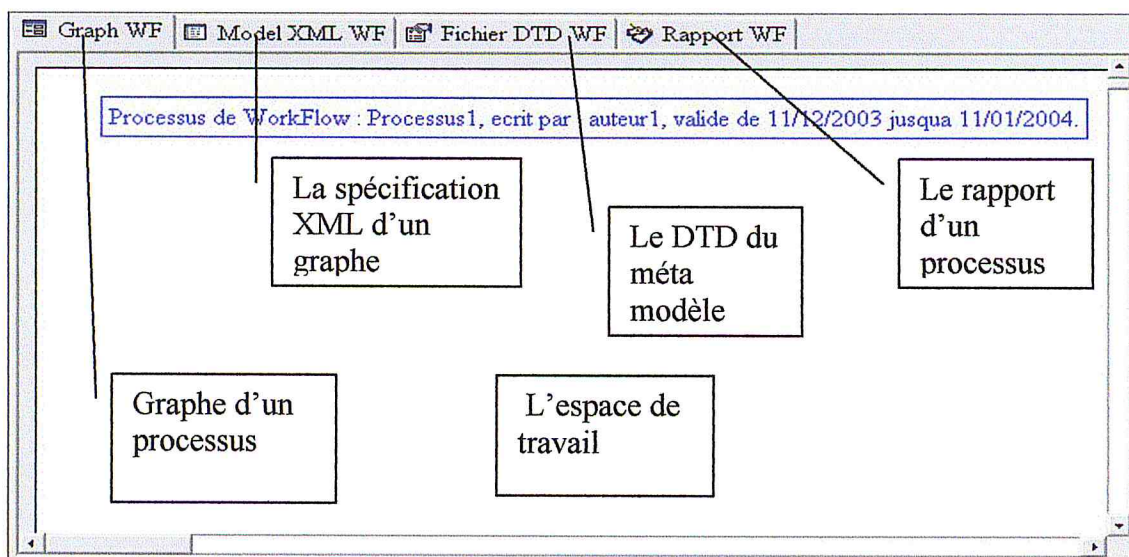
Figure III.28 : L'interface principale de L'outil de définition de workflow.

### 6.3.1.1. L'espace de travail

L'utilisateur utilise cette espace pour décrie graphiquement les élément (tâche, transition, connecteur..) d'un processus a l'aide de la boite à outils.

La figure III.29 représente un exemple d'un modèle du processus. Pour représenter un élément il faut d'abord sélectionner l'élément dans boite à outils, puis cliqué dans l'espace de travail, une boite de dialogue sera afficher, l'utilisateur remplir les propriétés et cliqué sur OK, l'élément s'affichera dans l'espace de travail.



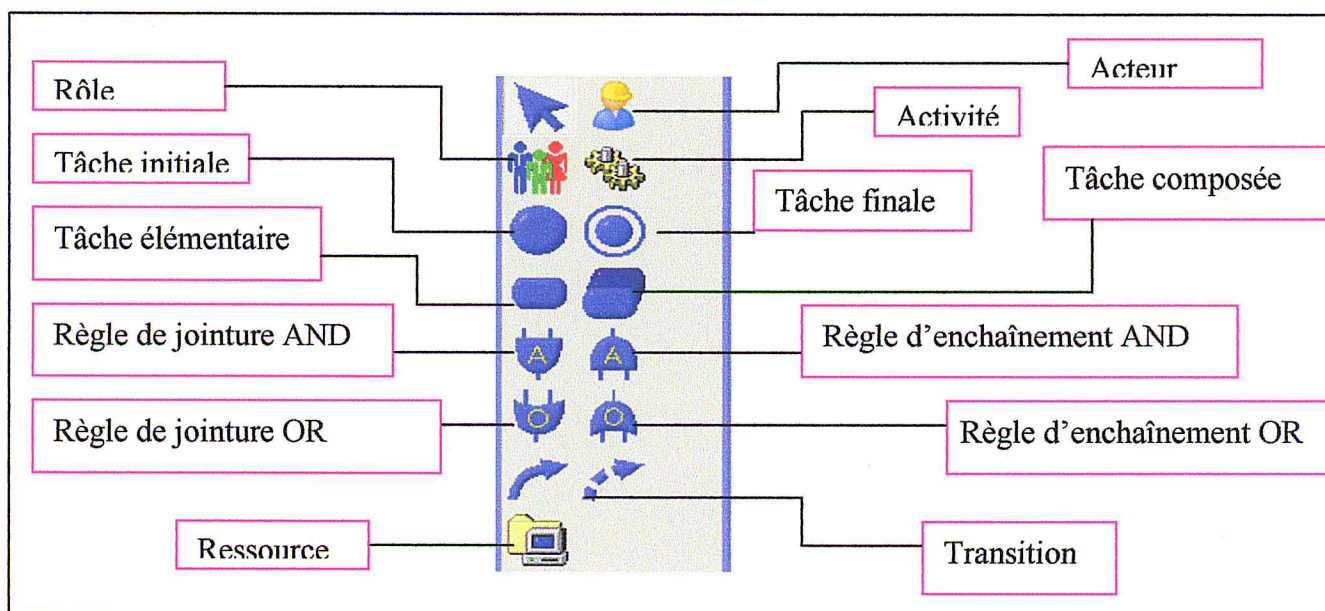


**Figure III.29:** L'environnement de l'espace de travail.

L'expert peut appliquer des fonctions (copier, supprimer, coller,...) sur les objets d'un processus par l'utilisation de la barre de menu et la barre d'outil.

### 6.3.1.2. La boîte à outil

La boîte à outil regroupe toutes les notations graphiques utilisées pour concevoir un processus. La figure III.30 représente la boîte à outil de notre logiciel.



**Figure III.30:** La boîte à outil.

### 6.3.1.3. L'exploiteur de projet

L'exploiteur de projet est affiché par le déroulement du menu projet puis cliqué sur l'exploiteur de projet. L'exploiteur de projet permet à l'utilisateur de visualiser le contenu d'un processus, les objets de modélisation sont apparaitent de façon ordonnée.

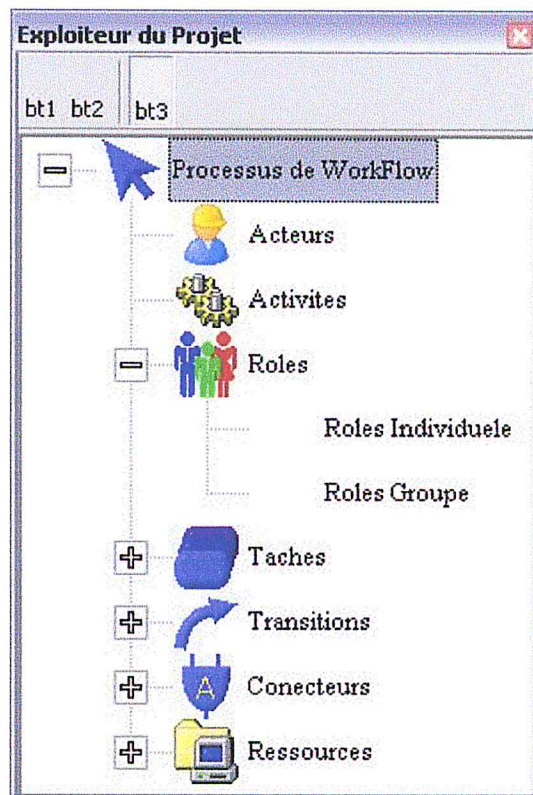


Figure III.31: L'exploiteur de projet.

### 6. 3.2. La génération de la définition

Notre outil de définition offre une fonction de génération du fichier XML, cette génération correspond à un processus workflow. Après la modélisation graphique d'un processus par un expert, l'outil de définition lui génère le fichier XML. La figure suivant illustre la procédure de génération.

Pour voir le code XML d'un graphe (processus worflow), il suffit de cliquer sur code XML, comme nous montre la figure III.32.

```

    <Acteur id="2">
      <nomacteur>Acteur2</nomacteur>
      <typeacteur>Humain</typeacteur>
    </Acteur>
    <Acteur id="3">
      <nomacteur>Acteur3</nomacteur>
      <typeacteur>Humain</typeacteur>
    </Acteur>
    <Acteur id="4">
      <nomacteur>Acteur4</nomacteur>
      <typeacteur>Humain</typeacteur>
    </Acteur>
    <Acteur id="5">
      <nomacteur>Acteur5</nomacteur>
      <typeacteur>Humain</typeacteur>
    </Acteur>
    <Acteur id="6">
      <nomacteur>Acteur6</nomacteur>
      <typeacteur>Humain</typeacteur>
    </Acteur>
  </ListeActeurs>
  <ListeActivites>
    <Activite id="1">
      <nomactivite>AnalyseEtDecision</nomactivite>
      <typeactivite>Controle</typeactivite>
    </Activite>
    <Activite id="2">
      <nomactivite>EtudeMarche</nomactivite>
      <typeactivite>Controle</typeactivite>
    </Activite>
    <Activite id="3">
      <nomactivite>EvaluationConcurrence</nomactivite>
      <typeactivite>Controle</typeactivite>
    </Activite>
    <Activite id="4">
      <nomactivite>DeveloppementPrototype</nomactivite>
      <typeactivite>Controle</typeactivite>
    </Activite>
  </ListeActivites>
  </Workflow>
  
```

Figure III.32: La génération du fichier XML.

L'utilisateur peut voir le code XML d'un concept de processus workflow, comme nous montre la figure suivante :

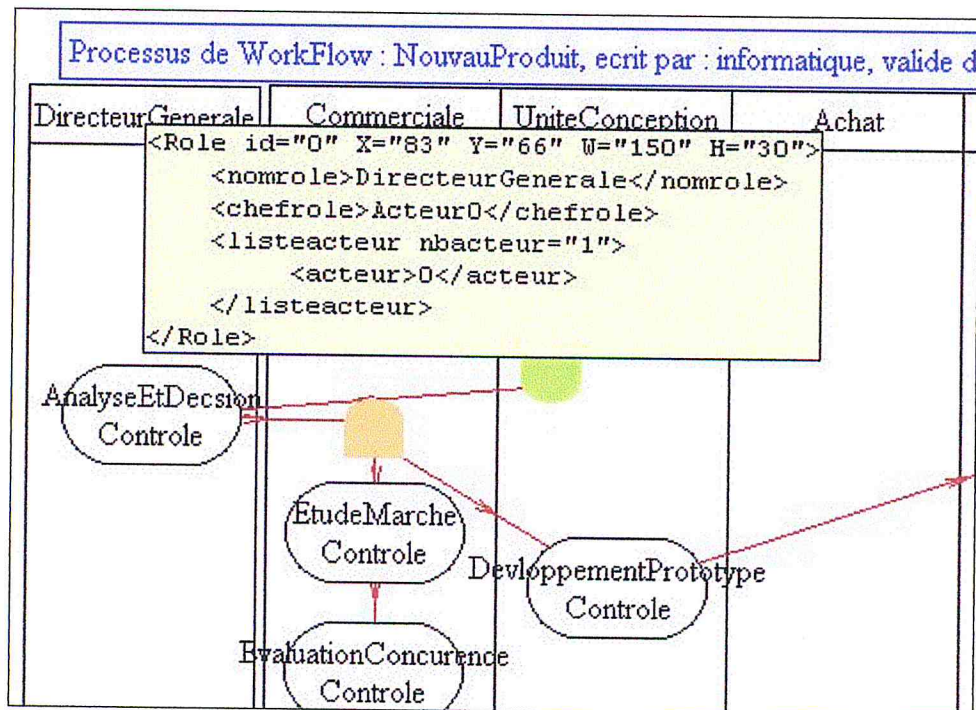


Figure III.33: La code XML d'un concept du processus workflow.

## 6.4. Le rapport du processus workflow

Le rapport du processus, contient une description détailler d'un processus workflow. Les erreurs qui sont fait par l'expert dans la modélisation, sont enregistrées dans le rapport. La figure suivante illustre un exemple d'un rapport :

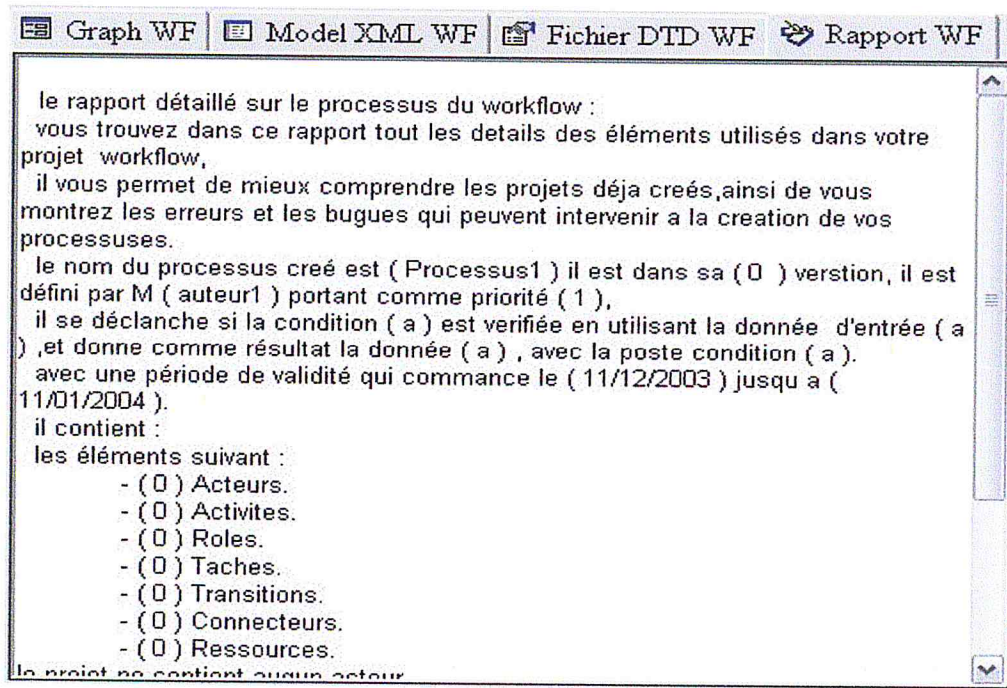


Figure III.34: Le rapport d'un processus workflow.

## 7. Test et validation

### 7.1. Le test

Le test permet de réaliser des contrôles pour la qualité du système, il s'agit de relever les eventiels de conception et de programmation [Som, 88].

Une suite de tests adéquats permettra de dire si une implantation est *conforme* à la spécification. La traçabilité permet notamment d'assurer une certaine couverture de la spécification : Chaque fonctionnalité du système devra être testée, au moins partiellement, en liant au moins un cas de test à l'expression du besoin correspondant [Guen, 01]. Le test permet de contrôler la qualité du système.

La validation consiste à évaluer l'adéquation du système développé vis-à-vis des besoins exprimés par ses futurs utilisateurs. La validation répond à la question "*is the right system being built?*" [Guen, 01].

### **7.1.1. Exemple d'un cas**

Nous illustrons notre représentation d'un processus par un exemple, qui est le développement d'un nouveau produit dans une entreprise, le type de ce workflow est ad-hoc (rarement étudié).

Pour mettre au point un tel produit et le lancer sur le marché pour en faire une réussite commerciale, de nombreuses entreprises suivent un processus structuré d'analyse et de planification pour transformer une nouvelle idée en un nouveau produit. Ce processus est le processus de développement du nouveau produit, qui comprend des « stades » au cours desquels le travail est effectué, et des « analyses » qui exigent l'évaluation des résultats et de l'avancement. Le graphique ci-dessous illustre le processus général de développement d'un nouveau produit.

### **7.1.2. Le modèle de processus**

Le modèle suivant est une représentation graphique du processus développement d'un nouveau produit dans une entreprise :

Le processus que nous présentés est composé de six rôles : *directeur générale, commerciale, unité conception, achat, vente, service production*. Chaque rôle est chargé d'exécuter un ensemble des tâches élémentaires ou composées.

### 7.1.3. La spécification du modèle

Le fichier texte présenté dans la figure suivante représente la spécification XML du processus de développement d'un nouveau produit dans une entreprise. Ce fichier est généré par le parseur DOM.

```

<workflow>
- <ListeActeurs nbacteurs= "7 ">
- <Acteurs id= "0 ">
  <nomActeur>Acteur0</nomActeur>
  <typeActeur>Humain</typeActeur>
  </Acteur>
+ <Acteurs id= "1 ">
+ <Acteurs id= "2 ">
+ <Acteurs id= "3 ">
+ <Acteurs id= "4">
+ <Acteurs id= "5">
+ <Acteurs id= "6">
  </ListeActeurs>
- <ListeActivite nbactivites= "3">
  <nomactivite>Activite0</nomactivite>
  </Activite>
+ <Activite id= "1">
+ <Activite id= "2">
  </ListeActivite>
- <ListeRole nbaroles= "6">
- <Role id= "0 " X= "88" Y="45" W="150" H="30">
  <nomrole> DirecteurGénérale</nomrole>
  <chefrole>Acteur0</chefrole>
  <listeacteur nbacteur = "1 ">
  <acteur>0</acteur>
  </listeacteur>
  </Role>

```

Figure III.35 : Une partie de fichier XML du processus de développement d'un nouveau produit.

### 7.2. La validation des cas d'utilisation

Dans cette phase nous présentons les fonctionnalités de notre système, et les procédures de validation des quelques cas d'utilisations.

### 7.2.1. Validation et définition des propriétés d'un processus workflow

Au début de la création d'un nouveau processus workflow, il faut définir les propriétés : nom, type de processus, version du processus, auteur du processus, la description, les entrées et les sorties. Comme la montre la figure III.36.

Le scénario de définir un processus workflow est comme suit :

- 1- L'expert cliqué sur nouveau projet dans la barre d'outil standard, ou appart ire du menu fichier.
- 2- Le système lui affiche un boîte de dialogue processus worflow.
- 3- L'expert remplié les propriétés du processus et cliqué sur OK.

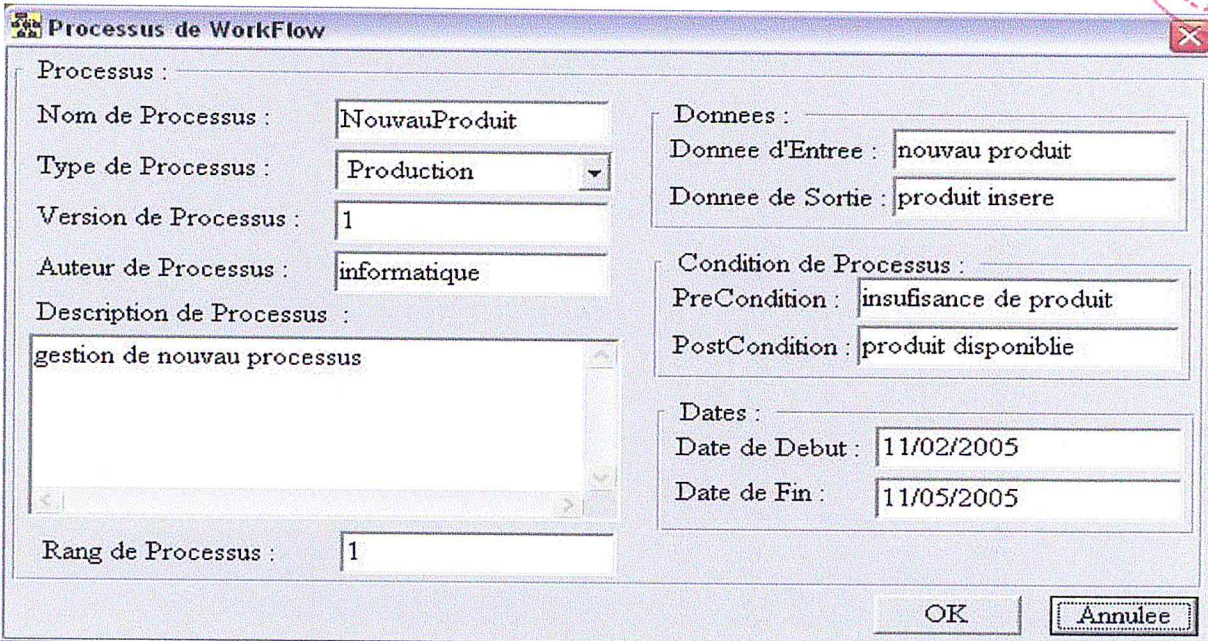


Figure III.36 : Définition et validation d'un processus workflow.

### 7.2.2. Validation du cas définition des acteurs

Au début de la modélisation d'un processus workflow, il faut définir tous les acteurs du système (figure III.37).

Le scénario de définir un acteur est comme suit :

- 1- L'expert cliqué sur l'icône de l'acteur, ou ajouté un acteur appart ire du menu projet.
- 2- Le système lui affiche un boîte de dialogue ajouter acteur.
- 3- L'expert remplié les propriétés de l'acteur et cliqué sur OK.

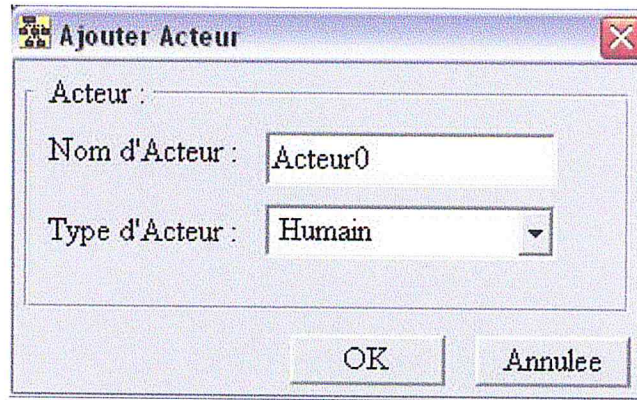


Figure III.37 : Définition des acteurs.

### 7.2.2.1. Validation de la définition des rôles

Après la définition des acteurs, l'expert doit définir les rôles (figure III.38). Le scénario de définition d'un rôle est comme suit :

- 1- L'expert cliqué sur l'icône du rôle, ou ajouté un rôle appart ire du menu projet.
- 2- Le système lui affiche un boite de dialogue ajouter rôle.
- 3- L'expert rempli les propriétés du rôle et cliqué sur OK.

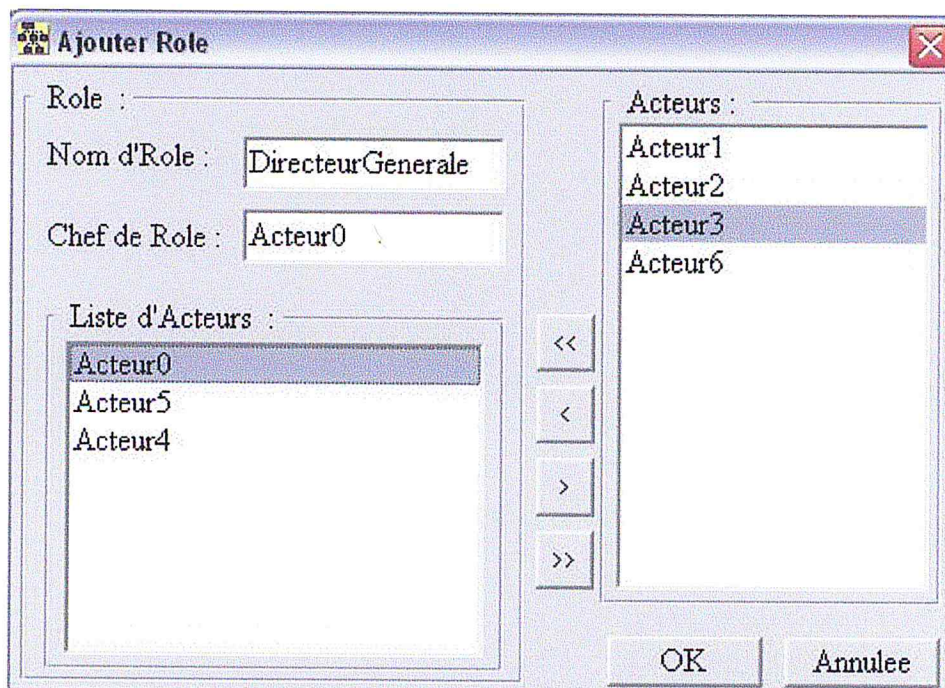


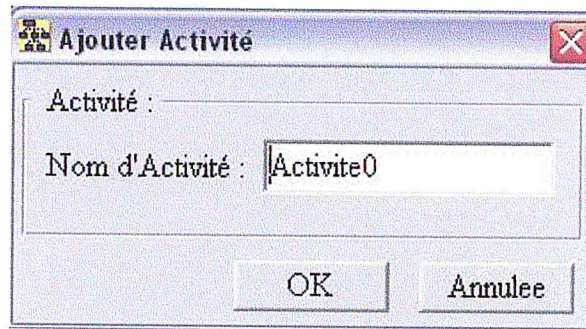
Figure III.38 : Définition d'un rôle.

### 7.2.3. Définition des activités

Après la définition des acteurs et des rôles, l'expert doit définir les activités (figure III.39). Le scénario de définition d'un activité est comme suit :



- 1- L'expert cliqué sur l'icône de l'activité dans la boîte à outil, ou ajouté une activité appartenant au menu projet.
- 2- Le système lui affiche une boîte de dialogue ajouter activité.
- 3- L'expert remplit les propriétés de l'activité et cliqué sur OK.



**Figure III.39** : Définition d'une activité.

#### 7.2.4. Définition des tâches

Le scénario de définir une tâche est comme suit :

- 1- L'expert cliqué sur l'icône de la tâche, ou ajouté une tâche appartenant au menu projet.
- 2- Le système lui affiche une boîte de dialogue ajouter tâche.
- 3- L'expert remplit les propriétés de la tâche et cliqué sur OK.

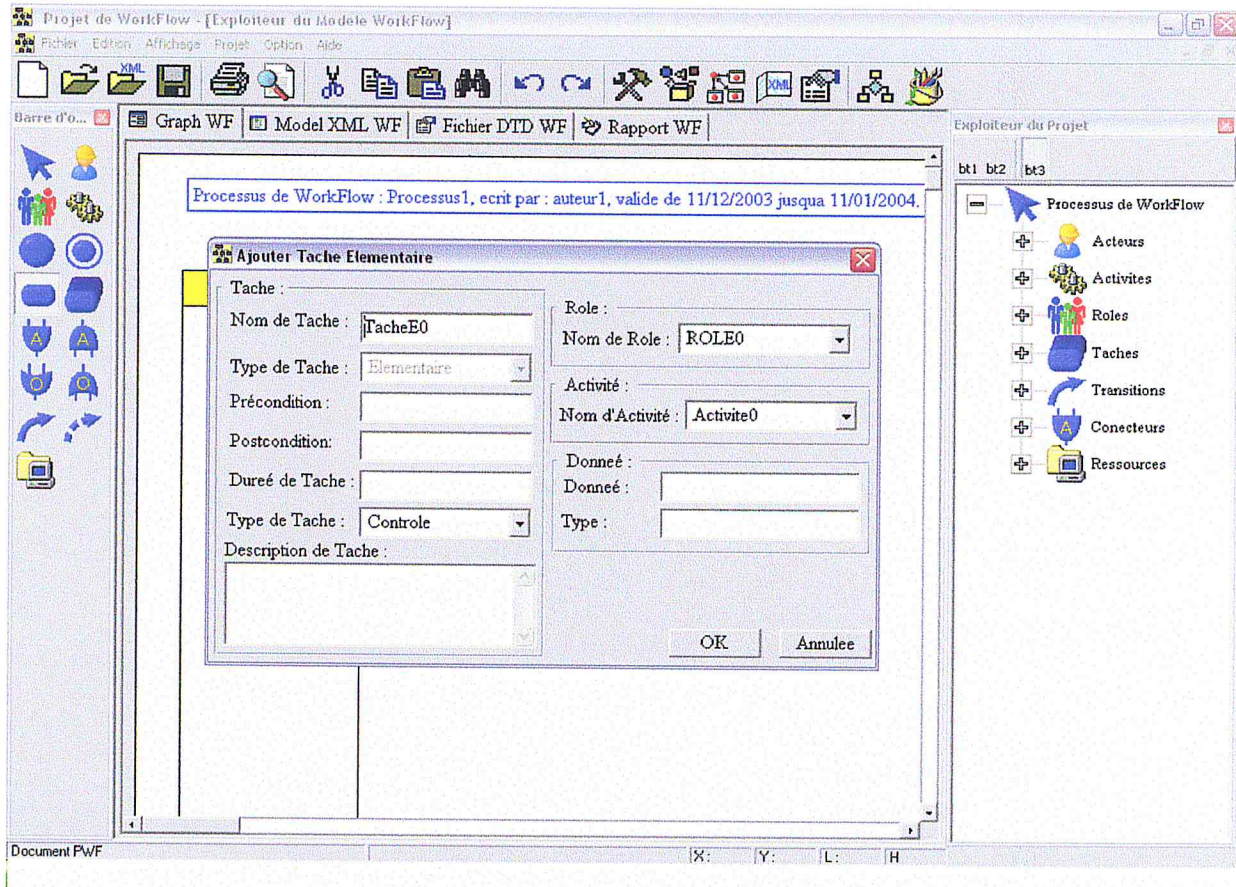


Figure III.40 : Définition d'une tâche.

### 7.2.5. Définition des transitions

Après la définition des tâches, L'expert définit les transitions existant entre les tâches. Le scénario de définir une transition est comme suit :

- 1- L'expert cliqué sur l'icône de la transition dans la boite à outil, ou ajouté une tâche appart ire du menu projet.
- 2- Le système lui affiche un boite de dialogue ajouter transition.
- 3- L'expert remplié les propriétés de la transition et cliqué sur OK.

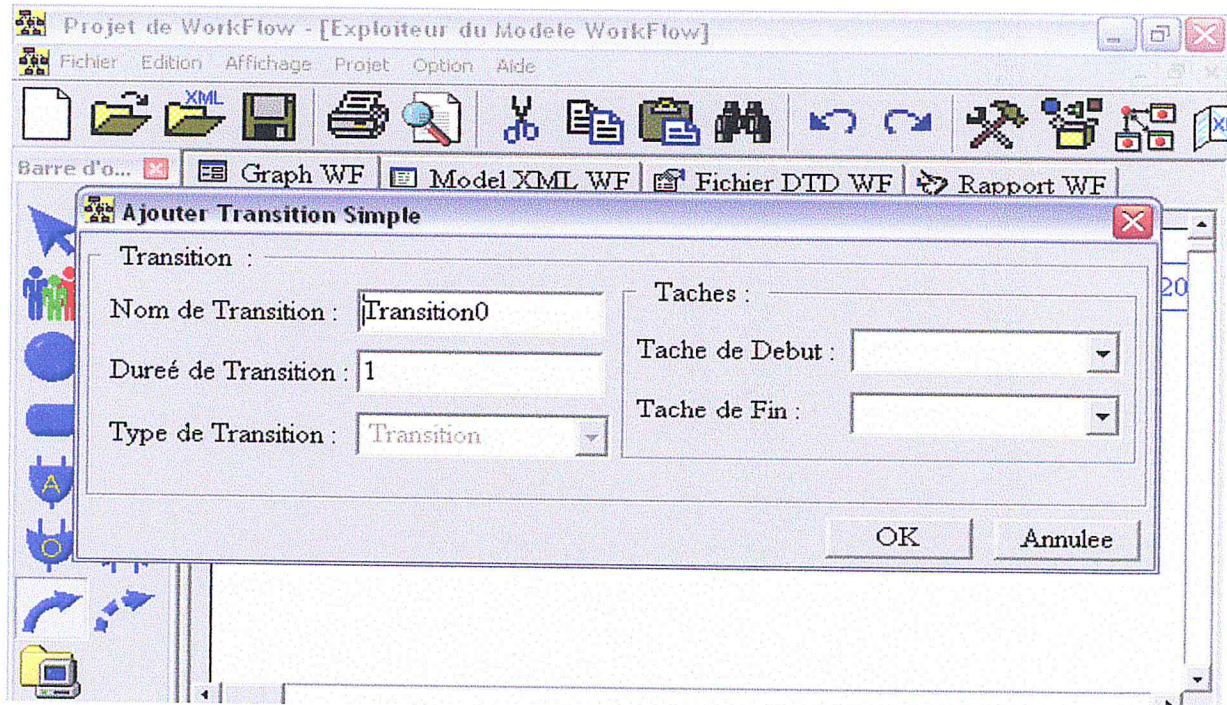


Figure III.41 : définition d'une transition.

### 7.2.6. Définition des ressources

Le scénario de définir une ressource est comme suit :

- 1- L'expert cliqué sur l'icône de la ressource, ou ajouté une tâche appartenant au menu projet.
- 2- Le système lui affiche un boite de dialogue ajouter ressource.
- 3- L'expert rempli les propriétés de la ressource et cliqué sur OK.

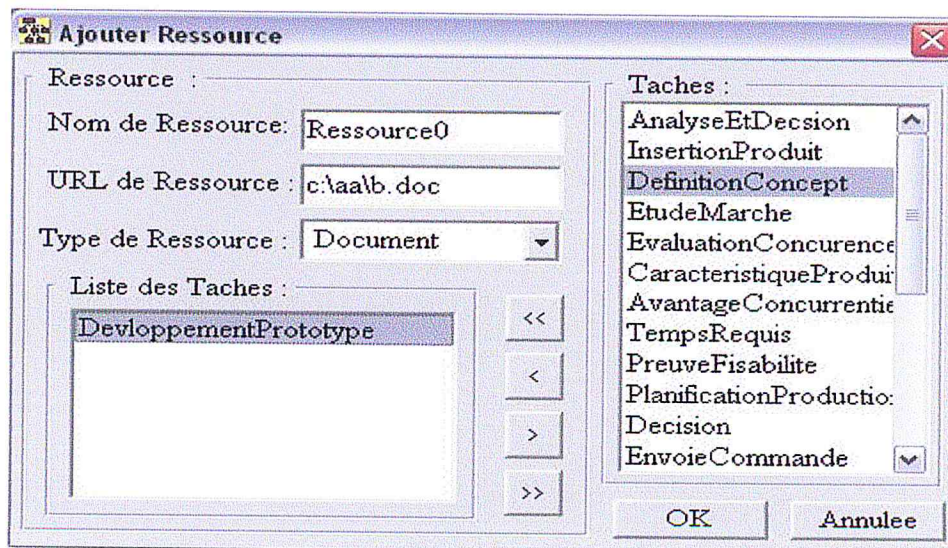


Figure III.42 : définition d'une ressource.

## **8. Conclusion**

Nous avons présenté dans ce chapitre le cycle vie de développement de notre outil de définition, en détaillons chaque étape de développement. Pour la modélisation du système nous avons utilisé la notation UML, qui dispose d'une notation graphique riche permettre de modéliser les objets métier du processus workflow. Nous avons utilisé le modèle en cascade pour représenter la démarche suivie pendant le cycle de développement associée.

Notre outil de définition permet a un expert qui doit être pas forcément un informaticien, de l'utiliser et permet de modéliser graphiquement un processus workflow et générer la définition associée à un format standard XML.



# Conclusion générale

Notre objectif dans cette mémoire est de développer un outil de définition d'un processus workflow permettre aux utilisateurs (experts) de modéliser graphiquement et générer une représentation textuelle (fichier XML) correspond a un processus workflow, ce schéma XML contient toutes les information du processus modélisé et compris leur structure graphique. La définition workflow sera considérée comme :

- a) Une spécification du workflow qui peut être exécuté par le moteur de workflow.
- b) Un cas de système de raisonnement à base de cas qui sera utilisé dans la deuxième étape « la sélection ».

L'objectif est de concevoir un logiciel qui correspond aux attentes des travailleurs de l'entreprise qui utilisent le workflow. Il doit être intuitif et robuste.

Pour la réalisation de ce projet, nous avons décidé d'utiliser la notation UML (Unified Modeling Language). Ce langage standard permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.

Pendant notre travail nous avons suivit le modèle en cascade, Il permet de construire, tester, installer et de fournir un support utilisateur. De plus, ce modèle est utilisable tout au long de la vie du produit.

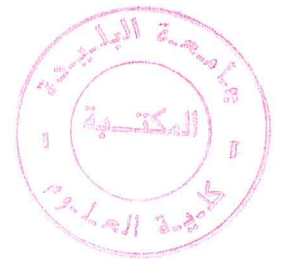
Nous avons interprété le méta modèle d'un processus workflow par le langage XML (Extensible Markup Language), et d'implémenter un parseur XML ou nous avons utilisé les interfaces DOM pour la génération textuelle et la validation d'un document XML par rapport au DTD du méta modèle.

Notre outil fournit aux experts des notations graphique permettant de modéliser graphiquement les éléments métier d'un processus workflow, générer une représentation textuelle qui peut être exécute par un moteur de workflow et de modéliser les processus sans l'intervention d'un informaticien.

L'avantage de notre outil de définition est que le logiciel est facile a utilisé, parce que nous avons mis tout les détaille concernant les composants logiciels, et permet de générer automatiquement la définition d'un processus workflow (fichier XML).

Le travail qu'on a réalisé peut conduire a des perspectives qui sont :

- 1- La réaliser d'un moteur de workflow qui interprète la définition d'un processus workflow (fichier XML).
- 2- Développement d'un système calcule la similarité entre un problème (shéma XML) avec les définitions workflow stocké dans la base de cas, afin de réaliser un système raisonnement à base de cas.
- 3- Développement d'un outil de définition sur l'intranet, permet aux utilisateurs de modéliser le même processus workflow.

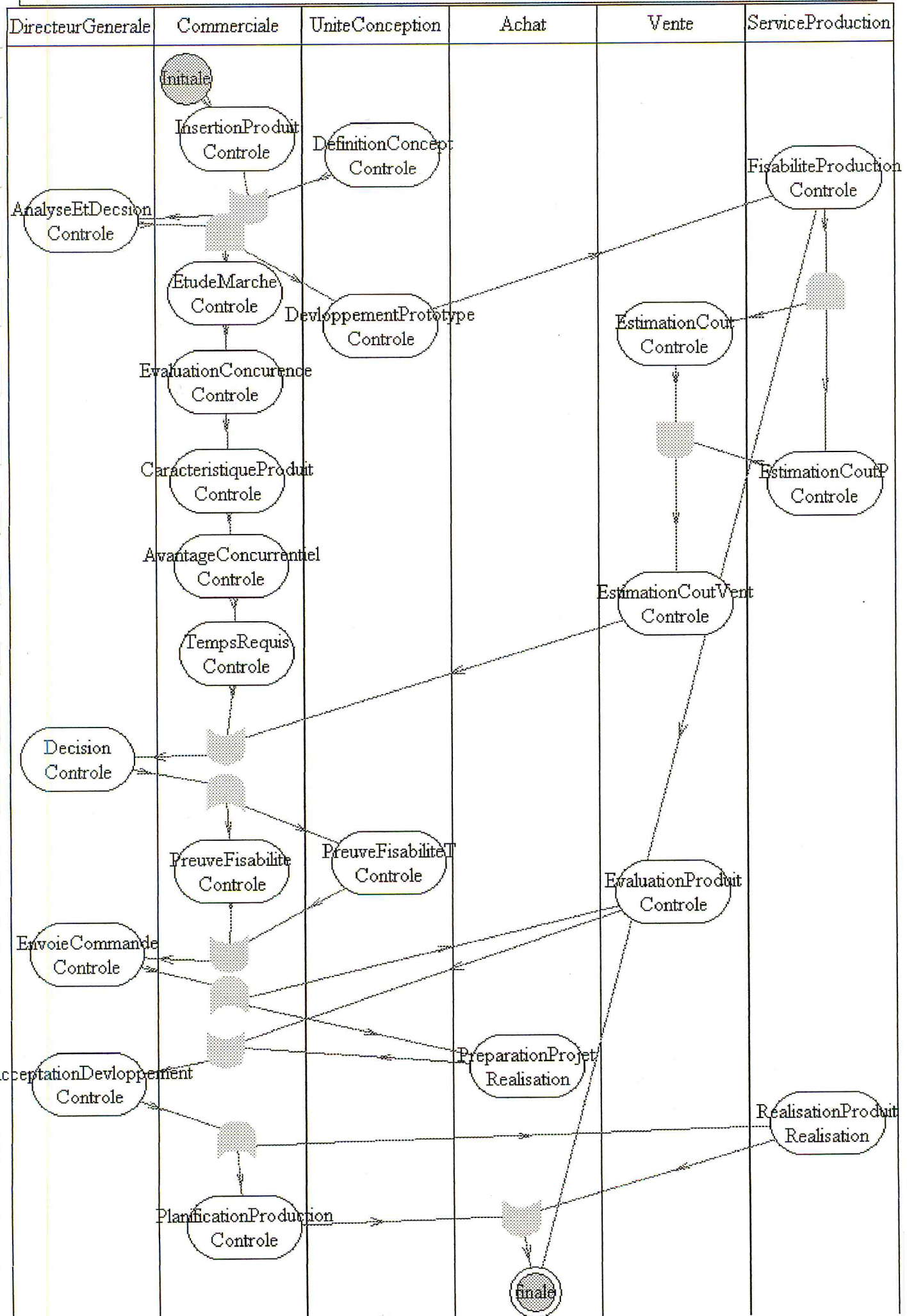


**Bibliographie**

- [Bech, 03] M. Bechchi « Conception et réalisation d'un système de recherche et d'indexation d'informations (SRII) » Mémoire de fin d'étude pour l'obtention d'ingénieur d'états en informatique, université mohamed V- Souissi, 2002/2003.
- [BOURES, 03] E. BOURES, «Cours de génie logiciel».
- [Claude, 96] M. Claude Gaudel « Précis de génie logiciel », professeur à l'université paris, MASSON 1996.
- [Clark, 98] J. Clark, S. Deach, Extensible Stylesheet Language (XSL) Version 1.0 - W3C Working Draft, num. WD-xsl-19980818, World Wide Web Consortium, August 1998. <http://www.w3.org/TR/WD-xsl>.
- [Dun, 00] R. Dindeleux – L. Guizzi Le management des processus : Au delà des contraintes normatives- De la définition des processus à l'automatisation dans le système d'information de l'entreprise, jitec, 2000.
- [Fred, 00] M .Frédéric «Workflow, groupware et systèmes à base de données, études des couvertures fonctionnelles, des critères de choix et des possibilités d'intégrations», Diplôme postgrade en informatique et organisation, université de lausanne, soutenue 2000-2001.
- [Frey, 99] FREY, «La gestion de documents à travers une application Workflow», diplôme post grade en informatique et organisation, université de lausanne, soutenue 1999-2000. Disponible à l'adresse : <http://inforge.unil.ch>.
- [Gard, 02] G. Gardarin « XML des bases de données aux services Web», DUNOD, 2002.

- [Glazman, 99] D. Glazman, CSS2: Feuilles de style HTML, Eyrolles, 1999.
- [LAURE, 02] E. Laure, «Mise en place d'un cycle de vie de développement logiciel », projet d'industriel 2002.
- [Levan, a99] Serge K Levan «Le projet Groupware» Eyrolles (Groupware), 1999.
- [Levan, b99] Serge K Levan «Le projet workflow» Eyrolles, 1999.
- [Mar, 02] A. Marie Hugues « Génie logiciel » 2002.
- [Marek, 98] S.khoshafiane, M. Buckiewicz , «Groupware et workflow», Masson, 1998.
- [Mélissa, 02] Dr. Mélissa Saadoun «Le Workflow Pour automatiser les procédures et tâches répétitives» Jeudi 24 octobre 2002 Demi-journée d'information au C.R.I.T.T. Pays de la Loire Productique.
- [Muller, 97] P. Alain Muler, « Modélisation objet avec UML», Eyrolles, 1997.
- [Patr, 98] P. ANDRIES. Langage de balisage extensible (XML) 1.0, disponible à l'adresse <http://www.w3.org/TR/1998/REC-xml-19980210>. Copyright © 1998 W3C (MIT, INRIA, Keio).
- [Pont, 00] M. PONTACQ Consultant et architecte Groupe Bull «Gestion des flux d'informations : workflow» Centre français d'exploitation.
- [Rivard, 03] F.Rivard, T.Plantain «L'EAI par pratique» Eyrolles,2003.
- [Rusty, 04] E. Rusty Harold «XML 1.1 Bible » 3rd Edition, Wiley Publishing, Inc, 2004.
- [Ross, 02] F. Rossi « Les API pour XML » Universit\_e Paris-IX Dauphine, disponible à l'adresse <http://apiacoa.org/contact.html>.
- [Sard, 99] E. SARDET «Intégration des approches modélisation conceptuelle







et structuration documentaire pour la saisie, la représentation, l'échange et l'exploitation d'informations. Application aux catalogues de composants industriels.» Pour l'obtention du grade de DOCTEUR DE L'UNIVERSITE DE POITIERS, Ecole Doctorale des Sciences pour l'Ingénieur, Soutenue le 4 Octobre 1999.

- [Sevig, 02] M. Sévigny, *AJLSM* « Introduction à XML » Notes de cours, 2002, disponible à l'adresse <http://www.ajlsm.com/formation/xml/>.
- [Som, 88] L. Smmerville, « Le génie logiciel et ses applications » Paris, 1988.
- [Van, 02] Wil. Van der Aalst and Kees van Hee «Workflow Management Models, Methods, and Systems» The MIT Press Cambridge, Massachusetts London, England 2002 Massachusetts Institute of Technology.
- [VIG, 03] N. VIGNER «-MEMOIRE-APPORTS D'UML DANS LA CONCEPTION ET REALISATIONDE NOUVELLES APPLICATIONS INFORMATIQUES» UNIVERSITE PARIS 7, DENIS DIDEROT, 2003.
- [Will, 00] K. Williams, M. Brundage, P. dancler, « XML et les bases de données », Eyrolles, 2000.
- [XML, 98] Langage de balisage extensible (XML) 1.0 Recommandation du W3C, 10 février 1998 disponible à l'adresse <http://www.w3.org/TR/1998/REC-xml-19980210>
- [XML, 02] Fiche de l'AWT Introduction au langage XML Créée le 09/05/01 Modifiée le 06/03/02. Disponible à l'adresse : [www.awt.be](http://www.awt.be), Agence Wallonne des Télécommunications.
- [XMLG, 00] G.Picadre «XML le guide de l'utilisateur», Elliotte Ruety Harold, 2000.

- [Glazman, 99] D. Glazman, CSS2: Feuilles de style HTML, Eyrolles, 1999.
- [LAURE, 02] E. Laure, «Mise en place d'un cycle de vie de développement logiciel », projet d'industriel 2002.
- [Levan, a99] Serge K Levan «Le projet Groupware» Eyrolles (Groupware), 1999.
- [Levan, b99] Serge K Levan «Le projet workflow» Eyrolles, 1999.
- [Mar, 02] A. Marie Hugues « Génie logiciel » 2002.
- [Marek, 98] S.khoshafiane, M. Buckiewicz , «Groupware et workflow», Masson, 1998.
- [Mélissa, 02] Dr. Mélissa Saadoun «Le Workflow Pour automatiser les procédures et tâches répétitives» Jeudi 24 octobre 2002 Demi-journée d'information au C.R.I.T.T. Pays de la Loire Productique.
- [Muller, 97] P. Alain Muler, « Modélisation objet avec UML», Eyrolles, 1997.
- [Patr, 98] P. ANDRIES. Langage de balisage extensible (XML) 1.0, disponible à l'adresse <http://www.w3.org/TR/1998/REC-xml-19980210>. Copyright © 1998 W3C (MIT, INRIA, Keio).
- [Pont, 00] M. PONTACQ Consultant et architecte Groupe Bull «Gestion des flux d'informations : workflow» Centre français d'exploitation.
- [Rivard, 03] F.Rivard, T.Plantain «L'EAI par pratique» Eyrolles,2003.
- [Rusty, 04] E. Rusty Harold «XML 1.1 Bible » 3rd Edition, Wiley Publishing, Inc, 2004.
- [Ross, 02] F. Rossi « Les API pour XML » Universit\_e Paris-IX Dauphine, disponible à l'adresse <http://apiacoa.org/contact.html>.
- [Sard, 99] E. SARDET «Intégration des approches modélisation conceptuelle

**Sites web :**

[CCM]            Disponible à l'adresse [www.commentcamarche.com](http://www.commentcamarche.com)

[MAY]            Disponible à l'adresse [www.mayetic.fr](http://www.mayetic.fr)

[uml.free]        Disponible à l'adresse [www.uml.free.com](http://www.uml.free.com)

## Réalisation d'un outil qui permet la modélisation graphique d'un processus Workflow et qui génère la spécification associée dans un schéma XML

**Résumé :** L'arrivée des nouveaux outils de communication a entraîné de nouvelles technologies, elles comprennent le groupware, dans cette technologie, le workflow fait partie des environnements les plus puissants pour automatiser les processus de travail. Le workflow relie les hommes, les applications, les processus, en gardant la trace de toutes les opérations intermédiaires entre le début et la fin d'une action.

*Le Système de gestion de workflow définit, implémente et gère l'exécution d'un ou plusieurs workflow à l'aide d'un environnement logiciel fonctionnant avec un ou plusieurs moteurs de workflow et capable d'interpréter la définition d'un processus, de gérer la coordination des participants et d'appeler des applications externes. Ce système est composé de deux composants : outil de définition de workflow, et le moteur. L'outil de définition est le cœur du système puisqu'il servira à formaliser la description des fonctions exercées en processus. Il permettra de définir également les données échangées, les interfaces avec les autres modules. Cet outil est confié à un expert métier qui définira et fera vivre le modèle de processus.*

L'outil de génération est utilisé pour générer la représentation du processus sous une forme exécutable par le système : texte, graphique ou langage formel.

Au cours de notre travail nous avons réalisé un outil de définition simple à utiliser, permettant aux usegers de représenter graphiquement leur métier, leurs activités, leurs enchainements et les règles associées, dans un formalisme fermé par le méta modèle correspondant au type d'activité modélisé.

