

UNIVERSITE POLYTECHNIQUE  
DE BOBO-DIOULASSO  
(U.P.B.)

ECOLE SUPERIEURE D'INFORMATIQUE  
(E.S.I.)

01 BP 1091 BOBO-DIOULASSO 01  
Tel : (226) 97-27-64 Fax : 98-25-77

CENTRE D'ETUDE ET DE RECHERCHE  
EN INFORMATIQUE  
(C.E.R.I.)

01 BP 1091 Bobo-Dioulasso 01

SYSTEME D'INFORMATION  
MULTIMEDIA POUR  
L'ENVIRONNEMENT SUBSAHARIEN  
(SIMES)

*Thème :*

*Acquisition de données multi-modales  
dans un environnement hétérogène*

**MEMOIRE DE FIN D'ETUDES**

Présenté et soutenu publiquement le 13 Février 2001  
pour l'obtention du

**Diplôme d'ingénieur de conception en informatique**

par

**Loé SANOU**

**Composition du jury :**

*Président :* Pr. M. Y. Théodore TAPSOBA

*Rapporteur :* M. Mesmin T. DANDJINO

*Examineur :* M. Ali B. KABA, *directeur du mémoire*  
*Maître de stage.*

---

**Année universitaire 1999/2000**

## SOMMAIRE

<b>REMERCIEMENTS</b> .....	<b>3</b>
<b>DEDICACE</b> .....	<b>4</b>
<b>INTRODUCTION</b> .....	<b>5</b>
<b>I – MOTIVATION DU SUJET</b> .....	<b>7</b>
I - 1 – LE DOMAINE D’ETUDE.....	7
I – 1 - 1 – <i>Les objectifs généraux du projet SIMES</i> .....	7
I – 1 - 2 - <i>L'approche d'intégration dans SIMES</i> .....	7
I - 2 – PROBLEMATIQUE .....	8
I – 2 - 1 - <i>Analyse spécifique du fonctionnement d'un site SIMES</i> .....	8
I – 2 - 2 – <i>Spécification du problème à résoudre</i> .....	9
<b>II – ETUDE METHODOLOGIQUE</b> .....	<b>12</b>
II – 1 – METHODOLOGIE ADOPTEE POUR LA CONCEPTION ET LA REALISATION DU PROJET .....	12
II – 2 – MODELISATION DU SYSTEME .....	15
II – 2 – 1 – <i>Description du système</i> .....	15
II – 2 – 2 - <i>UML : Unified Modeling Language</i> .....	15
II – 2 – 3 - <i>MODELISATIONS DU SYSTEME</i> .....	17
<b>III – IMPLEMENTATION DES OUTILS DU SYSTEME</b> .....	<b>59</b>
III - 1 - ARCHITECTURE FONCTIONNELLE DU SIMES .....	59
III - 2 - IMPLEMENTATION FONCTIONNELLE.....	60
III - 2 - 1 - <i>Architecture des couches : Vue fonctionnelle</i> .....	60
III - 2 - 2 - <i>ColdFusion : Présentation et usage dans le système</i> .....	61
III - 3 - APPLICATIONS WEB AVEC COLDFUSION.....	62
III - 3 - 1 - <i>Description d'application</i> .....	62
III - 3 - 2 - <i>Schémas fonctionnels</i> .....	63
III - 3 - 3 - <i>Informations partagées par les applications du système</i> .....	64
III - 3 - 4 - <i>Enchaînement des pages</i> .....	65
III - 3 - 5 - <i>Les pages entraînés par des bases de données</i> .....	65
III - 4 - REALISATION DE QUELQUES ELEMENTS DU SYSTEME.....	65
III - 4 - 1 - <i>La structuration des fonctionnalités par rubrique</i> .....	66
III - 4 - 2 - <i>Connexion d'un utilisateur et ouverture / fin d'une session</i> .....	66
III - 4 - 3 - <i>Les nouveautés et les événements du projet</i> .....	69
III - 4 - 4 - <i>Les textes et publications</i> .....	70
III - 4 - 5 - <i>Le gestionnaire de document</i> .....	71
III - 4 - 6 - <i>La mise en ligne d'un document</i> .....	73
<b>CONCLUSION</b> .....	<b>74</b>
<b>BIBLIOGRAPHIE</b> .....	<b>75</b>

## REMERCIEMENTS

La liste de ceux qui ont contribué à ma formation jusqu'à l'élaboration de ce document est très longue. Cependant rien n'aurait été possible sans les enseignements que j'ai reçus.

Il m'est particulièrement agréable de souligner le rôle précieux de mon directeur de mémoire à la fois efficace, avisé et motivé qu'a joué M. Ali B. KABA pendant plus de six (06) mois. Sa participation à la définition de mon sujet et à la validation objective des résultats méthodologiques et de conception, ainsi qu'à la mise en forme d'un texte plus précis et allégé, a été tout à fait significative. Qu'il trouve ici l'expression de ma très sincère reconnaissance !

Je tiens à exprimer mon amicale reconnaissance à monsieur Abdou - Karim DIALLO, Denis PLANCHAMP et à tous ceux qui m'ont aidé dans mes travaux. Que Mme KABA Elise, chef du centre informatique de l'UPB, et tout le personnel trouvent également ici l'expression de ma reconnaissance pour leur collaboration.

J'adresse aussi mes profonds remerciements à tous les enseignants de l'ESI pour les conseils prodigués.

Si par bonheur ce mémoire assez complexe a pu être réalisé, je le dois à l'ESI partenaire (membre) du projet SIMES qui a accepté m'accueillir dans le cadre de mes travaux. Ainsi, j'adresse mes sincères remerciements au Professeur Théodore TAPSOBA, Directeur de l'ESI ; Toundé Mesmin DANDJINOU directeur des études de l'ESI et l'ensemble du personnel de l'ESI.

J'en sais gré à Yacouba SANOU, Sié Chockail SANOU, Marie Cécile MEDAH et à toute ma famille pour le soutien qu'ils m'ont apporté.

Je ne saurais terminer sans remercier tous mes promotionnaires de l'ESI et tous les étudiants de l'ESI. Je tiens ainsi à leur exprimer mon amicale reconnaissance.

## **DEDICACE**

*A la famille SANOU à Orodara et en particulier à mon père  
DEKRIN qui n'est plus de ce monde, je dédie ce mémoire.*

## INTRODUCTION

Le mémoire de fin d'études consiste en la résolution d'un problème scientifique et technique nécessitant la mise en œuvre d'une démarche synthétique, faisant intervenir tout ou partie du savoir et du savoir-faire acquis par l'élève - ingénieur au cours de sa formation.

Notre étude s'est faite dans le cadre d'un projet de recherche SIMES (Système d'Information Multimédia pour l'Environnement Subsaharien) au Centre d'Etude et de Recherche en Informatique (CERI). Le CERI est un centre de l'Université Polytechnique de Bobo-Dioulasso (UPB). Le SIMES vise la maîtrise de l'environnement sub-saharien par l'utilisation des techniques informatiques.

Le SIMES a comme sources d'informations d'une part les données de ses partenaires et d'autre part les sources d'informations propres au projet lui-même. Celles propres au SIMES sont des bases de données où sont encapsulées des informations extraites des sources d'informations des partenaires. L'acquisition et le traitement de ces données constituent le problème majeur du projet. Notre travail devrait contribuer à la résolution de ce problème.

Dans ce document qui constitue le rapport de notre étude, on retrouvera trois (3) parties :

- une première partie où l'on présente le domaine d'étude, les raisons et le thème de l'étude ;
- une deuxième partie correspond à l'approche méthodologique utilisée et aux modèles conceptuels du système ;
- et enfin une dernière partie aborde l'implémentation de certains outils du système étudié.

***DOMAINE D'ETUDE  
ET  
PROBLEMATIQUE***

## I – MOTIVATION DU SUJET

### I - 1 – Le domaine d'étude

#### I – 1 - 1 – Les objectifs généraux du projet SIMES

SIMES signifie « Système d'Information Multimédia pour l'Environnement Subsaharien ». C'est un projet qui a pour objectifs scientifiques et technologiques la mise en place d'un système d'informations sur l'environnement subsaharien utilisant à fond les nouvelles technologies de l'informatique et de la communication. Au travers de ce projet, il est recherché le développement des compétences locales en informatique environnementale.

Les différentes tâches définies dans le cadre du projet sont :

- Acquisition de données, stockage et pré-traitement ;
- Traitement des données et mise en évidence des indicateurs ;
- Intégration des outils à l'intérieur d'un système générateur d'utilitaires conviviaux ;
- Implémentation des systèmes opérationnels pour les opérations pilotes ;
- Séries de conférences sur la société d'information.

Le projet SIMES regroupe onze (11) partenaires que sont : Institut National de Recherche en Informatique et en Automatique (INRIA - France), Institut de Recherche pour le Développement (IRD - France), Université d'Oxford (Angleterre), Université libre de Bruxelles (Belgique), Université de Berne (Allemagne), ERCIM (France), Ecole Supérieure Polytechnique de Dakar (Sénégal), Université de Yaoundé (Cameroun), Ecole Supérieure d'Informatique de Bobo-Dioulasso (Burkina Faso), Institut d'Economie Rurale (Mali) et CNTIG (Côte d'Ivoire).

#### I – 1 - 2 - L'approche d'intégration dans SIMES

Le système d'information SIMES peut être vu sous deux (2) angles : une vision analytique, centrée sur les composants ; et une vision fédérative faisant de ses éléments constitutifs un ensemble bien coordonné.

Analytiquement, le système est un ensemble de composants dont certains, notamment les outils, sont conçus de façon générique et de manière à être greffés ou intégrés facilement et rapidement dans un système ouvert et évolutif. Ces composants peuvent être grossièrement repartis en deux catégories :

- **les composants sources d'informations** : ils contiennent et fournissent de l'information au système SIMES. Ce sont d'une part les sources d'informations qui sont gérées par des organisations mettant à la disposition du projet SIMES les informations qu'elles ont collectées et qu'elles souhaiteraient continuer à gérer comme de par le passé, et d'autre part les sources d'informations propres au SIMES qui sont en fait des bases de données où sont encapsulées des informations extraites des sources d'informations de la catégorie précédente.

- **les composants outils** : ils servent à l'acquisition de l'information, la gestion de la connaissance, au traitement et à la diffusion de l'information, à l'administration et à la coordination du système.

Synthétiquement, on se focalise beaucoup plus sur l'aspect associatif des sources d'informations ainsi que celui de l'administration et de la coordination des différents composants. Le système SIMES qui ne se veut pas être un système figé, s'appuie sur des systèmes existants et capables d'évoluer par l'intégration ou le retrait de nouveaux composants. Par conséquent, une instanciation du système SIMES peut être vu comme une fédération de composants existants à un moment donné capable d'extraire en partie ou en totalité des informations contenues dans les systèmes locaux et de les intégrer dans un référentiel de connaissances tenant compte des spécificités des problématiques environnementales.

Ces deux visions "boîte à outils" et "fédération de composants" ne sont pas incompatibles mais se complètent et permettent d'apprécier toutes les particularités du système.

## **I - 2 – Problématique**

Le système SIMES puise ses données dans des sources diverses créées et maintenues par des structures autonomes. Il doit donc prendre en compte l'hétérogénéité des sources de données, non seulement par rapport aux types de données (fichiers textes, images cartographiques, ...), mais aussi par rapport aux types de système de gestion de bases de données (Access, Oracle, Fox Pro, ...) puisque toutes les données pertinentes ne sont pas centralisées, le faire aurait nécessité de grands moyens matériels et des procédures lourdes pour assurer les mises à jour des données produites par les sites distants.

En rejetant l'idée de la centralisation des données, il faut donc résoudre le problème de l'acquisition de l'information au moyen des composants outils.

### **I – 2 - 1 - Analyse spécifique du fonctionnement d'un site SIMES**

Pour le bon fonctionnement du système, il faut développer des outils (ou composants logiciels) capables d'exploiter les différentes bases de données des partenaires du projet SIMES et de faciliter la diffusion des informations entre les différents membres du projet.

Pour mieux percevoir la diversité des sources de données, penchons-nous sur l'exemple concret de la base de données d'un des partenaires du projet SIMES, celle ayant comme source d'information l'Observatoire de pêche dans le Delta Central du Niger (à la suite des recommandations de l'IRD, ex ORSTOM). Cette source d'information est basée à l'Institut d'Economie Rurale (IER) à Mopti au Mali. L'IER gère la base de données installée sur un PC et développée avec le logiciel Fox Pro.

Cette base de données relationnelle contient l'état d'occupation de 72 villages et campements (fréquence bimestrielle), l'activité et la mobilité des ménages (fréquence bimestrielle),



les débarquements des pirogues de pêche sur une quinzaine de sites et les mises en place de dispositifs fixes (environ 70 barrages).

Il s'agit d'un système de collecte et de restitution rapide d'informations pertinentes sur les ressources en poisson et leur exploitation, intégrant le comportement des acteurs et les modes de production.

Cette base de données comme bien d'autres, est une source d'informations et de données potentielles pour le SIMES. Son exploitation, de même que celui des autres sera d'un grand intérêt.

L'étude spécifique et la description des composants logiciels de ce système faites par le partenaire l'ont conduit à l'exploitation de la base par des programmes Java, vu les multiples avantages de ce langage orienté objet. Cette base de données est administrée à l'aide d'application Fox Pro.

Le système SIMES devra offrir une structure (catégorie) d'objets portables qui s'effectuent sur différente plate-formes et une grande flexibilité pour le développement des applications distribuées afin de pouvoir prendre en compte l'ensemble des applications déjà développées par les partenaires.

## **I - 2 - 2 – Spécification du problème à résoudre**

Le sujet faisant l'objet de notre mémoire de fin d'études au cycle des ingénieurs de conception en informatique est : « ACQUISITION DE DONNEES MULTI-MODALES DANS UN ENVIRONNEMENT HETEROGENE ».

Il s'agit pour nous de mettre en place un système global permettant d'acquérir des données ou informations de différents types dans un environnement composé de différents types de systèmes de gestion de bases de données contenant différents types de données (fichiers textes, images cartographiques, ...).

De son poste (client) distant, un utilisateur pourra mettre à jour une base de données se trouvant sur une autre machine (serveur). Les informations (à rechercher ou à stocker) seront saisies à partir du navigateur (dont dispose le client) qui auparavant aurait chargé l'interface du programme qu'il utilise.

Notre étude porte, d'une façon générale, sur l'exploitation des données des différents partenaires et l'échange et la communication entre les membres du projet. La résolution du problème passe par une modélisation d'un site de pilotage pour l'exploitation des ressources du SIMES. L'établissement de liens utiles entre les partenaires lors des communications et d'autres mécanismes accessoires compléteront les fonctionnalités du système. Le système sera un ensemble de composants logiciels. Leur mise en place devra permettre, d'une façon globale, l'exploitation des données à travers un environnement hétérogène.

Ces composants selon la terminologie objet seront sous une forme encapsulée, c'est-à-dire qu'on utilisera certains programmes écrits en des langages tels que Java, script CGI, script perl, etc interfaçants des applications Fox Pro ou autres.

Le principe d'encapsulation consiste à regrouper dans un même emballage sous forme de boîte noire les attributs et les méthodes d'un objet de sorte qu'un utilisateur (humain ou autre objet) ne puisse les atteindre directement. L'accès aux propriétés d'un objet se fait uniquement par envoi

de messages qui sont des noms de méthodes avec des arguments. L'utilisateur n'est concerné que par les services à effectuer par l'objet et non les détails et la façon dont ces services sont effectués.

L'encapsulation consiste donc à masquer les détails d'implémentation des objets en définissant une interface qui est une vue externe de l'objet et qui définit les services accessibles aux utilisateurs. C'est le principe de la boîte noire. Les programmes serviront d'interface par usage d'un navigateur (Nestcape, Internet explorer, ...).

Une description de la coopération de ces composants lors de l'acquisition des données facilitera la résolution du problème. Il faudra donc retrouver le principe de fonctionnement des utilitaires (les outils de travail) et les interactions possibles entre-eux afin de satisfaire les besoins. La façon d'implémenter les programmes sera un point critique de la réalisation pratique du système informatique car la réutilisation des outils est d'un grand intérêt. Cela nécessitera de réaliser les agents de pilotage et les outils d'intégration.

Le système à mettre en place devra être un facilitateur d'échanges de données, d'informations, de ressources, de documents entre les entités du système.

Le système SIMES est une application utilisée sur le réseau mondial à travers le Web. Le site SIMES est un site interactif et par conséquent les différents outils du système sont exécutés grâce aux navigateurs disponibles (Nestcape navigator, Internet explorer, Appletviews, Mosaïc, ...). Le système peut être vu comme un site de pilotage qui ne serait composé que d'outils intégrables et d'agents de pilotage. Ce serait donc un système d'intégration administré.

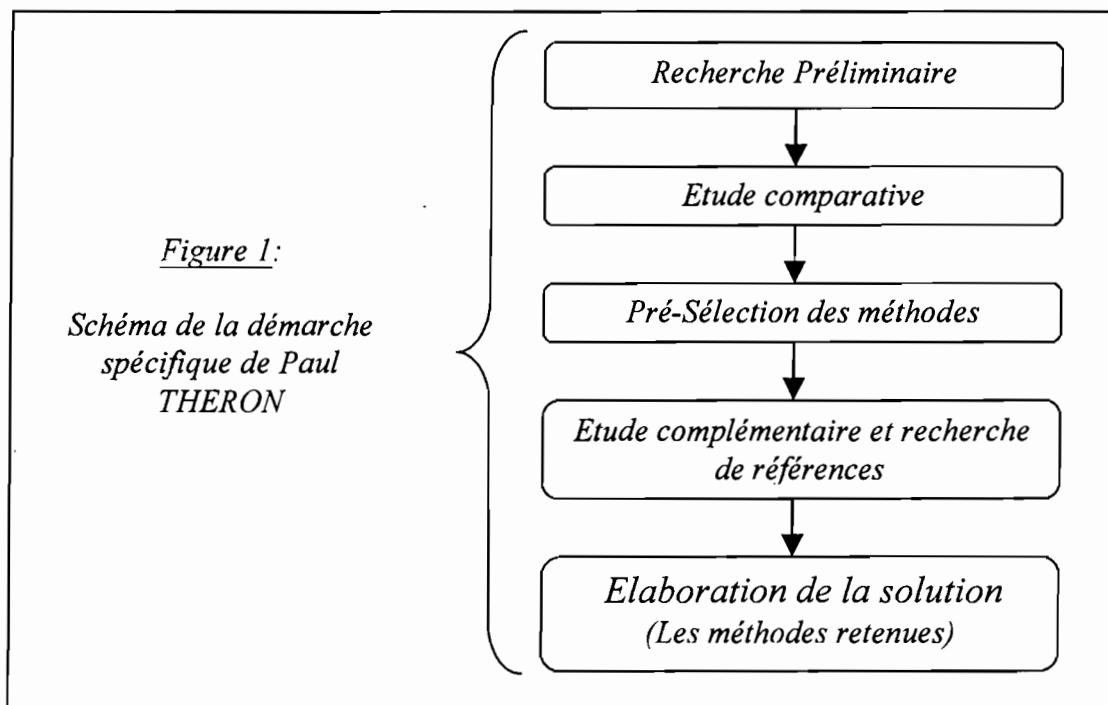
***METHODES  
ET  
MODELES***

## II – ETUDE METHODOLOGIQUE

### II – 1 – Méthodologie adoptée pour la conception et la réalisation du projet

Comment modéliser ? Quelle méthode choisir ? Pour répondre à ces questions et afin d'être objectif, nous avons utilisé, pour le choix des moyens méthodologiques, *une démarche spécifique développée dans [Theron 1988]* composée de cinq (5) étapes :

- *la recherche préliminaire* : il s'agit d'effectuer un tour d'horizon du marché de l'offre et d'éliminer d'emblée tous les outils et méthodes qui ne conviennent pas aux règles politiques de l'organisation et notamment au budget assigné. Il n'est donc retenu que ceux qui s'adaptent au travail demandé ;
- *l'étude comparative* : elle s'effectue au niveau de la vocation des moyens, de leurs objectifs, de leurs principes, de leurs origines et de leurs états de développement ;
- *la pré-sélection* : elle consiste à ne retenir que deux ou trois solutions parmi plusieurs. Chacune d'elles fera ensuite l'objet d'une étude complémentaire visant au choix final ;
- *l'étude complémentaire et recherche de références* : c'est l'étude des solutions sur le terrain afin de déterminer les plus adéquates au système à réaliser et aux tests effectués sur ces solutions ;
- *l'élaboration de la solution* : c'est la définition claire et précise de la solution retenue.



Le système SIMES sera modélisé à l'aide d'une combinaison de méthodes. Il s'agit d'une méthode hybride (Figure 2). C'est une méthode qui fait appel à plusieurs autres méthodes d'approche ou de conception. Les méthodes composant notre méthode hybride sont issues de l'application de la démarche spécifique.

Il sera utilisé une méthode systémique (MERISE) afin d'avoir une vision globale de notre système d'information (description globale du système, définition du périmètre d'étude).

Les méthodes systémiques reposent en premier lieu sur la compréhension du Système d'Information (SI) en tant que système parmi l'ensemble des systèmes de l'organisation. Elles mettent l'accent sur l'aspect global du système d'information qu'elles analysent comme un système entier (ou unique), sur l'importance de la décomposition du système en éléments et sur la mise en évidence des relations entre éléments.

Une méthode de conception par objet (UML) sera employée pour permettre de traiter aisément le problème de portabilité et d'évolution du système. Cette méthode permet la définition des outils et classes du système, la détermination des acteurs et favorise la définition de composants réutilisables.

Les méthodes de conception par objet permettent de bâtir un système d'application non pas à partir d'un modèle de données qu'on exploite et parcourt de façon navigationnelle pour construire les traitements (approches systémiques), mais comme un réseau de nœuds applicatifs, pouvant s'exécuter en parallèle, de façon autonome, et communiquer entre eux par échange de messages. Chaque nœud exploite un sous-ensemble du modèle de données et peut être porté sur n'importe quelle machine. Les méthodes de conception par objet sont basées sur le principe selon lequel tout système est composé d'un ensemble d'objets qui coopèrent par échange de message. Les méthodes de conception par objets (objets purs ou objets relationnels) utilisent le concept de type abstrait à l'aide de classes, les notions d'encapsulation, d'agrégation et d'héritage à l'aide des classes, des associations entre classes, des hiérarchies de classes et des messages entre objets. L'approche orientée objet permet l'évolution d'applications complexes et leur maintenance (modularité des applications, meilleur reflet du monde réel). Elle facilite l'intégration de systèmes d'informations hétérogènes (basées sur différentes technologies) par le biais de l'encapsulation des données et la coopération d'applications informatiques distribuées.

Une approche par les données et une approche par les traitements permettront de spécifier les applications à développer. L'approche par les composants [Ali B. KABA 2000] a permis de déterminer les constituants de chaque application et la façon dont l'application va s'exécuter.

L'approche par les données aboutie à un modèle initial du système qui est le modèle conceptuel de données sur lequel on va spécifier les applications à développer, les chemins d'accès que les fonctions ont besoin de suivre pour s'exécuter. Les premiers concepts utilisés sont ceux de rubrique, de relation et d'entité. La suite de la phase de conception est consacrée à étudier les opérations capables de modifier chacune des rubriques, relations et entités ou de les consulter.

L'approche par les traitements conduit à déterminer les constituants de chaque traitement et la façon dont il va s'exécuter (avec quelles données, selon quel algorithme, à la suite de quel événement, pour quel résultat) à partir d'un modèle initial de traitement établi sur la base d'une liste, en principe structurée, des fonctionnalités attendues. On commence par étudier les concepts d'activités et de règles d'exécution.

L'approche par les composants consiste à assembler des composants élémentaires pour construire un système complexe. Un composant logiciel fournit un ensemble de services sans être rattaché à une application particulière, et qui favorise la réutilisation des composants pour construire différents systèmes. Dans cette approche, il faut savoir identifier et organiser les composants en fonction d'une stratégie de réutilisation, des contraintes de déploiement et des composants disponibles.

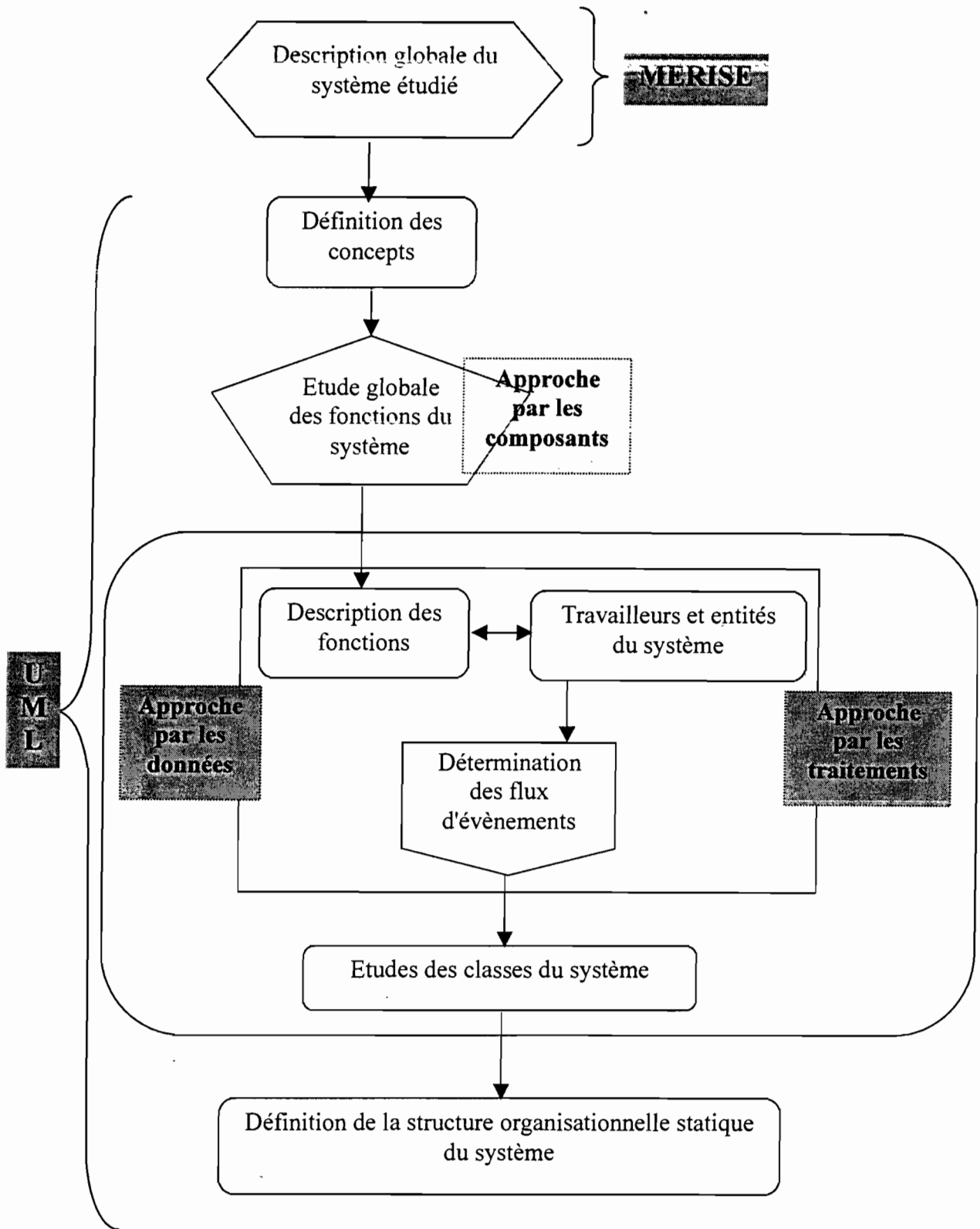


Figure 2: Schéma de méthode d'étude de modélisation du système

## II – 2 – Modélisation du système

### II – 2 – 1 – Description du système

Le système à mettre en place est un site de pilotage c'est-à-dire un support pour le travail en collaboration. Il devra faciliter les échanges d'informations (données à exploiter ou déjà exploitables) et la communication entre les entités du projet SIMES.

Le système d'une façon globale devrait offrir trois (03) fonctionnalités principales à savoir :

- *L'échange et le partage* : ils concernent soit les informations sur l'organisation des réunions, le planning d'évènements, les contacts, les groupes de travail, ..., soit les documents (système de navigation hiérarchisé et structuré en rubriques et sous rubriques, consultation et téléchargement de documents, mise en ligne de documents, archivage) ou encore les ressources bibliographiques.
- *La communication* : elle se résume en la messagerie électronique (listes de diffusion, client de messagerie), aux échanges asynchrones (forums de discussion par thèmes) et aux conférences électroniques (espaces de discussions en temps réel sur le site du projet).
- *L'administration du système* : elle consiste à ajouter de nouvelles rubriques et sous-rubriques à un document ou un entrepôt de données, à modifier l'organisation du site, et à maintenir des fichiers sur le serveur. Cette fonctionnalité permet au système d'évoluer.

### II – 2 – 2 - UML : Unified Modeling Language

UML est un langage de modélisation objet unifié. Il est issu de la fusion de trois (3) méthodes à savoir OMT (Object Modeling Technic), OOD (Object Oriented Design) et OOSE (Object Oriented SoftwareEngineering). C'est le résultat de l'évolution des méthodes d'analyse et de conception objet. Il contient des concepts, une syntaxe et une sémantique. De plus, UML possède une notation sous forme visuelle graphique fondée sur des diagrammes. Il n'impose pas de méthode de travail particulière. Cela permet d'intégrer UML dans tout processus de développement logiciel de manière transparente tout en véhiculant une démarche méthodique, ce qui facilite son intégration et limite les risques. Néanmoins UML préconise une démarche guidée par les besoins des utilisateurs du système, centrée sur l'architecture logicielle, itérative et incrémentale.

Une démarche guidée par les besoins des utilisateurs consiste à délimiter le périmètre du système. Les besoins des utilisateurs servent de fil conducteur tout au long du cycle de développement. A chaque phase (analyse, conception, test, ...) on vérifie l'effectivité de la prise en compte des besoins des utilisateurs.

Une démarche centrée sur l'architecture consiste à définir une architecture adaptée qui décrit les choix stratégiques déterminant la qualité du système (adaptabilité, performances, fiabilité, ...).

Une démarche itérative et incrémentale consiste à modéliser un système complexe en affinant son analyse par étapes. Cette démarche s'applique à tout le cycle de développement et favorise le prototypage.

Les constituants d'UML sont les éléments de modélisation et les diagrammes :

- ◆ Les éléments de modélisation représentent toutes les propriétés du langage. Ils ne sont pas limités par le symbolisme graphique utilisé. Cependant, ils doivent être clairs et précis. Pour ces constituants, UML définit des mécanismes d'extension qui permettront d'adapter le langage à des utilisations et des domaines particuliers voire de le faire évoluer.
- ◆ Les diagrammes constituent l'expression visuelle et graphique. Ils permettent de présenter une perspective de l'architecture du système à modéliser. Il n'y a pas suffisamment de symboles et de formes pour tout représenter. La simplicité et la cohérence doivent rester de mise. La correspondance entre les éléments de modélisation et les symboles graphiques n'est pas bijective. UML définit neuf (9) diagrammes : déploiement, composants, cas d'utilisation, classes, objet, collaboration, séquences, état - transition et activités. Parmi ces diagrammes, nous utilisons :
  - *Les diagrammes de cas d'utilisation* : ils permettent de présenter la description du comportement du système. Leur représentation utilise les formalismes de la figure 3.
  - *Les diagrammes d'état - transition* : ils modélisent une situation (un état) dynamique dans laquelle une ou plusieurs conditions peuvent se produire. Ils présentent les relations entre un état source et un état cible. La représentation est faite à l'aide des formalismes de la figure 3.
  - *Les diagrammes de collaboration* : ils montrent les interactions entre les objets et leurs liens.
  - *Les diagrammes de classes* : ils montrent les aspects statiques du modèle. Ils permettent de représenter des classes et leurs relations.

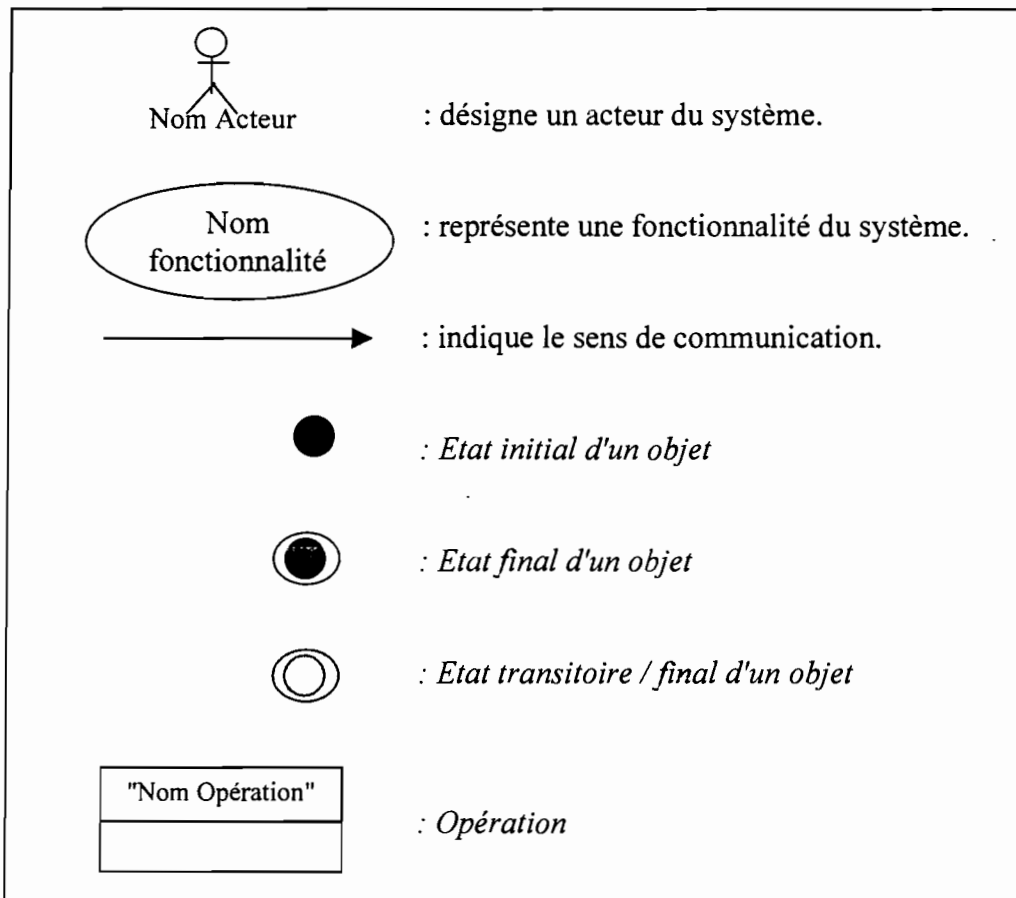


Figure 3 : Les différents formalismes de représentation des diagrammes en UML.



## II – 2 – 3 - MODELISATIONS DU SYSTEME

L'objectif de ces modélisations est la mise en place d'un modèle du système SIMES. Ces modélisations devront mettre en évidence les structures statiques organisationnelles du système et ses fonctions dynamiques. Cette modélisation passera par les cinq (5) étapes suivantes :

- L'étude du périmètre et des acteurs du système ;
- L'étude des fonctionnalités du système ;
- L'étude des travailleurs et des entités du système ;
- L'étude des workflows des fonctionnalités ;
- L'étude des structures organisationnelles.

### II – 2 – 3 – 1 – Etude du périmètre et des acteurs du système

Il s'agit de définir le contour du système à modéliser. Tout ce qui est en dehors du système et qui interagit avec lui est appelé «acteur».

Le système se limite aux mécanismes permettant le partage de logiciels et de données dans un environnement distribué. Ces objets sont en général de simples fichiers binaires. Leur descriptif permettra leur exploitation (exécution, exploitation, ...).

Comme acteurs du système, en partant des principales fonctionnalités, nous pouvons définir : Administrateur système, Coordinateur d'exécution, Accédeur aux données, Nommeur, Disposeur de données, Applicatifs, Invité, Membre SIMES, Administrateur SIMES.

Les trois (3) derniers font partie du système en tant que des travailleurs (critères de classification : Stéréotypes) mais extérieur au périmètre d'étude par la vue de l'approche par les données.

### II – 2 – 3 – 2 – Etudes des fonctionnalités du système

Dans le cadre de l'étude de cet aspect de notre système, nous utilisons les notions de « concept », de « fonctionnalité » et de « vue globale ».

Un concept peut être considéré comme une formalisation d'un objet ou d'un outil d'un système, autrement dit la manière de voir ou d'identifier une entité ou un objet du système.

Les concepts ci-dessous ont été retenus suite au "Rapport de réflexion introductive sur l'intégration dans le projet SIMES" [Ali B. KABA 2000] :

- **Activité** : C'est une séquence d'appels d'opérations (éventuellement limitée à un seul). Une activité peut être interactive (nécessitant la présence de l'utilisateur qui l'a lancée) ou non. Elle peut être immédiate (sa durée n'excède pas la session de l'utilisateur qui l'a lancée) ou non. Plusieurs activités peuvent être lancées en parallèle.
- **Artéfact** : c'est un objet du système qui permet d'atteindre des opérations mises à la disposition de l'utilisateur. Il peut être assimilé à un outil simple du système.
- **Bureau** : C'est un objet de l'espace de travail visible par l'utilisateur.
- **Entrepôt** : C'est un conteneur d'objets (qui peuvent être des fichiers, des bases de données, etc.) accessibles par les utilisateurs selon des droits associés à ces objets.
- **Espace de travail** : Il comporte selon les droits possédés par l'utilisateur la liste des services (ou fonctions) fournis à l'utilisateur et celle de ses activités en cours d'exécution.

- **Opération** : C'est une procédure élémentaire consistant au lancement d'une commande mise à disposition de l'utilisateur et à l'invocation d'un outil en local ou à distance.
- **Profil utilisateur** : Ce sont les informations de configuration qui sont conservées pour chaque utilisateur. Ces informations comprennent notamment les objets accessibles par l'utilisateur, l'espace de travail de l'utilisateur, les paramètres de bureau de l'utilisateur (tels que les fenêtres disponibles, les contenus des fenêtres, les fenêtres actives) et les paramètres de réactivation des connexions interrompues.
- **Rôle** : C'est l'ensemble des actions autorisées. Quand un compte utilisateur est autorisé à jouer un rôle, cet utilisateur bénéficie de toutes les permissions et de tous les droits de ce rôle. Les droits d'un rôle définissent l'accès d'un utilisateur (ou groupe d'utilisateurs) aux ressources du site SIMES et les actions qu'il peut y exécuter. On pourrait distinguer deux (2) rôles sur le site SIMES à savoir : Administrateur et Invité.
- **Session** : C'est la durée qui s'écoule entre une connexion et une déconnexion (normale ou anormale) d'un utilisateur au système. A l'ouverture d'une session le profil utilisateur est chargé et l'environnement de travail est configuré en fonction de ce profil.
- **Utilisateur** : C'est une personne humaine qui accède au système SIMES selon des droits qui lui ont été attribués.

Une fonctionnalité au niveau d'un système peut être vue comme un ensemble d'activités internes du système dont l'objectif est de fournir un résultat observable et mesurable pour un utilisateur du système.

Les fonctionnalités du système ont été déterminées par l'approche dite "par les composants". Un composant est un objet ou un ensemble d'objet appartenant à un même sous-système et ayant des relations sémantiques entre-eux. Les composants ont comme support de description le modèle conceptuel de base définis à partir des concepts de l'approche orientée objet.

La modélisation des fonctionnalités du système se fera par l'utilisation des cas d'utilisation (et vice versa). En considérant les concepts et les acteurs précédents, les principales fonctionnalités identifiées pour notre système sont les suivantes [Ali B. KABA 2000] :

- Accès utilisateur
- Administration des droits des utilisateurs et des groupes
- Administration de l'entrepôt de données
- Administration du système d'information
- Administration des paramètres système
- Administration d'une application
- Alimentation de l'entrepôt de données
- Consultation d'un objet
- Exécution d'un outil
- Exécution d'une activité
- Gestion d'une session
- Information d'un utilisateur
- Nommage d'un artéfact
- Nommage d'un ordinateur
- Nommage d'un utilisateur
- Stockage d'un objet
- Traitement d'une commande
- Traitement d'une donnée
- Visualisation de l'entrepôt de données

La vue globale des fonctionnalités quant à elle consiste à étudier les différents acteurs intervenant dans des fonctionnalités ayant des interactions entre elles. Elle ne permet aucunement de mettre en évidence les réactions entre les acteurs du groupe, ni les collaborations entre les fonctionnalités. La vue globale des fonctionnalités présente tout simplement les communications possibles entre des acteurs à l'aide de fonctionnalités.

Une description du comportement du système est présentée à l'aide des diagrammes de cas d'utilisation. Les figures 4 à 9 représentent les principaux diagrammes de cas d'utilisation modélisant notre système SIMES.

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobé par la limite du système, des associations de communication entre les acteurs et les situations d'utilisation des communications.

Un diagramme de cas d'utilisation indique le sens de communication entre acteurs et processus et souvent le nom de la relation. Les formalismes utilisés pour leur représentation ont été décrits au niveau de la figure 3.

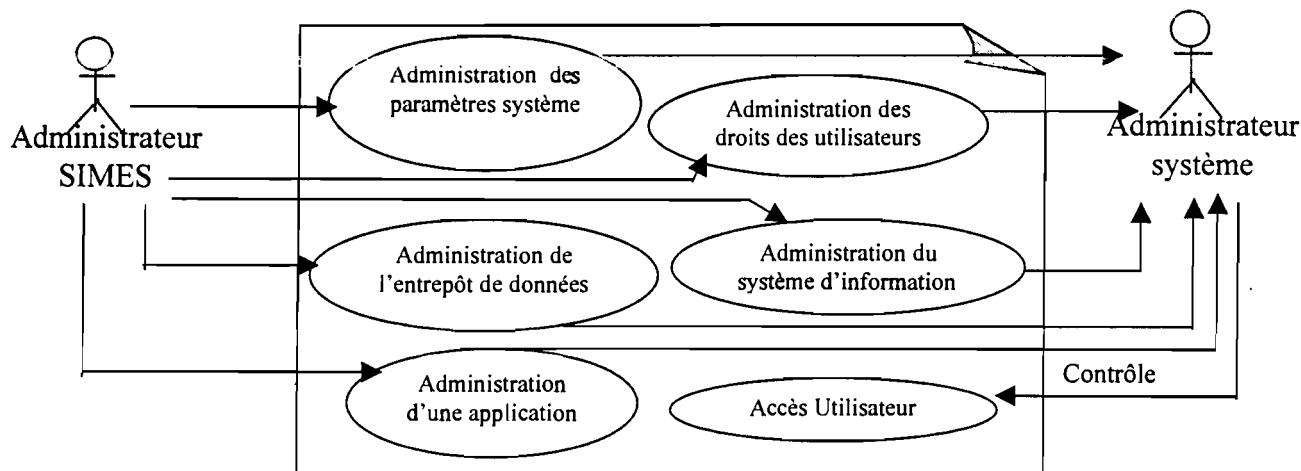


Figure 4 : Communications possibles entre Administrateur SIMES et Administrateur système

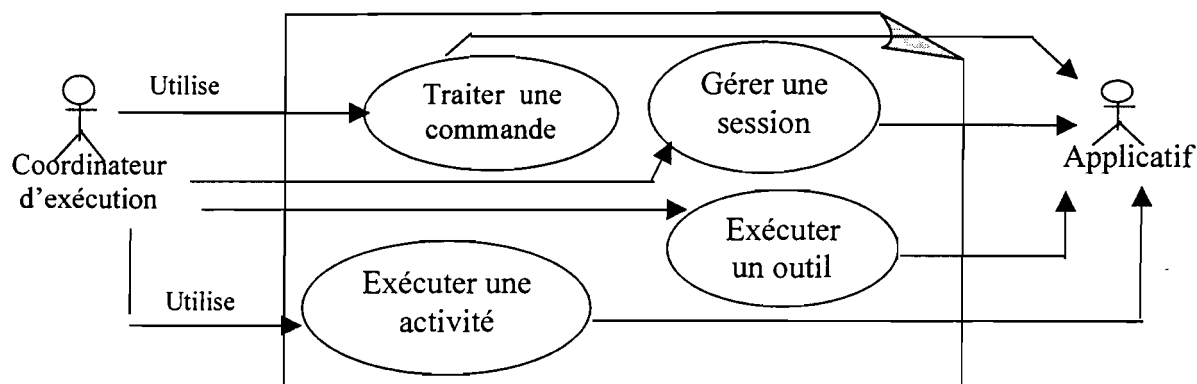


Figure 5 : Communications possibles entre Coordinateur d'exécution et Applicatif

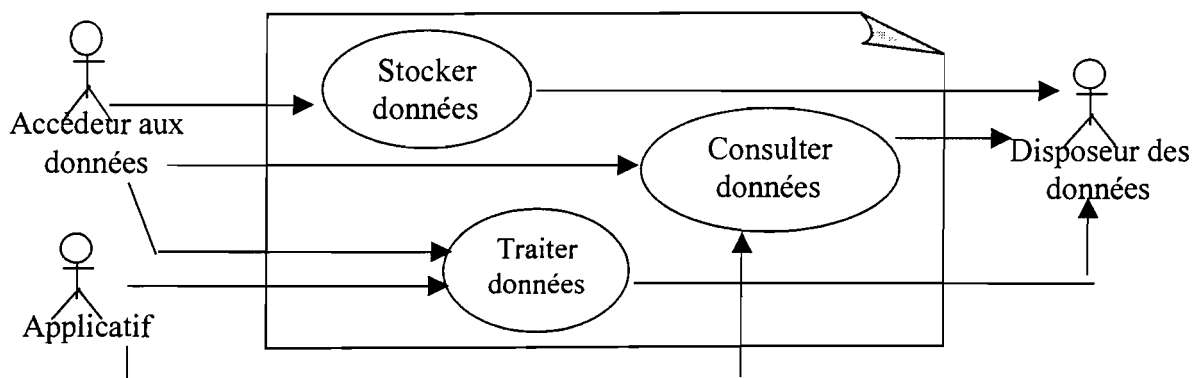


Figure 6 : Communications possibles entre Accédeur aux données et Disposeur des données

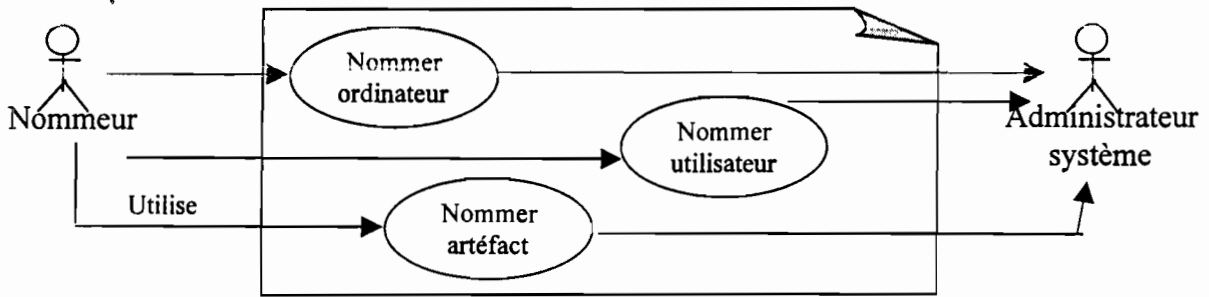


Figure 7 : Communications possibles entre Nommeur et Administrateur système

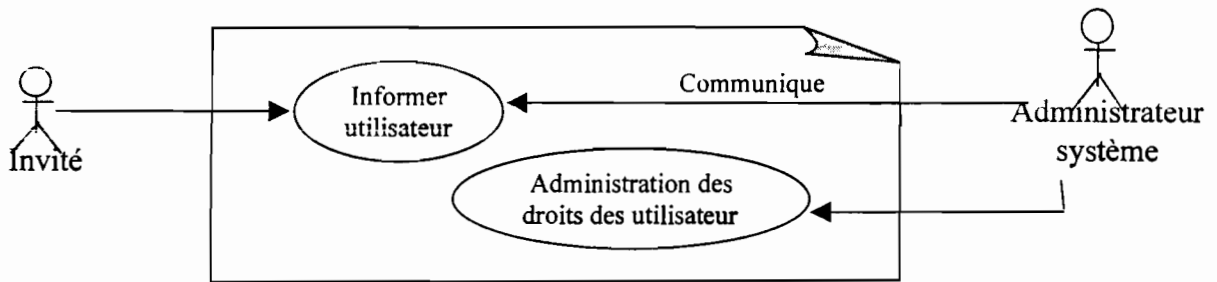


Figure 8 : Communications possibles entre Invité et Administrateur système

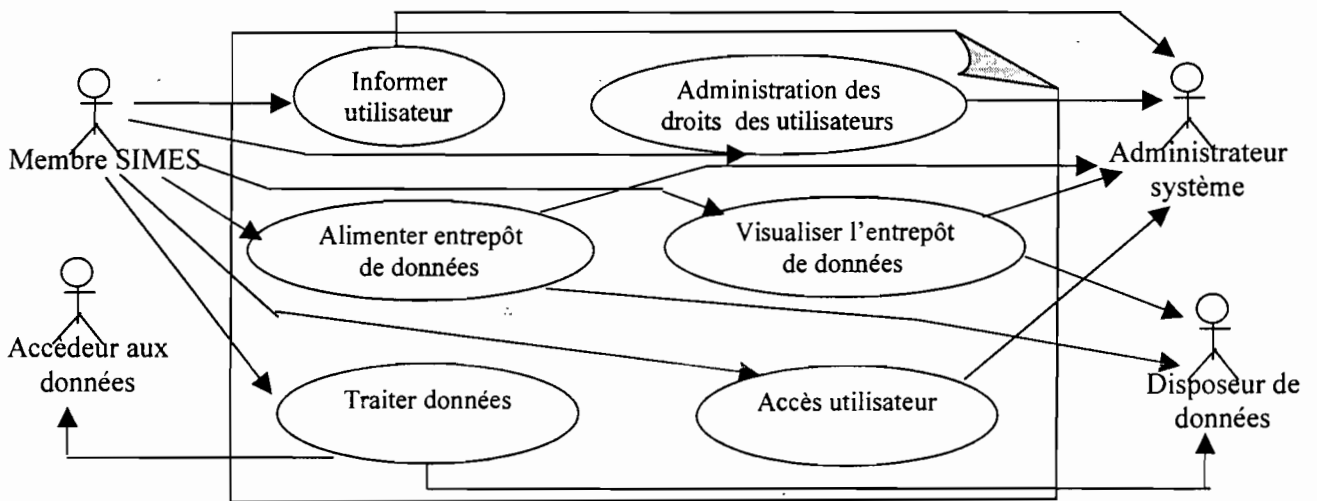


Figure 9 : Communications possibles entre Membre SIMES, Disposeur de données et Administrateur système.

Le cœur de notre travail est constitué de trois (3) parties fondamentales à savoir la gestion de l'entrepôt de données, la gestion des outils du système et la gestion des facilitateurs de services aux utilisateurs.

La gestion de l'entrepôt de données se compose des fonctionnalités suivantes :

- Administration de l'entrepôt de données,
- Administration d'une application,
- Alimentation de l'entrepôt de données,
- Visualisation de l'entrepôt de données.

La gestion des outils du système se résume aux fonctionnalités ci-après :

- Consultation d'un objet,
- Exécution d'un outil,
- Exécution d'une activité,
- Nommage d'un utilisateur,
- Nommage d'un ordinateur,
- Nommage d'un artefact,
- Stockage d'un objet,
- Traitement d'une commande,
- Traitement d'une donnée.

Le reste des fonctionnalités telles que Administration du système d'information, Gestion d'une session, Information d'un utilisateur, Accès utilisateur, Administration des paramètres système et Administration des droits des utilisateurs et des groupes, constitue des facilitateurs de services aux utilisateurs. Il favorise et facilite les activités des fonctionnalités de base. Il rend beaucoup plus agréable le pilotage du site.

Pour chacune des fonctionnalités, nous décrivons à l'aide de diagrammes d'états – transition comment évoluent les objets utilisés en réaction aux événements au cours des différentes activités.

Ces diagrammes d'états - transition utilisent les formalismes de la figure 3.

### ✓ Administration de l'entrepôt de données

Cette fonctionnalité permet de gérer les répertoires du système. Elle possède les activités suivantes :

- *Gestion des sauvegardes* : archiver l'entrepôt de données ou un répertoire de données au moyen d'outils de compression classiques aux fins de transfert ou de conservation.
- *Organiser l'entrepôt* : organiser l'arborescence de l'espace partagé en créant de nouvelles rubriques (dossiers) et sous-rubriques (sous-dossier) dans cet espace (pour les objets du site), modifier, renommer, supprimer ou déplacer des rubriques.

- *Gestion des artefacts* : modifier les informations attachées à un objet partagé du système SIMES, à un document ; supprimer ou déplacer un document vers une autre rubrique de destination. Ces traitements ont pour conséquence la mise à jour de l'espace de noms des objets.

Une représentation de ses activités est illustrée à l'aide de la figure 10.

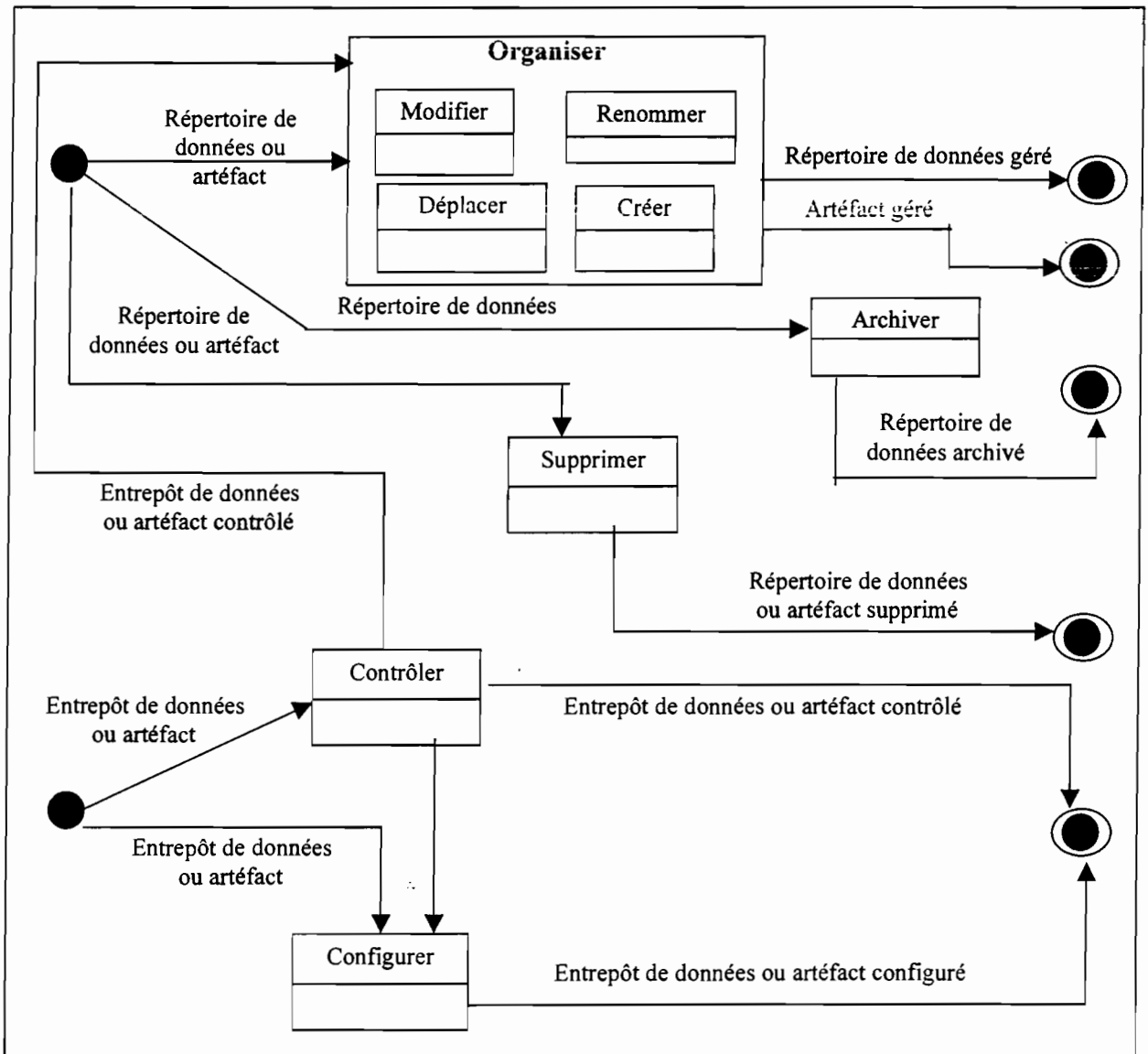


Figure 10 : Diagramme d'état – transition pour l'Administration de l'entrepôt de données

✓ **Administration d'une application**

Cette fonctionnalité consiste à définir de nouvelles applications pour le système et à décrire leurs sources de données et mode d'emploi. Elle se compose de :

- la définition de tables,
- la définition d'applications,
- la définition de sources de données et
- la gestion de tables, d'applications et de données.

Ces activités sont illustrées par la figure 11.

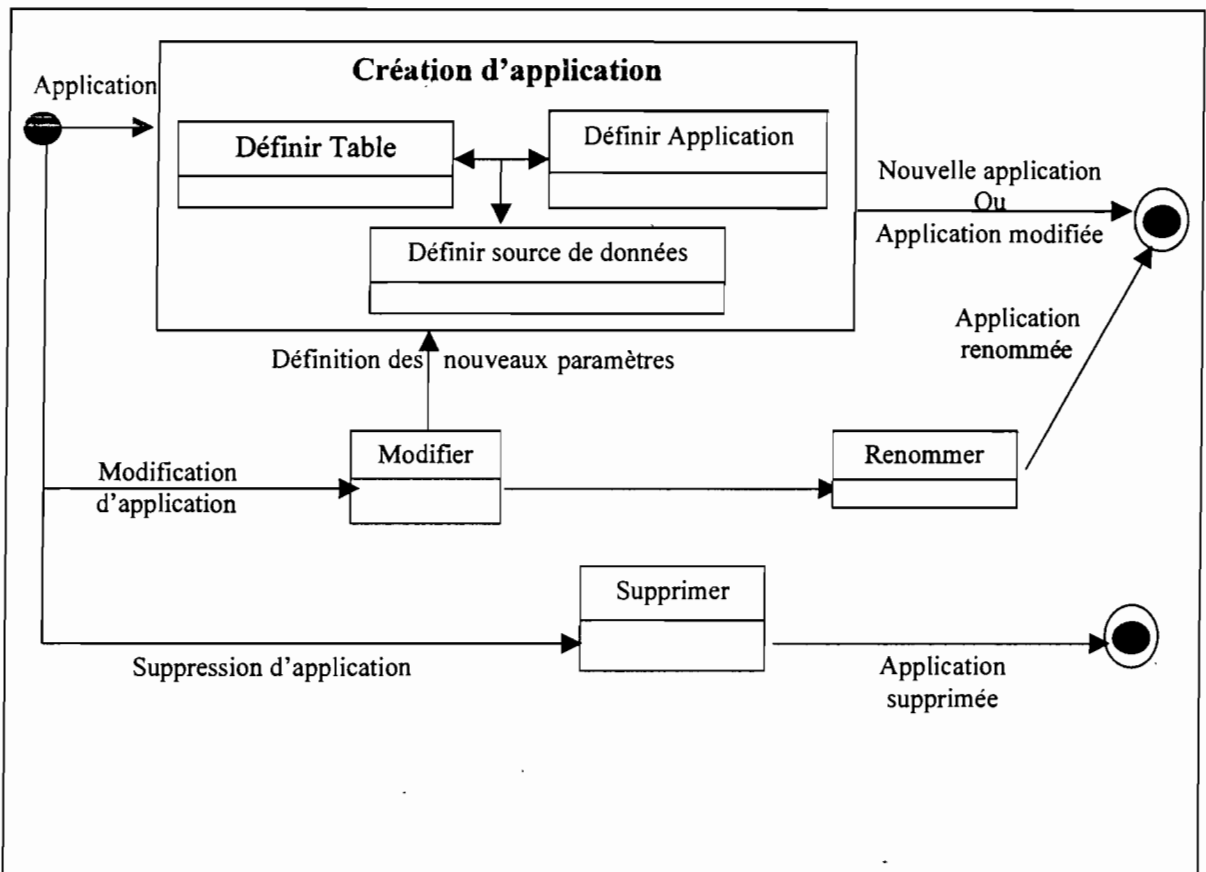


Figure 11 : Diagramme d'état – transition pour l'Administration d'une application.



✓ **Alimentation de l'entrepôt de données**

Elle a pour objet la mise en ligne des documents, outils et activités dans un espace de stockage partagé destiné aux utilisateurs du système SIMES. Les principales activités possibles sont :

- *Déposer un document* : fournir des renseignements sur le document, envoyer le document et les informations qui lui sont associées sous une rubrique préalablement sélectionnée d'un serveur SIMES et prévenir par courrier électronique les listes de diffusion de la mise en ligne du nouveau document.
- *Décrire un outil* : les propriétés, les fonctions de l'outil, les listes et les types de paramètres de ces fonctions et les rôles habilités à exécuter l'outil.
- *Décrire une activité* : les propriétés, les séquences d'outils qui entrent dans sa composition et les flots d'entrées et de sorties des outils ainsi que la destination des flots de stockage obligatoirement nommés sont données.

La description du comportement des objets de cette fonctionnalité est schématisée par la figure 12.

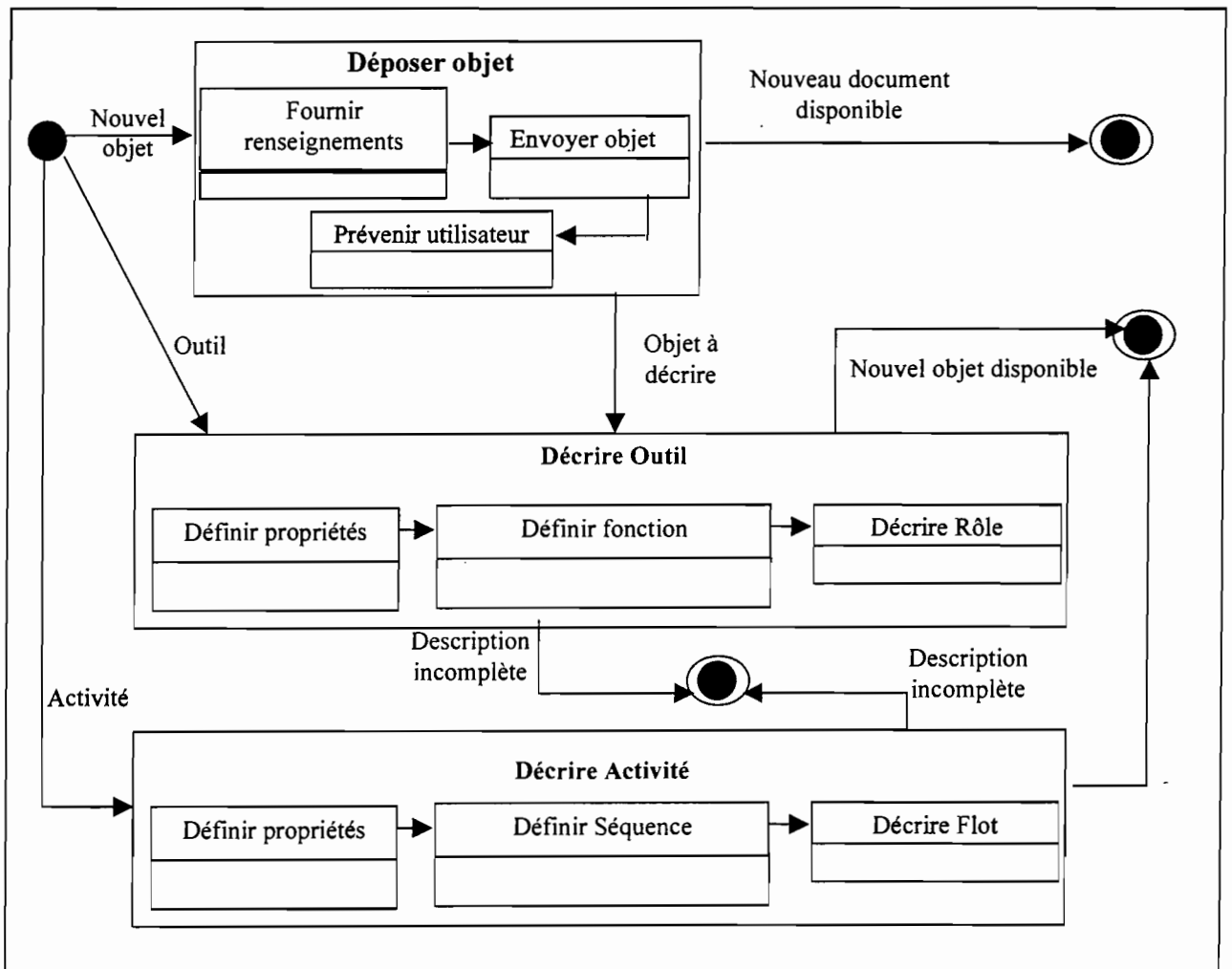


Figure 12 : Diagramme d'état – transition pour l'Alimentation de l'entrepôt de données.

✓ **Visualisation de l'entrepôt de données**

Cette fonctionnalité permet de voir l'espace de stockage des objets (documents, outils, etc.) partagés et stockés sur un serveur. Il comprend deux activités décrites à l'aide de la figure 13 :

- *L'accès au plan du site* : accéder grâce à un arbre à l'arborescence générale des rubriques, sous-rubriques et objets partagés du site.
- *Ouvrir un document* : consulter les informations des objets partagés du site, télécharger ces objets sur le disque local de la machine de l'utilisateur.

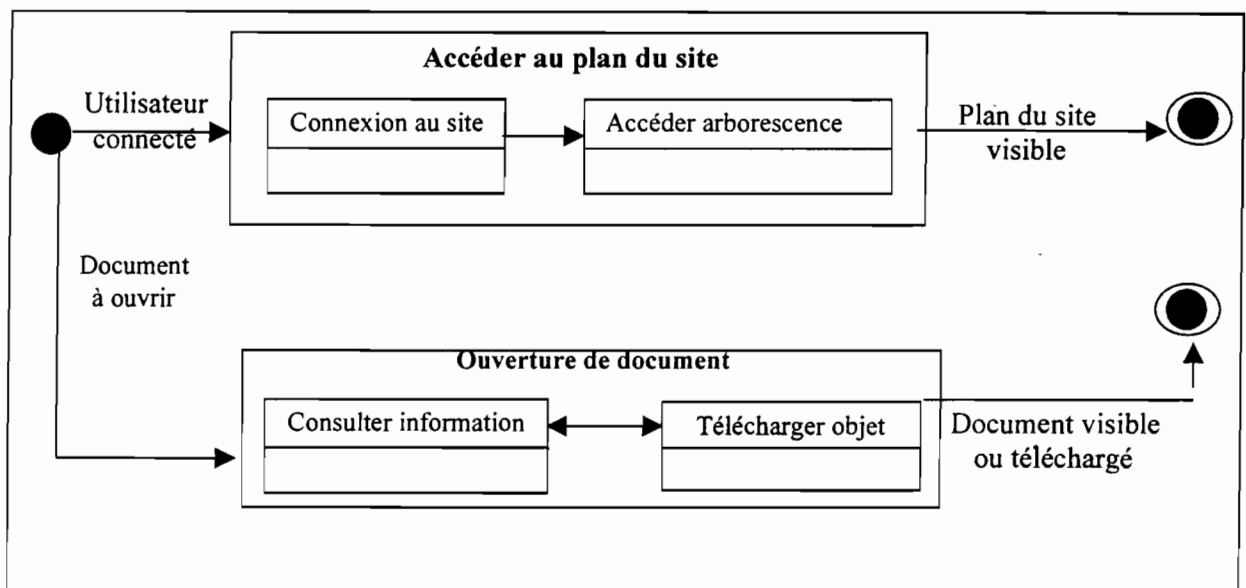


Figure 13 : Diagramme d'état-transition pour la Visualisation de l'entrepôt de données.

✓ **Traitement d'une donnée**

Cette fonctionnalité consiste à lancer et à exécuter des objets du système. Elle comprend :

- *Lancer une commande* : utiliser une invite de commandes pour effectuer des actions élémentaires.
- *Exécuter un outil* : choisir un outil parmi la liste d'outils fournie sur le site SIMES, fournir les paramètres effectifs.
- *Exécuter une activité* : changer les états successifs d'une activité accessible par un utilisateur.

La description de cette fonctionnalité est représentée par la figure 14.

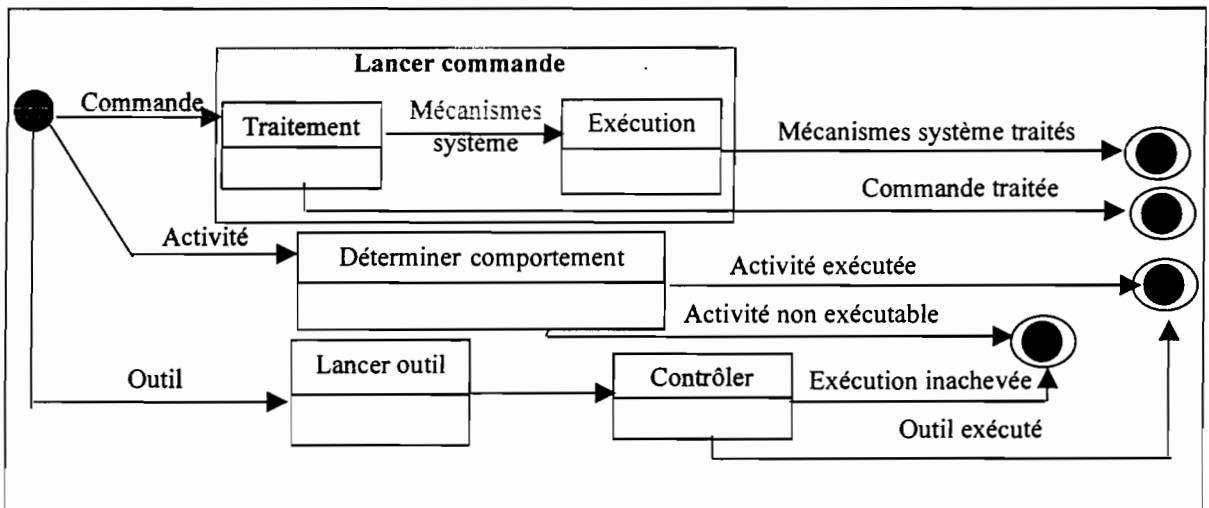


Figure 14 : Diagramme d'état – transition pour le Traitement d'une donnée.

✓ **Traitement d'une commande**

Cette fonctionnalité fournit des mécanismes de commandes permettant d'exécuter les mécanismes du système. Elle est illustrée sur la figure 15.

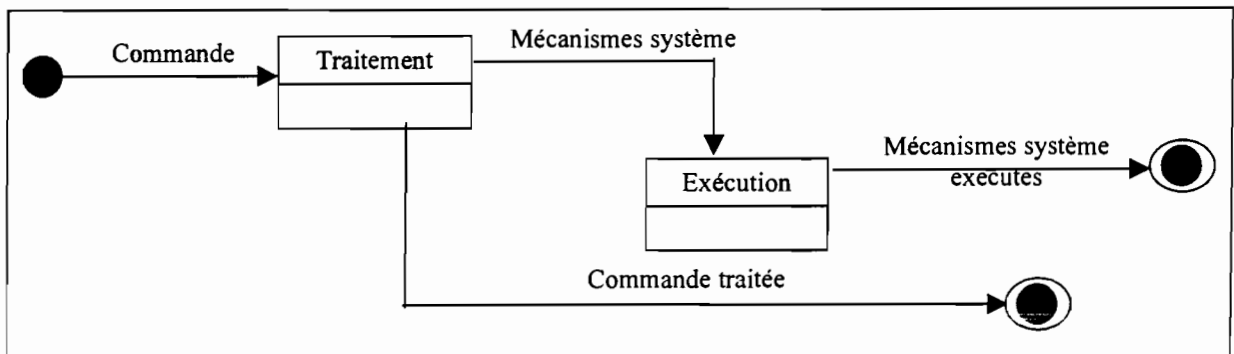


Figure 15 : Diagramme d'état – transition pour le Traitement d'une commande.

✓ **Stockage d'un objet**

Cette fonctionnalité consiste à transférer des objets décrits par l'utilisateur sur un site SIMES, et à les rendre visibles aux autres utilisateurs lorsque ces objets sont publics. Elle traite du stockage du contenu d'un objet, de sa description et de l'indexation du répertoire de données (figure 16).

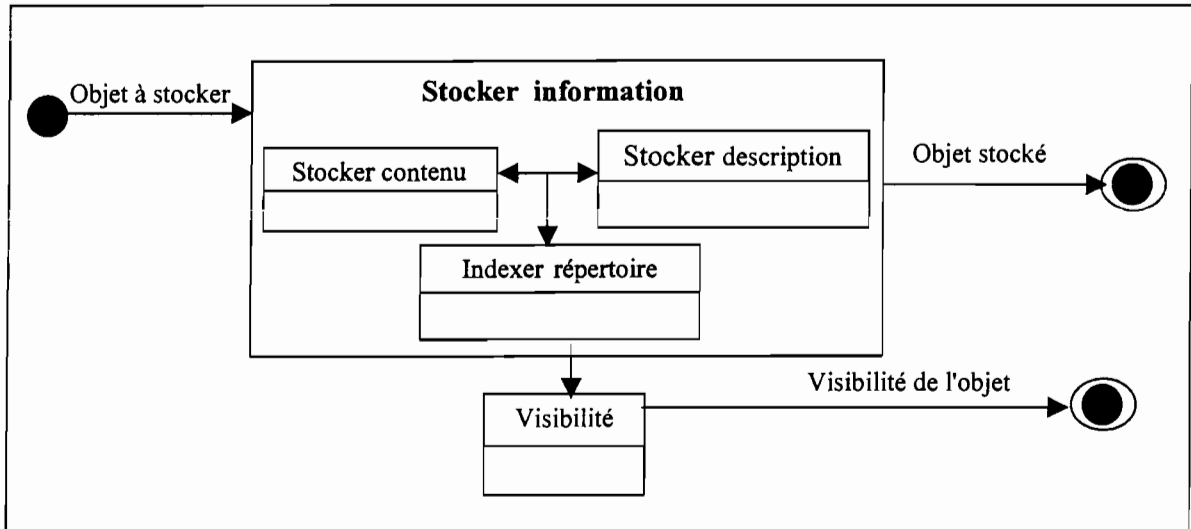


Figure 16 : Diagramme d'état – transition pour le Stockage d'un objet.

✓ **Consultation d'un objet**

Cette fonctionnalité permet l'accès aux objets du système SIMES. Il s'agit d'obtenir le contenu et la description de l'objet. Elle est décrite à l'aide de la figure 17.

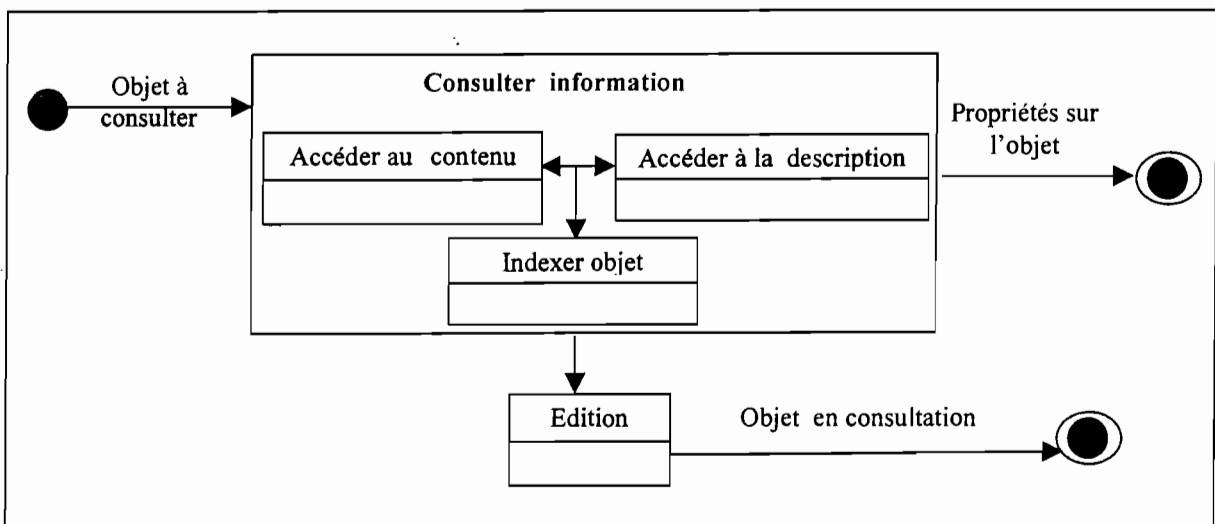


Figure 17 : Diagramme d'état – transition pour la Consultation d'un objet.

✓ **Exécution d'une activité**

Cette fonctionnalité permet de déterminer le comportement d'une activité et à changer ses états successifs (figure 18). Une activité au début est décrite. Elle est activée lorsqu'on la démarre. A ce stade, l'activité peut être suspendue ou peut terminer son exécution. Après sa suspension il est possible de reprendre son exécution.

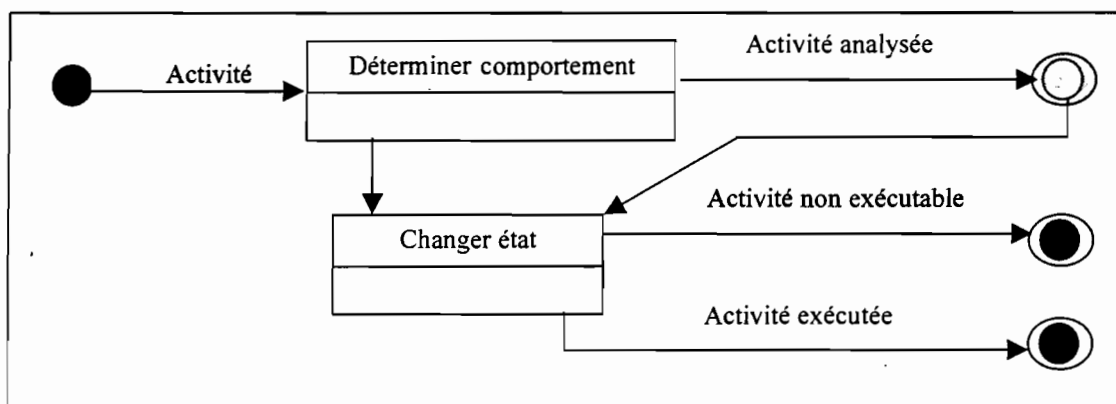
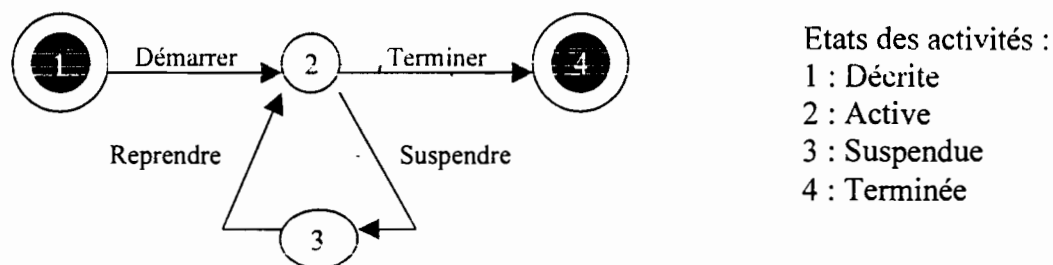


Figure 18 : Diagramme d'état – transition pour l'Exécution d'une activité.

✓ **Exécution d'un outil**

Cette fonctionnalité permet d'exécuter un outil et d'assurer aussi sa terminaison. Elle fournit les paramètres effectifs aux outils mis en œuvre (figure 19).

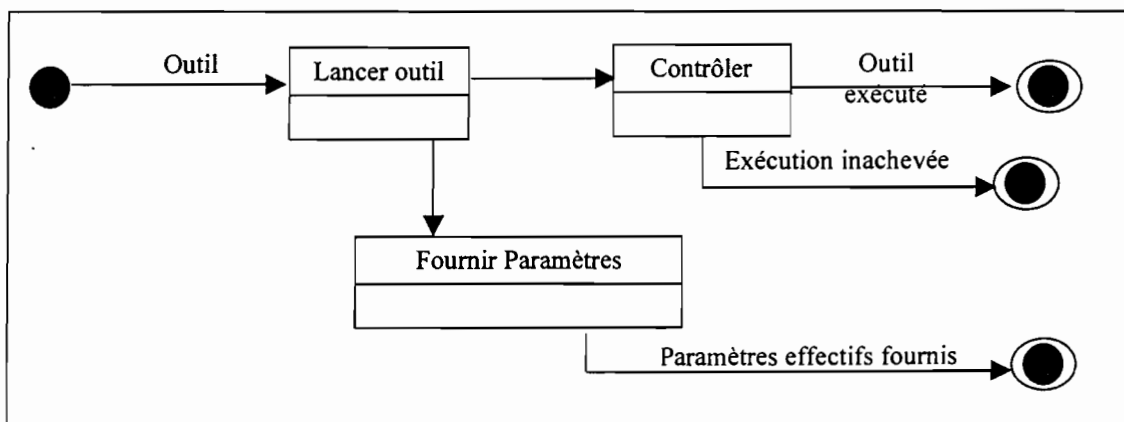


Figure 19 : Diagramme d'état – transition pour l'Exécution d'un outil.

✓ **Nommage d'un objet**

Les services de nommage constituent un service de nom statique qui permet de décrire, d'identifier et d'accéder via un nom logique à des objets du système. Ces objets peuvent être des ordinateurs, des utilisateurs ou des artefacts. La fonctionnalité du service de nommage est décrite à l'aide de la figure 20.

Les noms définis lors des configurations (pendant le nommage) sont mis à la disposition de solveurs effectuant des requêtes de résolution de nom sur le site.

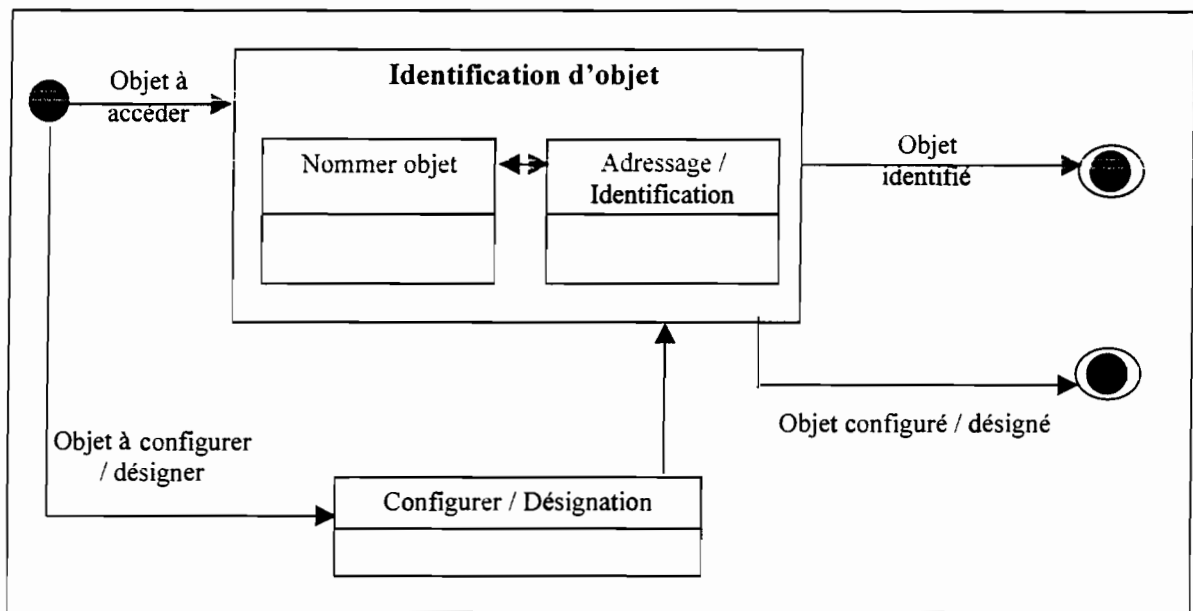


Figure 20 : Diagramme d'état – transition pour le Nommage d'un objet.

## ✓ Administration du système d'information

Cette fonctionnalité consiste à gérer l'ensemble des informations à mettre à la disposition d'un utilisateur. Elle est illustrée à l'aide de la figure 21. Elle se compose des activités suivantes :

- *Gestion du calendrier des événements* : éditer un calendrier pour ajouter, modifier ou supprimer des événements relatifs au système SIMES.
- *Gestion des thèmes et messages du forum* : ajouter, modifier, supprimer ou indexer des thèmes et des messages du forum du système SIMES.
- *Gestion des listes de diffusion et des messages archivés* : rendre publiques les listes de diffusion du système SIMES, gérer et indexer les messages de ces listes de diffusion afin d'en optimiser l'accès.
- *Gestion des thèmes de la rubrique* : ajouter, modifier ou supprimer des thèmes et des liens utiles au système.
- *Gestion des thèmes de la bibliothèque* : créer, modifier ou supprimer des références bibliographiques utiles pour le système.

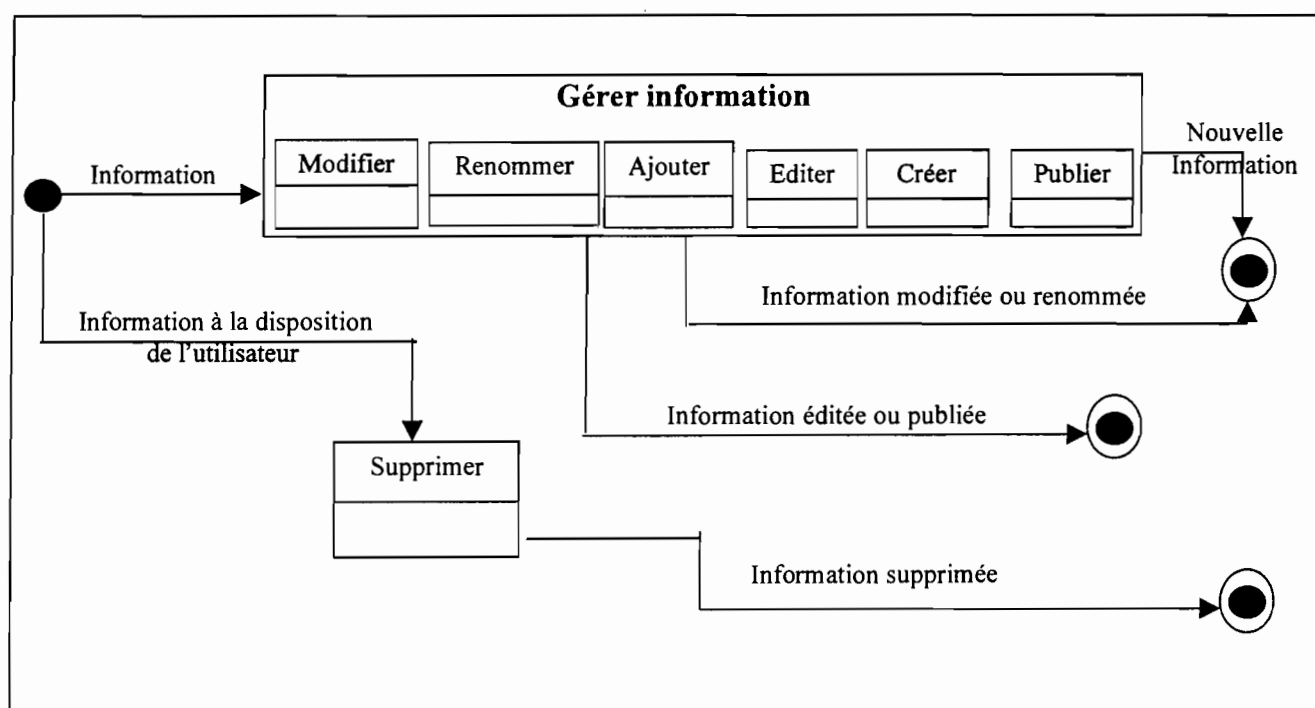


Figure 21 : Diagramme d'état – transition pour l'Administration du système d'information.

✓ **Gestion d'une session**

Cette fonctionnalité permet de coordonner l'exécution des activités et des outils par la mise en place d'une table contenant les informations sur les activités immédiates. La fin d'une session se traduit par la destruction des descriptifs correspondants. L'illustration de la description est présentée par la figure 22.

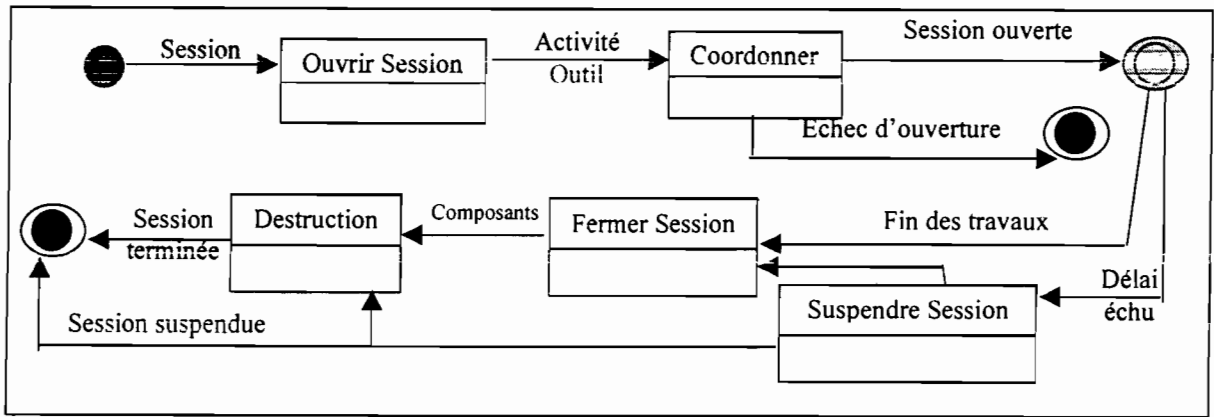


Figure 22 : Diagramme d'état – transition pour la Gestion d'une session.

✓ **Accès utilisateur**

Cette fonctionnalité permet à un utilisateur d'accéder à son compte et de changer ses paramètres de connexion. Il s'agit des activités suivantes illustrées par la figure 23 :

- *Connexion / Déconnexion* : l'accès au système SIMES est contrôlé par un mot de passe. La déconnexion est automatique au bout d'un certain temps d'inaction. Cependant, l'utilisateur a la possibilité de se déconnecter à tout moment.
- *Changement de mot de passe* : la possibilité est offerte à l'utilisateur de changer son mot de passe.

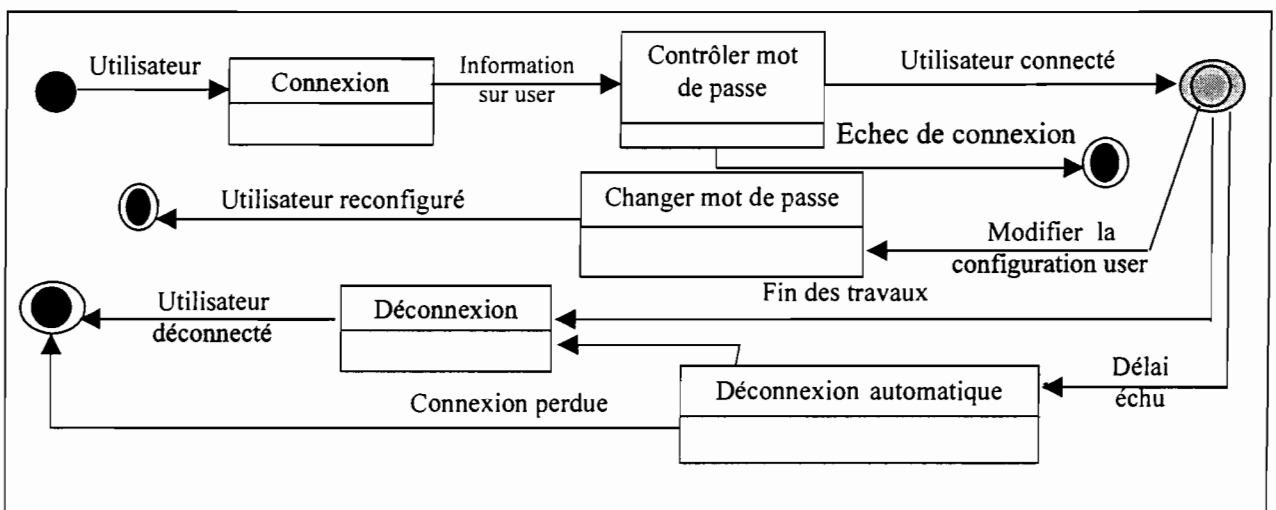


Figure 23 : Diagramme d'état – transition pour l'Accès Utilisateur.



✓ **Information d'un utilisateur**

Cette fonctionnalité consiste à mettre à la disposition d'un utilisateur l'ensemble des données (informations) du système SIMES auquel il a droit (figure 24). Ces renseignements portent sur les nouveautés, l'échéancier, les évènements privés, les adresses, la liste de diffusion et les forums.

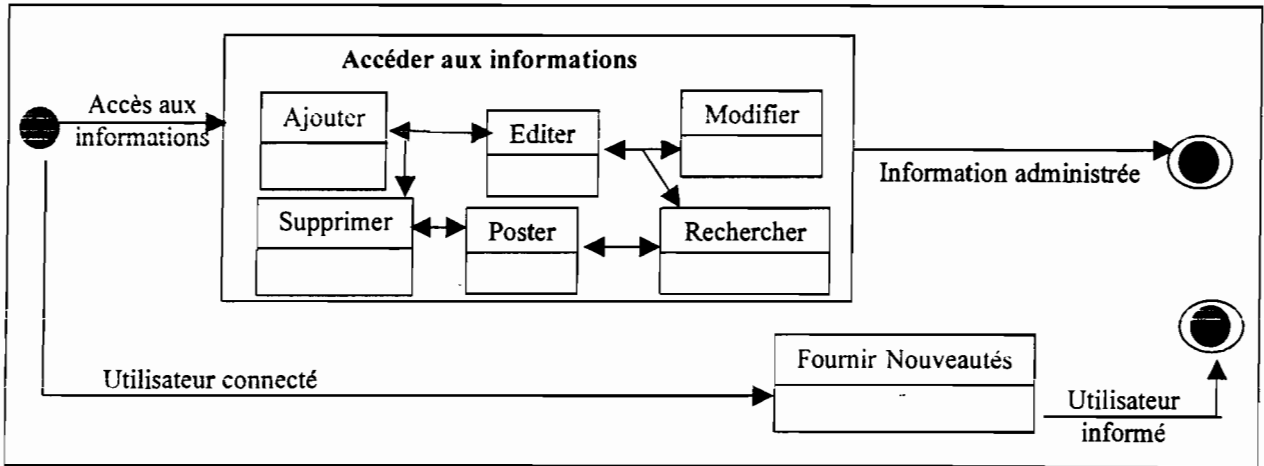


Figure 24 : Diagramme d'état – transition pour Information d'un utilisateur.

✓ **Administration des paramètres système**

Cette fonctionnalité permet la configuration des paramètres d'un serveur SIMES. Il s'agit de :

- Configurer les paramètres d'identification du serveur,
- Configuration des services accessibles,
- Configuration de la durée d'inaction d'un utilisateur,
- Configuration de la racine de l'entrepôt de données,
- Configuration des noms des autres serveurs SIMES,
- Configuration de la persistance des nouveautés,
- Gestion d'informations générales,
- Intégration d'une application.

Sa description graphique est donnée par son diagramme d'état – transition à la figure 25.

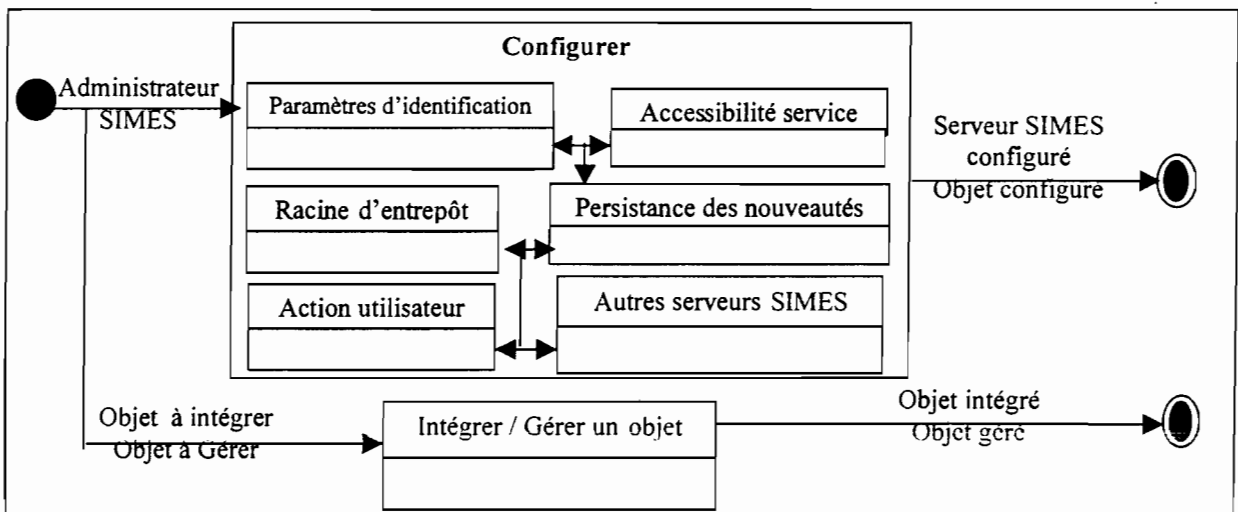


Figure 25 : Diagramme d'état – transition pour l'Administration des paramètres système

✓ Administration des droits des utilisateurs et groupes

Cette fonctionnalité consiste à gérer les droits d'accès au système, les membres du système ainsi que les actions possibles pour chaque utilisateur (voir diagramme d'état – transition de la figure 26). Elle comporte les activités suivantes :

- *Gestion des utilisateurs* : authentification possible des membres par tous les serveurs reconnus par le système SIMES ; ajout, modification, suppression ou configuration de l'environnement d'un utilisateur.
- *Gestion des rôles* : création, modification, suppression ou renommation des rôles assignés à un utilisateur.

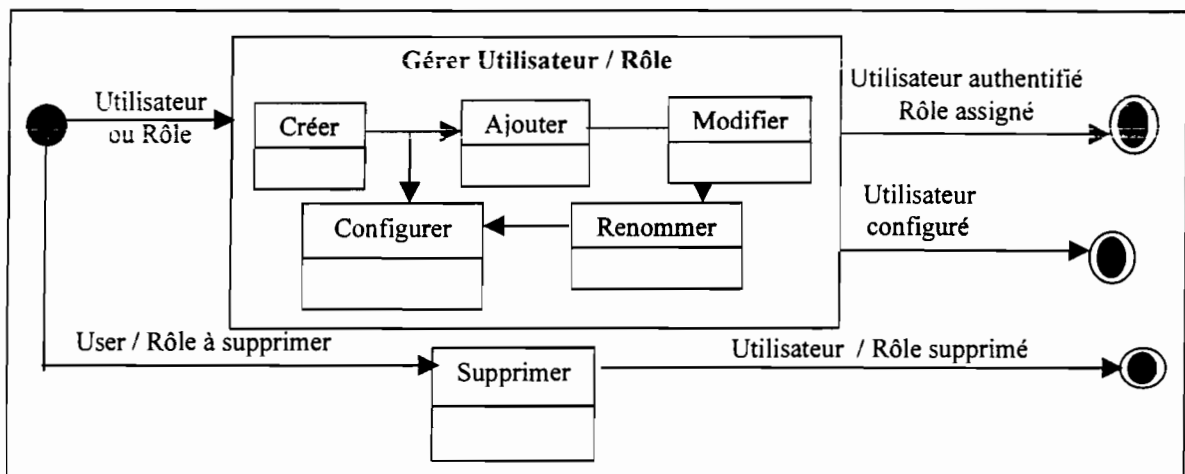


Figure 26 : Diagramme d'état – transition pour l'Administration des droits des utilisateurs et des groupes.

La description des fonctionnalités du système nous a fait remarquer l'existence de sous-ensembles communs à plusieurs fonctionnalités et le traitement particulier de certaines de ces "sous – fonctionnalités". Cette inclusion provoque la duplication de fonctionnalité. Pour parer à cela, nous devons étudier les relations possibles entre nos fonctionnalités.

UML définit deux (2) types de relations entre les fonctionnalités d'un système : la relation "UTILISE" pour un sous-ensemble commun et la relation "ETEND" pour les sous-ensembles nécessitant un traitement particulier. Pour notre système SIMES, elles sont représentées sous forme de graphe à la figure 28 au moyen des formalismes de la figure 27.

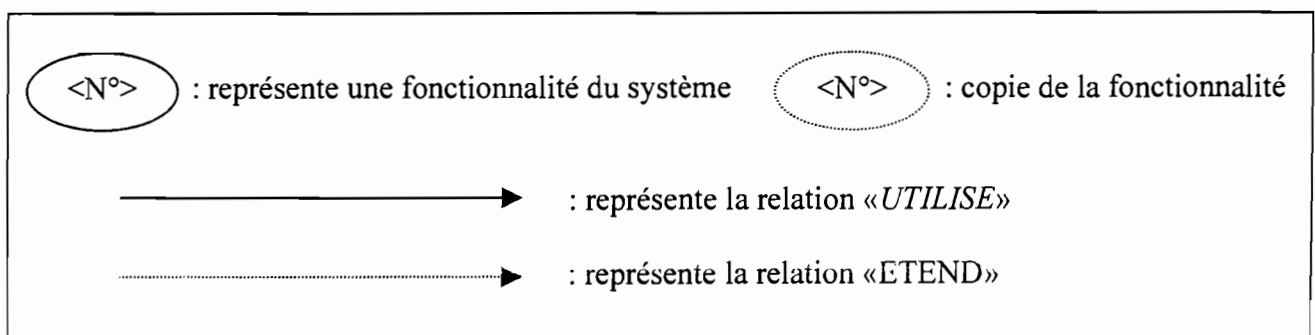
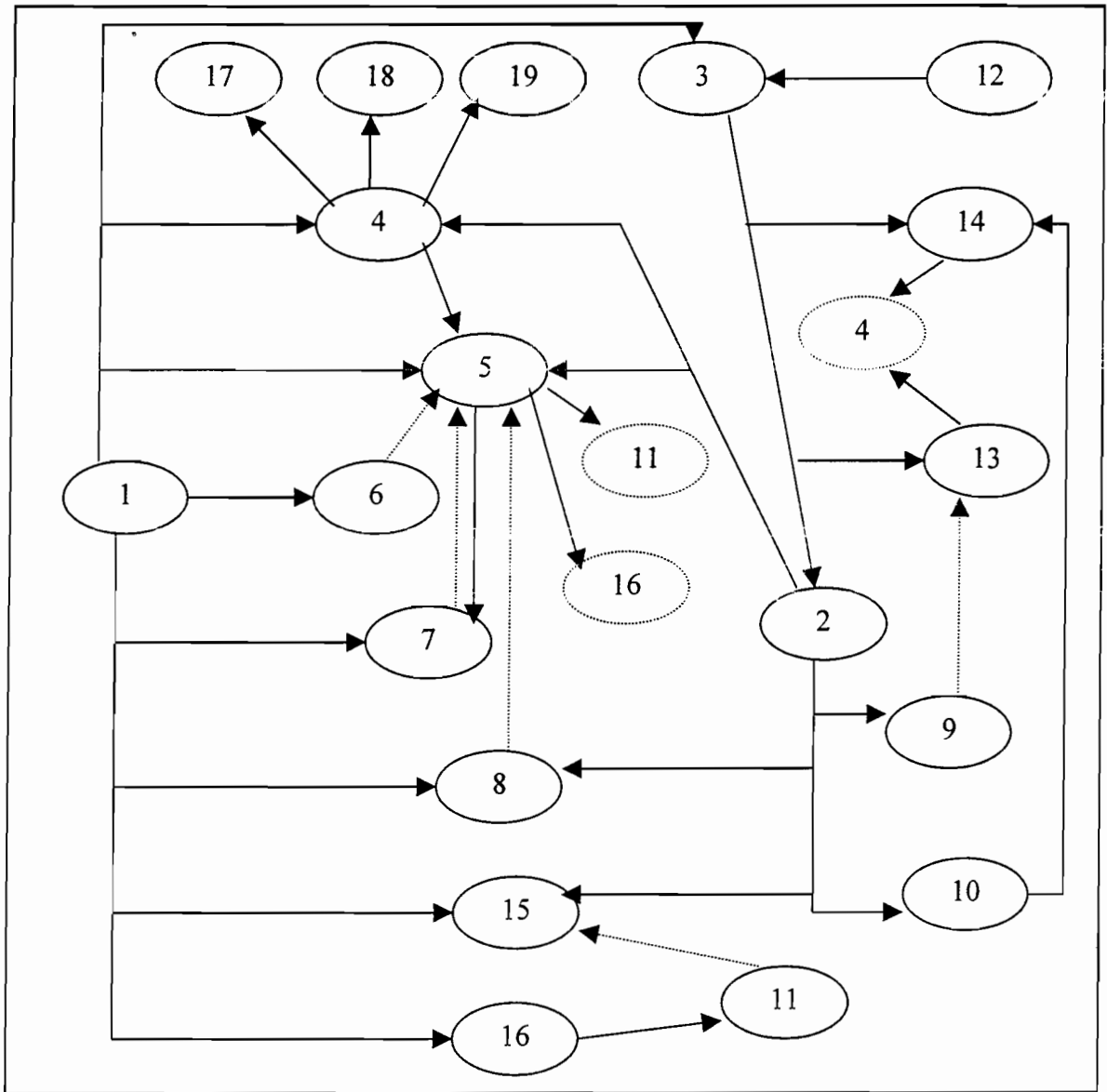


Figure 27 : Formalismes de représentation des relations entre les fonctionnalités du système.



- 1 : Administration du système d'information
- 2 : Administration d'une application
- 3 : Administration de l'entrepôt de données
- 4 : Traitement d'une donnée
- 5 : Accès utilisateur
- 6 : Nommage d'un ordinateur
- 7 : Nommage d'un utilisateur
- 8 : Nommage d'un artéfact
- 9 : Stockage d'un objet
- 10 : Consultation d'un objet
- 11 : Gestion d'une session
- 12 : Information d'un utilisateur
- 13 : Alimentation de l'entrepôt de données
- 14 : Visualisation de l'entrepôt de données
- 15 : Administration des paramètres système
- 16 : Administration des droits des utilisateurs et groupes
- 17 : Traitement d'une commande
- 18 : Exécution d'une activité
- 19 : Exécution d'un outil.

*Légende des fonctionnalités*

Figure 28 : Relations entre les fonctionnalités du système SIMES.

## II – 2 – 3 – 3 – Etude des travailleurs et des entités du système

Les travailleurs du système sont ceux qui interagissent, communiquent et travaillent ensemble pour exécuter les fonctionnalités du système.

Pour le système, nous pouvons retenir :

- l'Administrateur SIMES,
- le Membre SIMES,
- l'Invité.

Ces travailleurs manipulent des objets du système : ce sont les entités du système. Dans le cas de cette étude les entités suivantes peuvent être définies :

- *Activité,*
- *Artéfact,*
- *Bureau,*
- *Entrepôt de données,*
- *Fenêtre,*
- *Profil utilisateur,*
- *Rôle,*
- *Service,*
- *Session,*
- *Utilisateur.*

Ces entités sont considérées comme les différentes classes d'étude du système SIMES permettant l'exécution des fonctionnalités du système. Ce sont des redéfinitions des concepts prédéfinis pour mieux cibler le système. Elles possèdent des attributs permettant de compléter leur définition dans le système. Leur représentation est faite par la figure 30 à l'aide des formalismes de représentation de la figure 29.

Une classification des classes (travailleurs et entités) d'un système est aussi possible selon des critères qu'UML appelle les « Stéréotypes ». Dans notre système on en dénombre trois (3) à savoir le travailleur d'interface, le travailleur interne et l'entité.

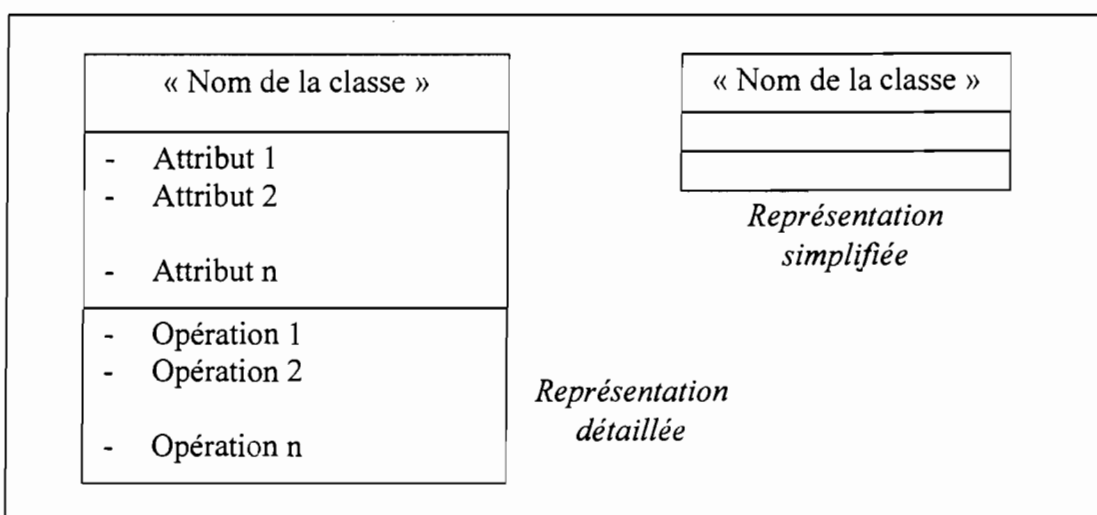


Figure 29 : Formalisme de représentation d'une classe.

Figure 30 : Les différentes classes du système SIMES

Activité
<ul style="list-style-type: none"> <li>- Nom</li> <li>- Etat</li> <li>- Expression</li> <li>- Description</li> </ul>
<ul style="list-style-type: none"> <li>- Décrire</li> <li>- Démarrer</li> <li>- Suspendre</li> <li>- Reprendre</li> <li>- Terminer</li> </ul>

Artéfact
<ul style="list-style-type: none"> <li>- Nom</li> <li>- Type</li> <li>- Format de représentation</li> <li>- Situation artéfact</li> <li>- Lien physique</li> </ul>
<ul style="list-style-type: none"> <li>- Créer</li> <li>- Modifier</li> <li>- Supprimer</li> </ul>

Bureau
<ul style="list-style-type: none"> <li>- Liste de fenêtres</li> <li>- Liste des objets</li> </ul>
<ul style="list-style-type: none"> <li>- Afficher</li> <li>- Restaurer</li> <li>- Insérer objet</li> <li>- Retirer objet</li> </ul>

Espace de travail
<ul style="list-style-type: none"> <li>- Nom</li> <li>- Configuration</li> </ul>
<ul style="list-style-type: none"> <li>- Définir</li> <li>- Modifier</li> </ul>

Fenêtre
<ul style="list-style-type: none"> <li>- Nom</li> <li>- Application</li> <li>- Contenu fenêtre</li> <li>- Position</li> <li>- Dimension</li> </ul>
<ul style="list-style-type: none"> <li>- Sauvegarder</li> <li>- Modifier</li> <li>- Repositionner</li> <li>- Redimensionner</li> </ul>

Profil utilisateur
<ul style="list-style-type: none"> <li>- Répertoire d'initialisation</li> <li>- Script d'initialisation</li> <li>- Paramètres environnement</li> <li>- Données d'application</li> <li>- Données temporaires</li> </ul>
<ul style="list-style-type: none"> <li>- Définir le profil</li> <li>- Modifier le profil</li> <li>- Charger le profil</li> </ul>

Rôle
<ul style="list-style-type: none"> <li>- Nom rôle</li> <li>- Description du rôle</li> <li>- TypeObjet associé</li> </ul>
<ul style="list-style-type: none"> <li>- Créer</li> <li>- Assigné rôle</li> <li>- Définir droit</li> <li>- Changer de propriétaire</li> </ul>

Service
<ul style="list-style-type: none"> <li>- Nom service</li> <li>- Type service</li> <li>- Type démarrage</li> <li>- Chemin d'exécution</li> <li>- Arguments</li> <li>- Nom d'exécution</li> </ul>
<ul style="list-style-type: none"> <li>- Lancer</li> <li>- Tuer</li> <li>- Suspendre</li> </ul>

Session
<ul style="list-style-type: none"> <li>- Identifiant utilisateur</li> <li>- Ordinateur utilisateur</li> <li>- Date ouverture session</li> <li>- Temps d'inaction</li> <li>- Invité</li> </ul>
<ul style="list-style-type: none"> <li>- Démarrer</li> <li>- Déconnecter</li> </ul>

Utilisateur
<ul style="list-style-type: none"> <li>- Nom utilisateur</li> <li>- Prénom utilisateur</li> <li>- Login</li> <li>- Mot de passe</li> <li>- E-mail</li> </ul>
<ul style="list-style-type: none"> <li>- Créer un utilisateur</li> <li>- Changer le mot de passe</li> <li>- Affecter à un groupe</li> </ul>

La modélisation complète de ces éléments se fait grâce à un diagramme de classes. A ce stade d'étude sa représentation serait incomplète car nous n'avons pas encore étudié en profondeur les relations existant entre ces classes. *Le diagramme de classe sera établi après l'étude des interactions ou des collaborations entre les éléments du système.*

Néanmoins, la classification peut se faire comme suit :

L'Administrateur SIMES, le Membre et l'Invité sont des travailleurs d'interface. Ce sont les travailleurs qui sont visible du point de vue d'un acteur.

Profil utilisateur, Bureau, Session et Espace de travail sont des travailleurs internes. Ils ne sont pas visible du point de vue d'un acteur.

Utilisateur, Fenêtre, Activité, Service, Rôle, Artéfact et Entrepôt de données sont les entités. Elles sont associées à toute entité manipulée dans le système par les travailleurs.

## II – 2 – 3 – 4 – Etude des Workflows des fonctionnalités

Cette étape de l'étude de modélisation en UML porte sur la description des flux d'évènements et les activités des fonctionnalités. Le flux d'évènements est appelé Workflow. La description complète du workflow d'une fonctionnalité donne lieu à deux diagrammes d'interaction fondés sur les mêmes informations : le diagramme de séquence et le diagramme de collaboration.

Un diagramme de séquence présente les interactions entre les objets, arrangées en séquences dans le temps. En particulier, il montre les objets participant dans l'interaction par leur «lignes de vie» et les messages qu'ils échangent dans le temps. Il ne montre pas les associations entre les objets.

Un diagramme de collaboration montre les interactions entre les objets et leurs liens. Contrairement au diagramme de séquence, un diagramme de collaboration montre les relations entre les objets en n'intégrant pas la dimension du temps.

Le diagramme de collaboration permet de définir les relations entre les classes. S'il existe un message entre deux objets, il existe alors une relation entre leurs classes respectives. De cet fait, nous n'étudierons que les collaborations entre les objets du système, à l'aide des formalismes de la figure 31. L'ensemble des interactions sous forme de collaboration entre les éléments du système SIMES est présenté par fonctionnalité de la figure 32 à la figure 46.

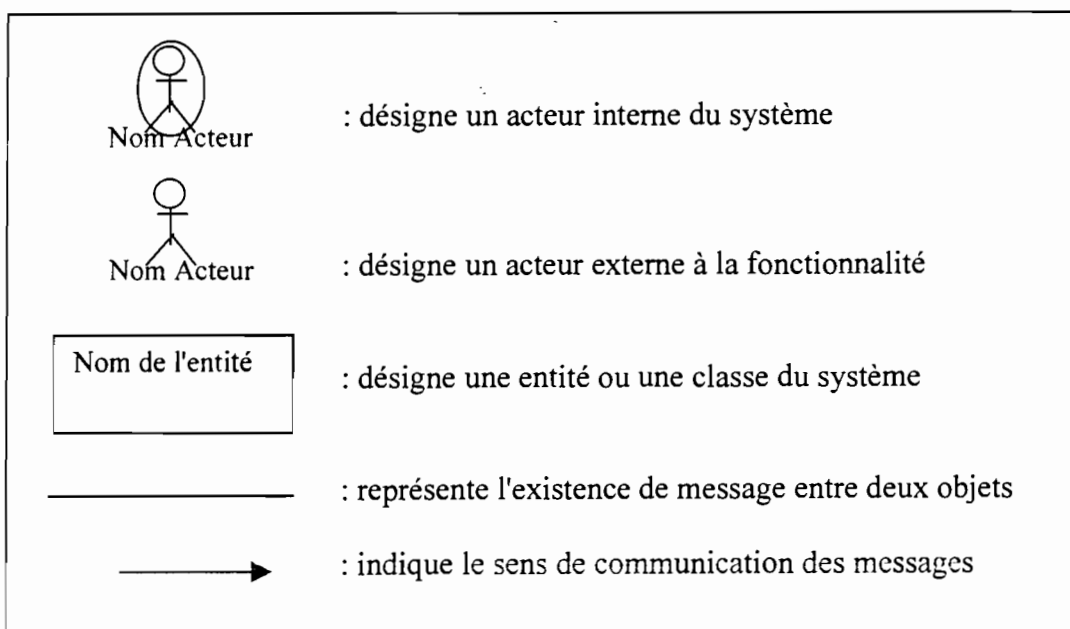


Figure 31 : Formalismes de représentation des diagrammes de collaboration

- ✓ **Administration de l'entrepôt de données**
  - Gérer les sauvegardes
  - Organiser l'entrepôt, Gérer les artéfacts

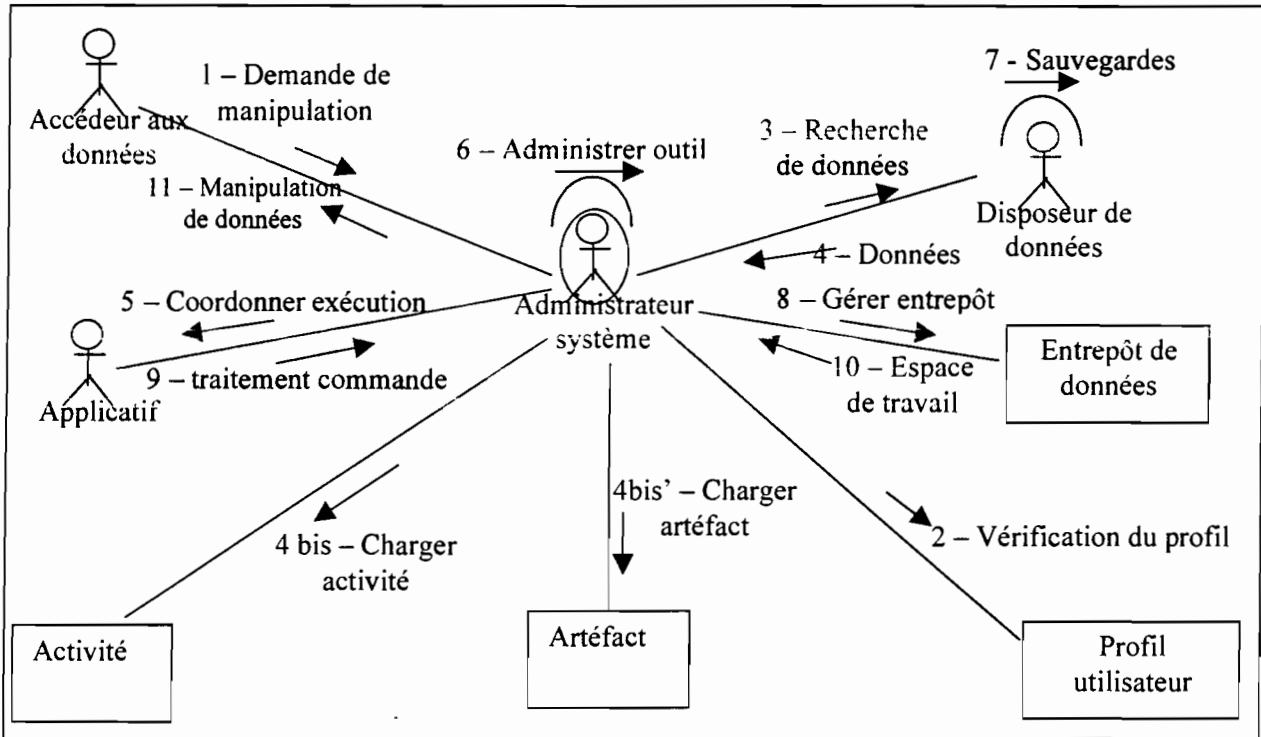


Figure 32 : Diagramme de collaboration pour l'Administration de l'entrepôt de données.

- ✓ **Administration d'une application**
  - Définir nouvelles applications
  - Ecrire leurs sources de données, Ecrire leur mode d'emploi.

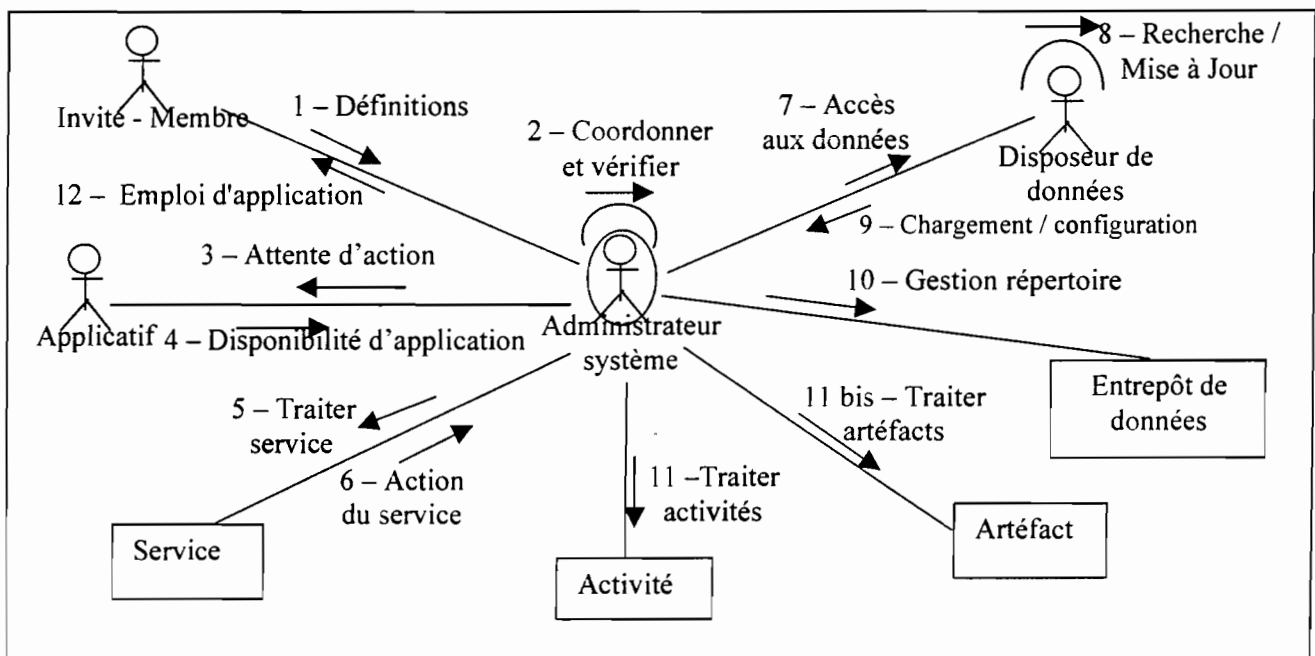


Figure 33 : Diagramme de collaboration pour l'Administration d'une application.

✓ **Alimentation de l'entrepôt de données**

- Déposer un document
- Décrire un outil
- Décrire une activité

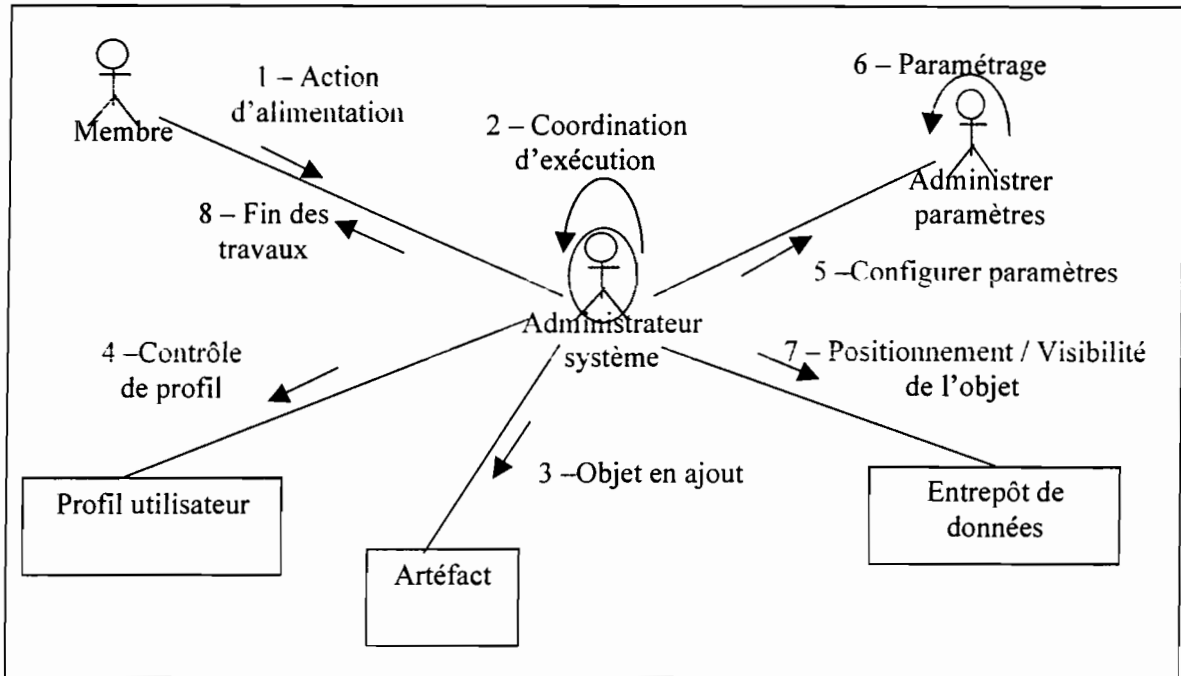


Figure 34 : Diagramme de collaboration pour l'Alimentation de l'entrepôt de données.

✓ **Visualisation de l'entrepôt de données**

- Accéder au plan du site
- Ouvrir un document

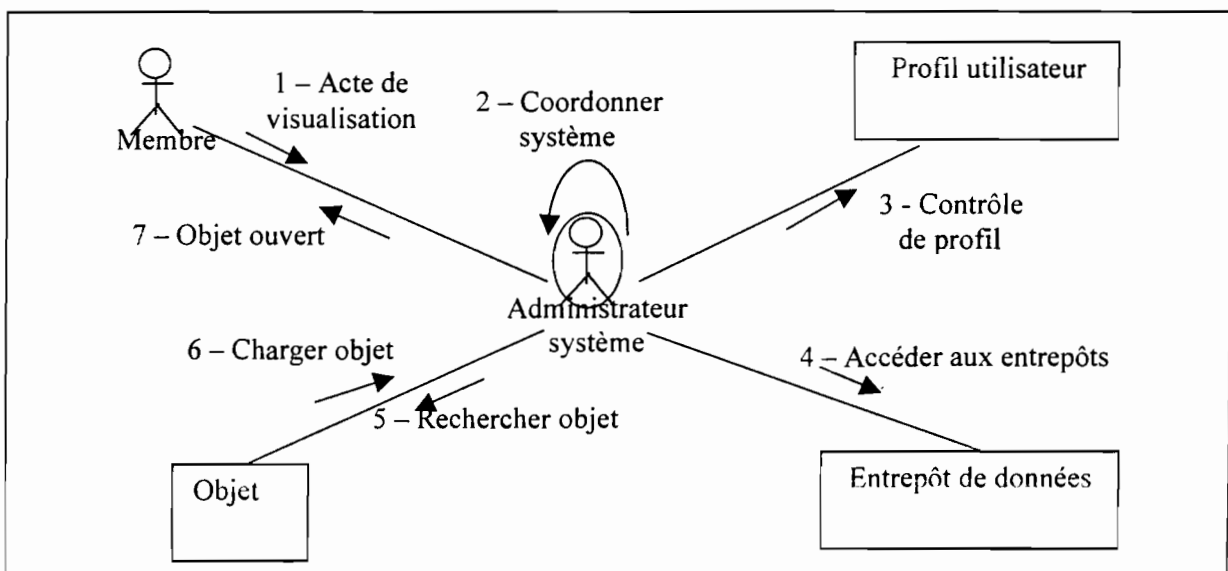


Figure 35 : Diagramme de collaboration pour la Visualisation de l'entrepôt de données.



✓ **Traitement / Exécution d'un objet**

*Lancer une commande. Exécuter un outil. Exécuter une activité .*

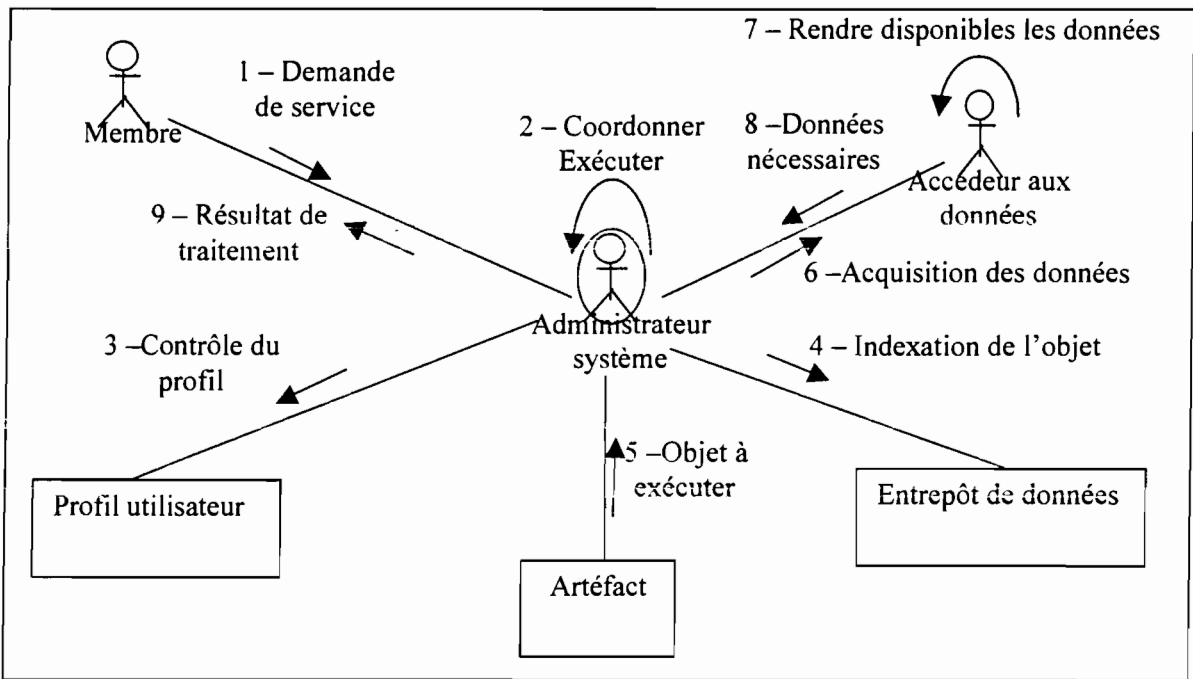


Figure 36 : Diagramme de collaboration pour le Traitement et l'exécution d'un objet.

✓ **Stockage d'un objet**

*Décrire l'objet, Transférer l'objet (contenu, description, indice du répertoire), Rendre visible / invisible l'objet*

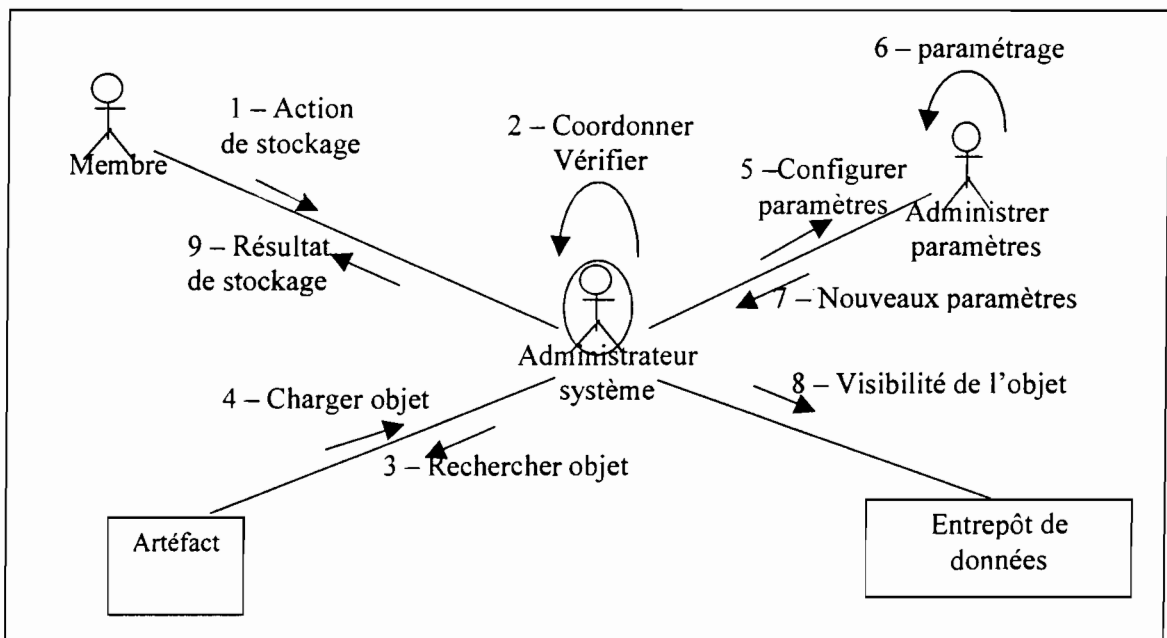


Figure 37 : Diagramme de collaboration pour le Stockage d'un objet.

✓ **Consultation d'un objet**

*Accéder aux objets: Obtenir contenu et description.*

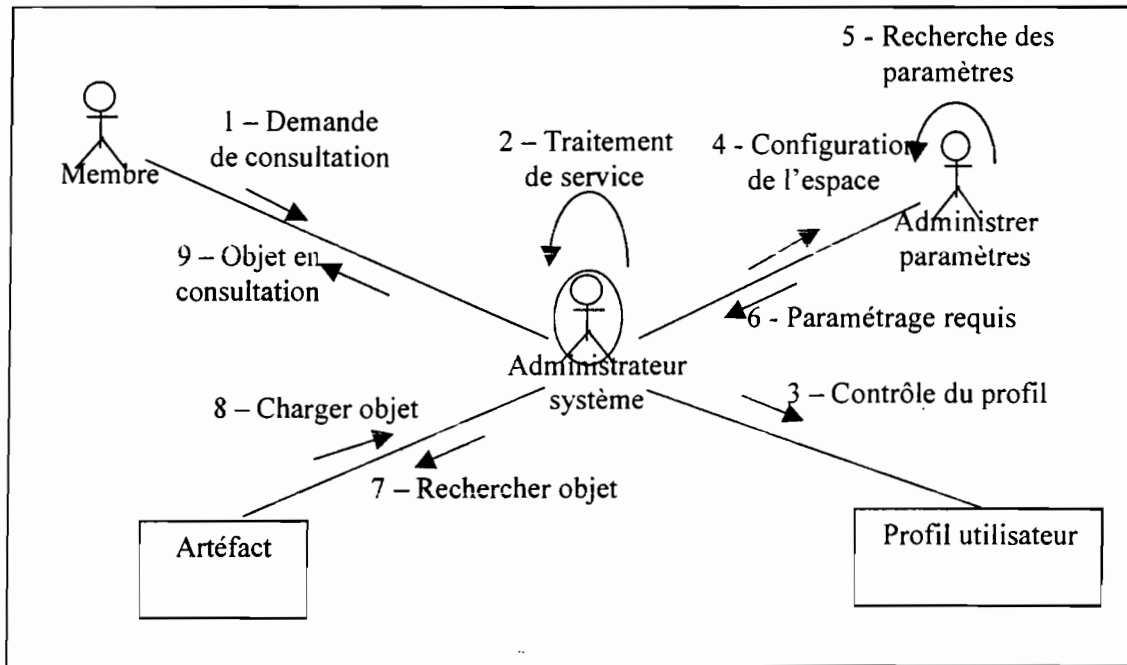


Figure 38 : Diagramme de collaboration pour la Consultation d'un objet.

✓ **Nommage d'un ordinateur**

- Décrire ordinateur,
- Identifier ordinateur,
- Accéder ordinateur

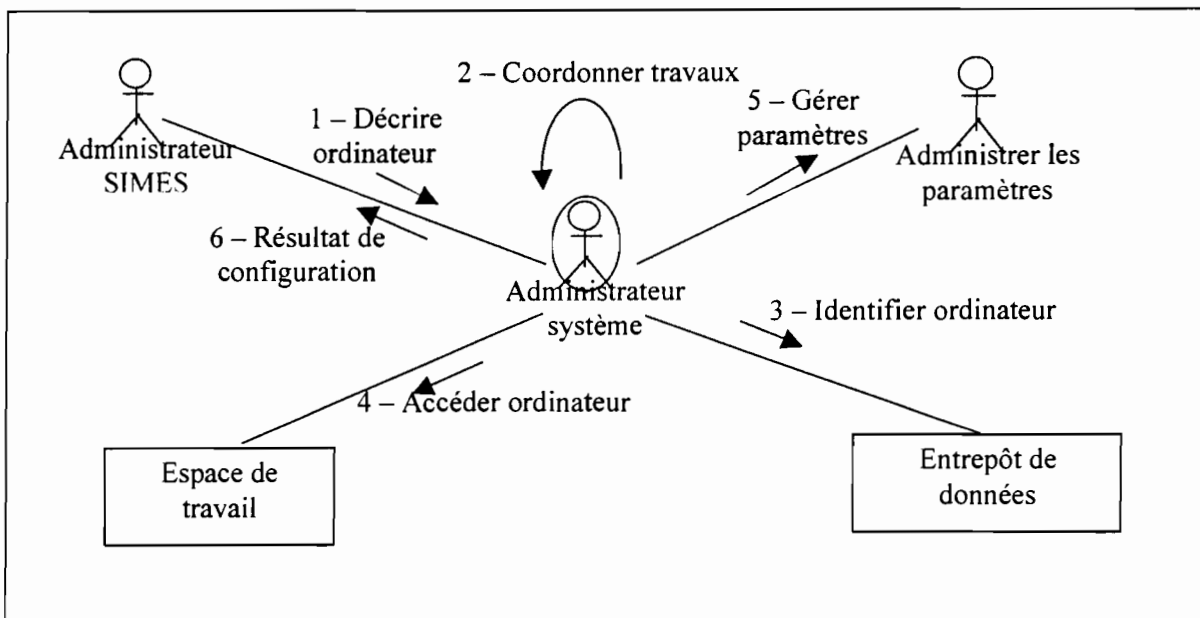


Figure 39 : Diagramme de collaboration pour le Nommage d'un ordinateur.

✓ **Nommage d'un utilisateur**

- Décrire utilisateur,
- Identifier utilisateur,
- Accéder utilisateur

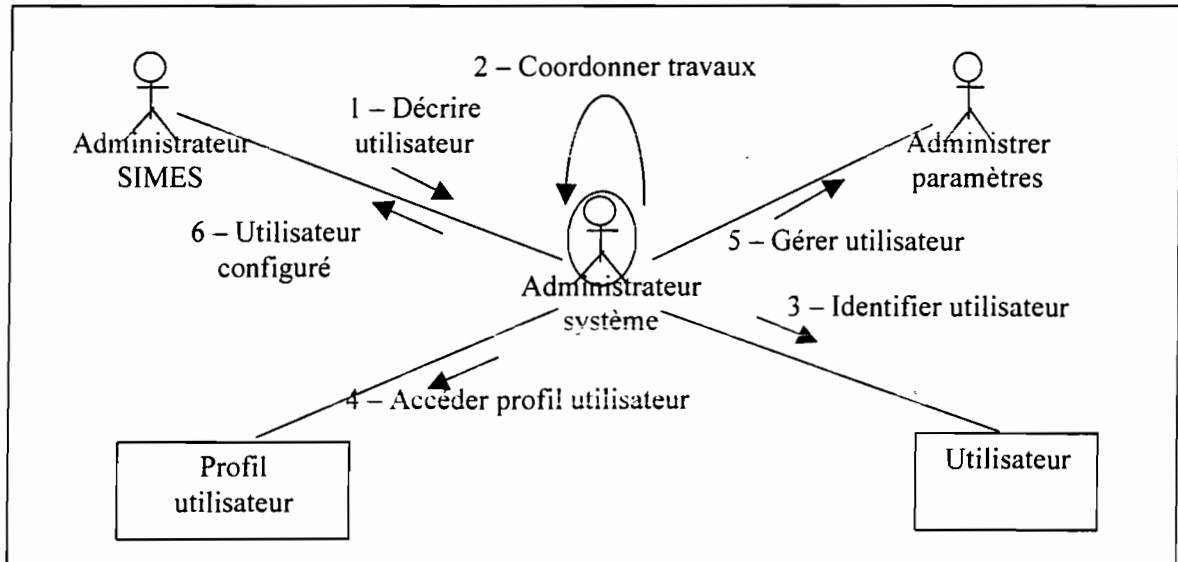


Figure 40 : Diagramme de collaboration pour le Nommage d'un utilisateur.

✓ **Nommage d'un artéfact**

- Décrire artéfact,
- Identifier artéfact,
- Accéder artéfact

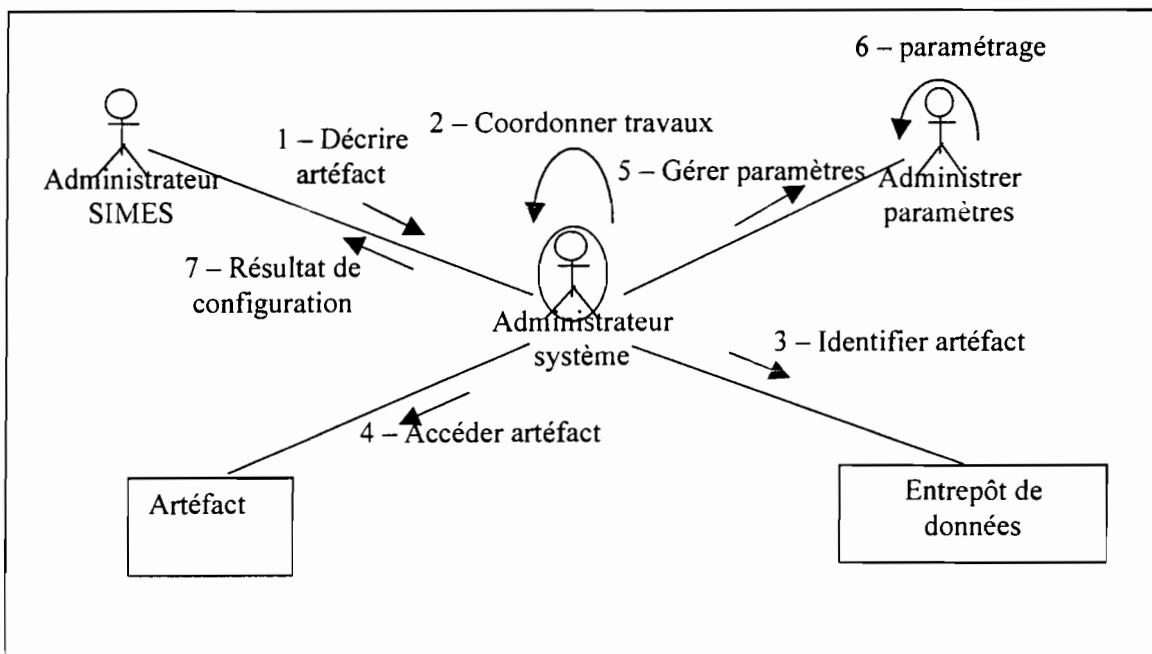


Figure 41 : Diagramme de collaboration pour le Nommage d'un artéfact.

✓ **Administration du système d'information**

- Gérer le calendrier des évènements
- Gérer les thèmes et messages du forum
- Gérer les listes de diffusion et les messages archivés
- Gérer les thèmes de la rubrique, de la bibliothèque

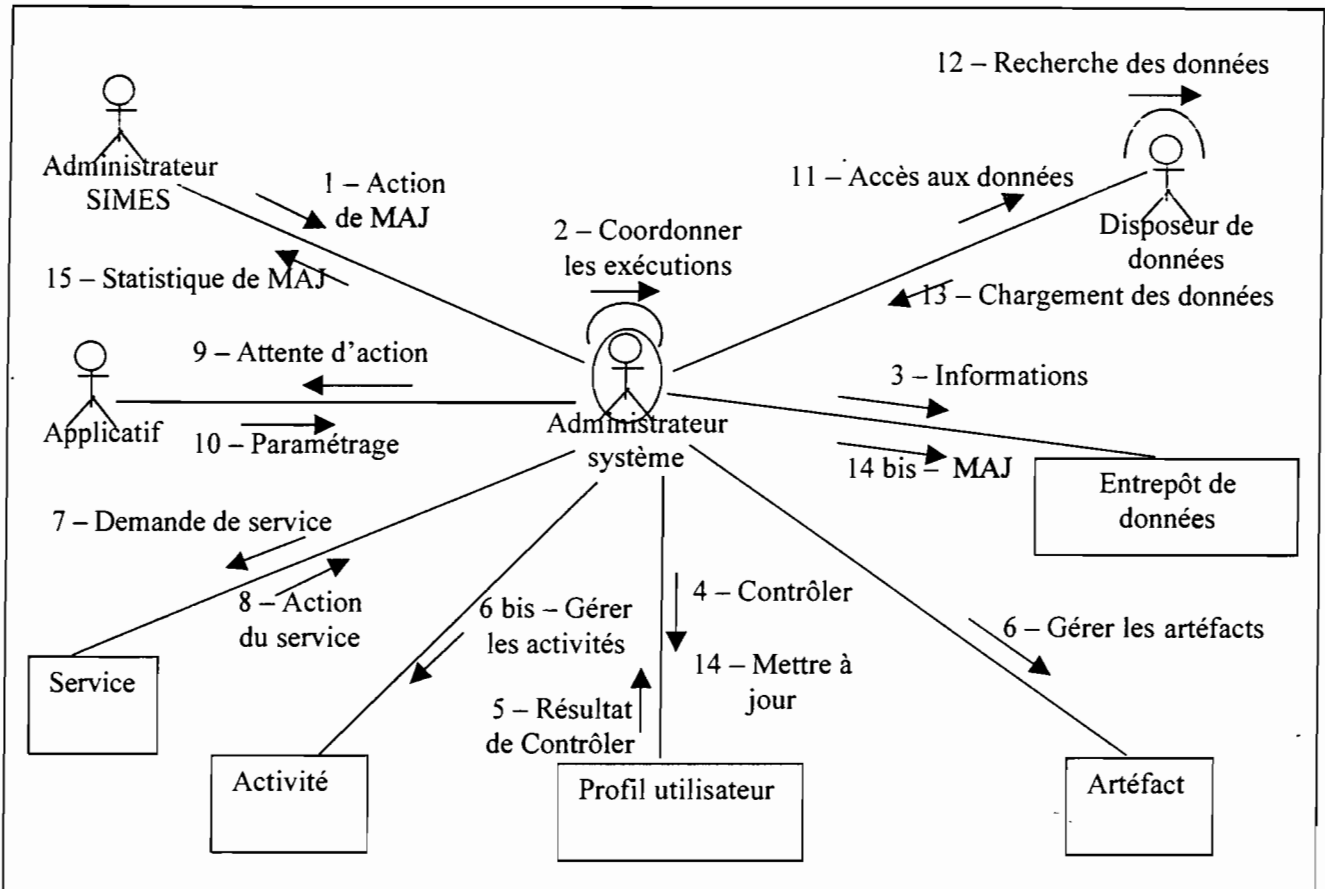


Figure 42 : Diagramme de collaboration pour l'Administration du système d'information.

✓ **Gestion d'une session**

*Définir les descriptifs. Coordonner l'exécution. Détruire les descriptifs.*

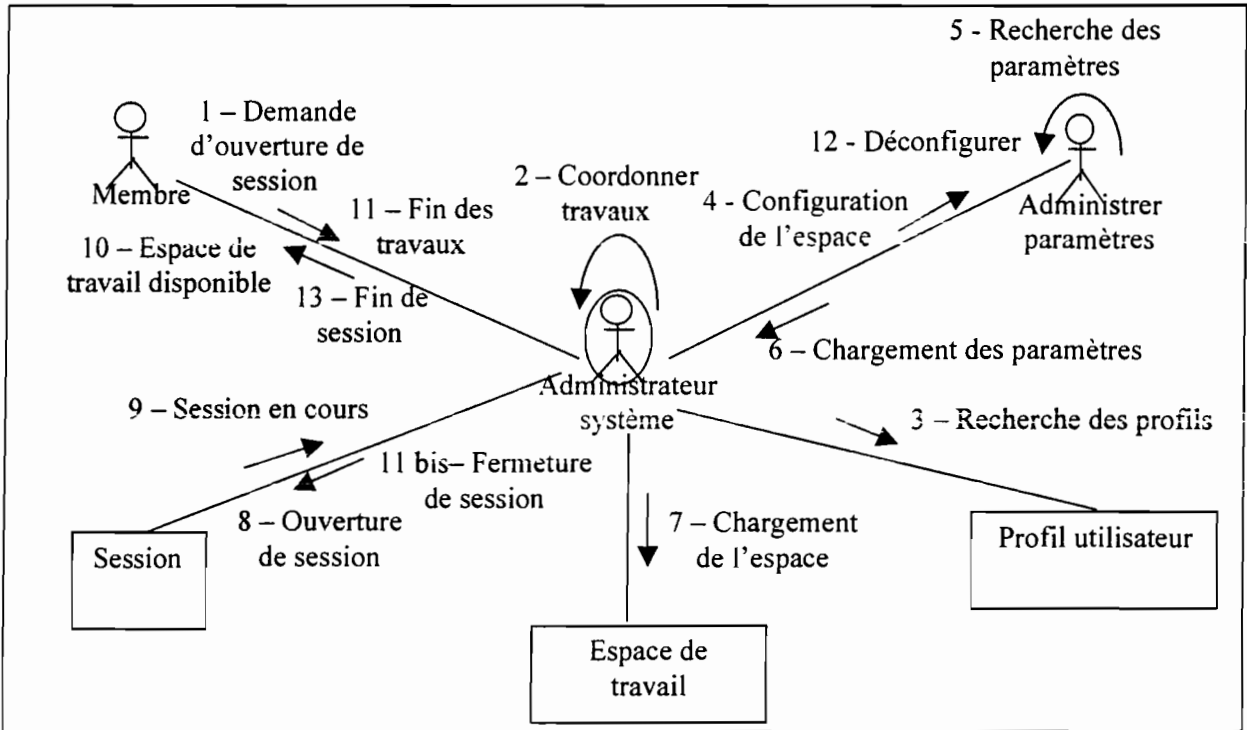


Figure 43 : Diagramme de collaboration pour la Gestion d'une session.

✓ **Information d'un utilisateur**

*Rechercher les informations, Rendre visibles les informations.*

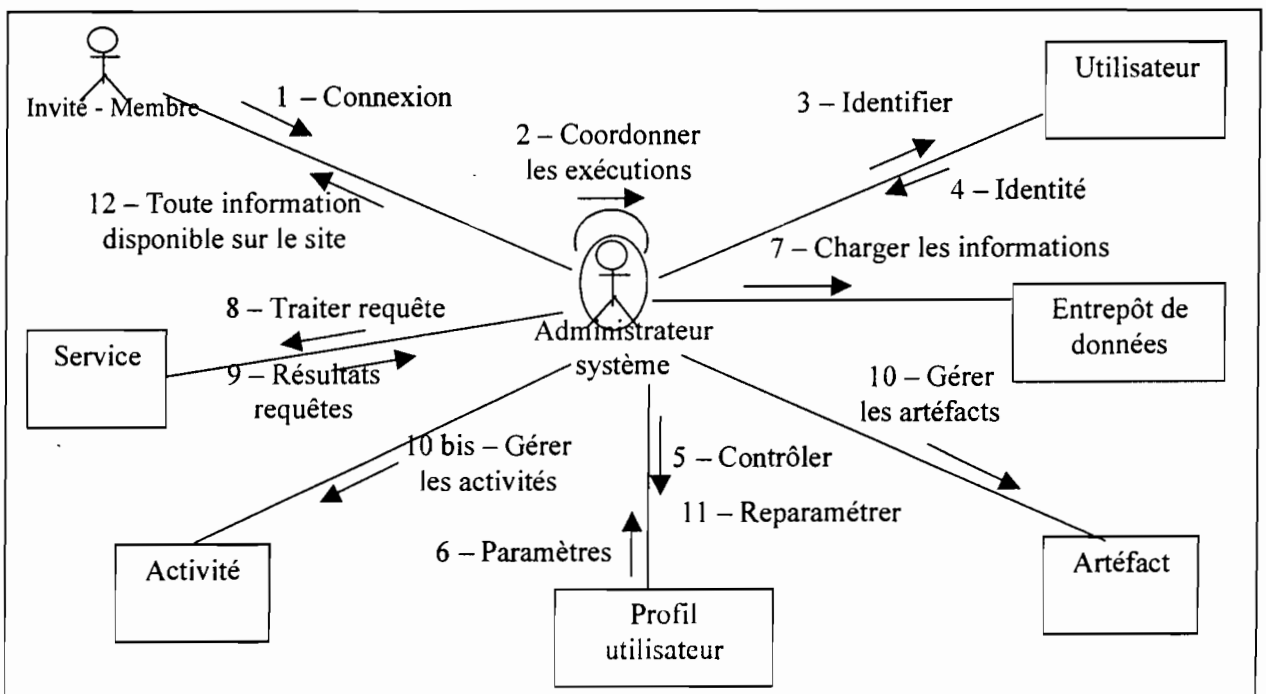


Figure 44 : Diagramme de collaboration pour la Gestion d'une session.

✓ **Accès utilisateur**

- Se connecter
- Changer de mot de passe,
- Se déconnecter.

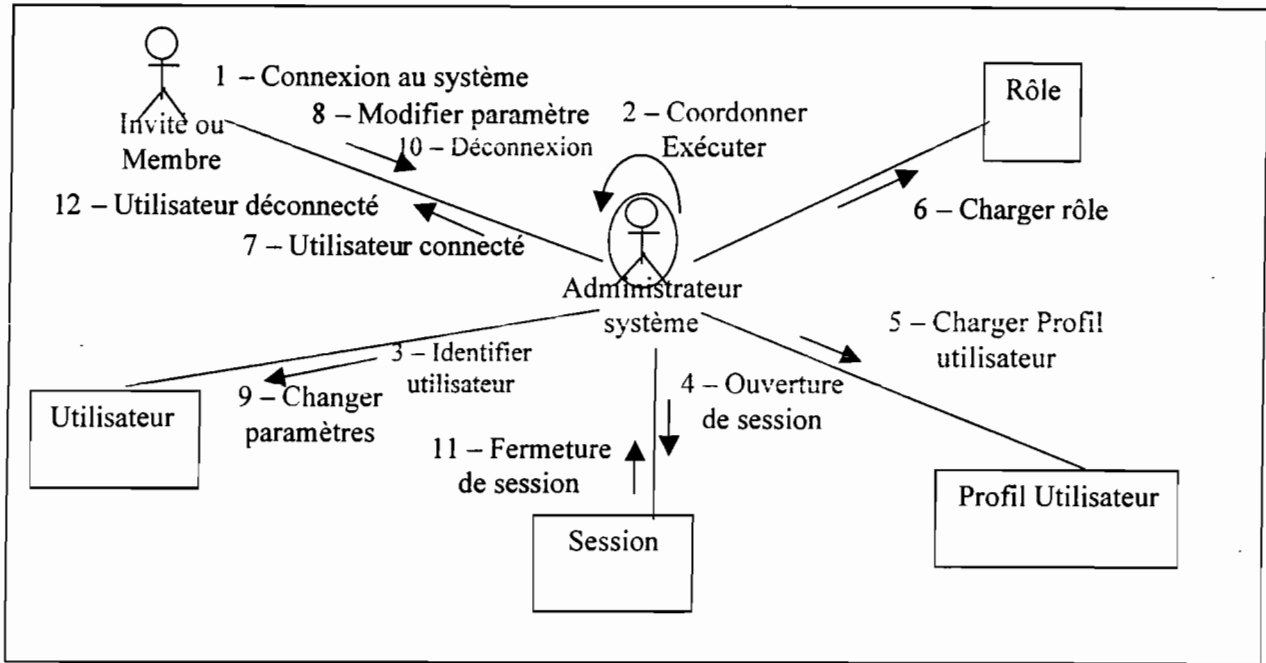


Figure 45 : Diagramme de collaboration pour l'Accès Utilisateur.

✓ **Administration des droits des utilisateurs et groupes**

- Gérer les utilisateurs
- Gérer les rôles

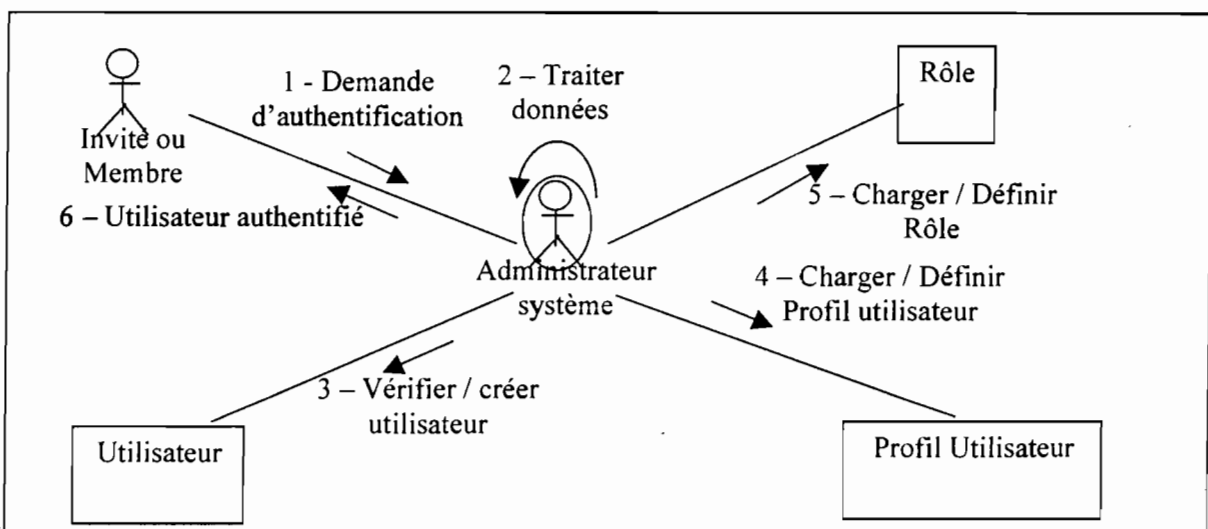


Figure 46 : Diagramme de collaboration pour l'Administration des droits des utilisateurs et groupes.

Comme il est possible de déduire des relations entre les objets à partir du diagramme de collaboration, les liens entre objets pourront être considérés comme des instances de relations entre classes.

Il existe plusieurs types de relations entre classes : *l'association, l'agrégation, la classe - association, la généralisation et la dépendance.*

L'association est la relation la plus courante et la plus riche du point de vue sémantique. Une association est une relation statique n-aire (même si le plus souvent, l'association est binaire) c'est-à-dire qu'elle relie plusieurs classes entre-elles. L'association existe entre les classes et non entre les instances ; elle est introduite pour montrer une structure et non pas des échanges de données. Une association n-aire possède n rôles qui constituent les points terminaux ou terminaisons de l'association : c'est-à-dire le point où l'association s'accroche à la classe qui se trouve au bout de l'association. Un rôle montre en fait la tâche d'une classe dans une association. Il est porteur de toute information qui relie l'association à la classe.

L'agrégation n'est pas un autre type de relation, mais une variante de l'association ou plus exactement, un attribut sur un seul des rôles d'une association. L'agrégation définit une relation « partie de » entre instances de classes. UML définit deux (2) formes d'agrégation : la composition stricte et l'agrégation partagée. La composition stricte est une forme forte d'agrégation qui requiert qu'une instance appartienne au plus à une instance composée à un instant donné : cette appartenance peut changer dans le temps. Une agrégation partagée est à l'inverse de la composition, une agrégation qui autorise qu'une instance composante fasse partie de plusieurs objets composites en même temps.

La Classe-association définit une association qui est en même temps une classe. Non seulement elle connecte des classes entre elles mais elle possède aussi un ensemble de fonctionnalités propres qui n'appartiennent à aucune des autres classes mais qui sont liées à l'existence de l'association. La classe-association se comporte comme une vraie classe avec une structure, un comportement et d'autres relations. La classe-association n'est pas équivalente à une relation ternaire. La multiplicité de la classe-association est implicite dans l'association.

La généralisation de classe définit une relation de classification entre une classe plus générale et une classe plus spécifique. La classe plus spécifique (appelée aussi classe fille, classe dérivée, classe spécialisée, classe descendante, ...) est cohérente avec la classe plus générale (appelée aussi classe mère, classe de base, classe générale, ...), c'est-à-dire qu'elle contient par héritage tous les attributs, les membres, les relations de la classe générale, et peut en contenir d'autres. La généralisation existe dans la description statique. Les relations de généralisation peuvent donc être vues de deux (2) manières : la généralisation et la spécialisation.

La dépendance est une relation qui existe entre les éléments de modélisation sans nécessairement requérir des instances. C'est une relation unidirectionnelle qui signifie que le changement de la cible peut impliquer un changement dans la source. Plusieurs formes de dépendances sont prédéfinies dans UML : la Trace, le Raffinement et l'Utilisation.

Pour décrire ces différentes relations on aura recours aux formalismes présentés à la figure 47 et beaucoup utilisés dans les diagrammes de classes.

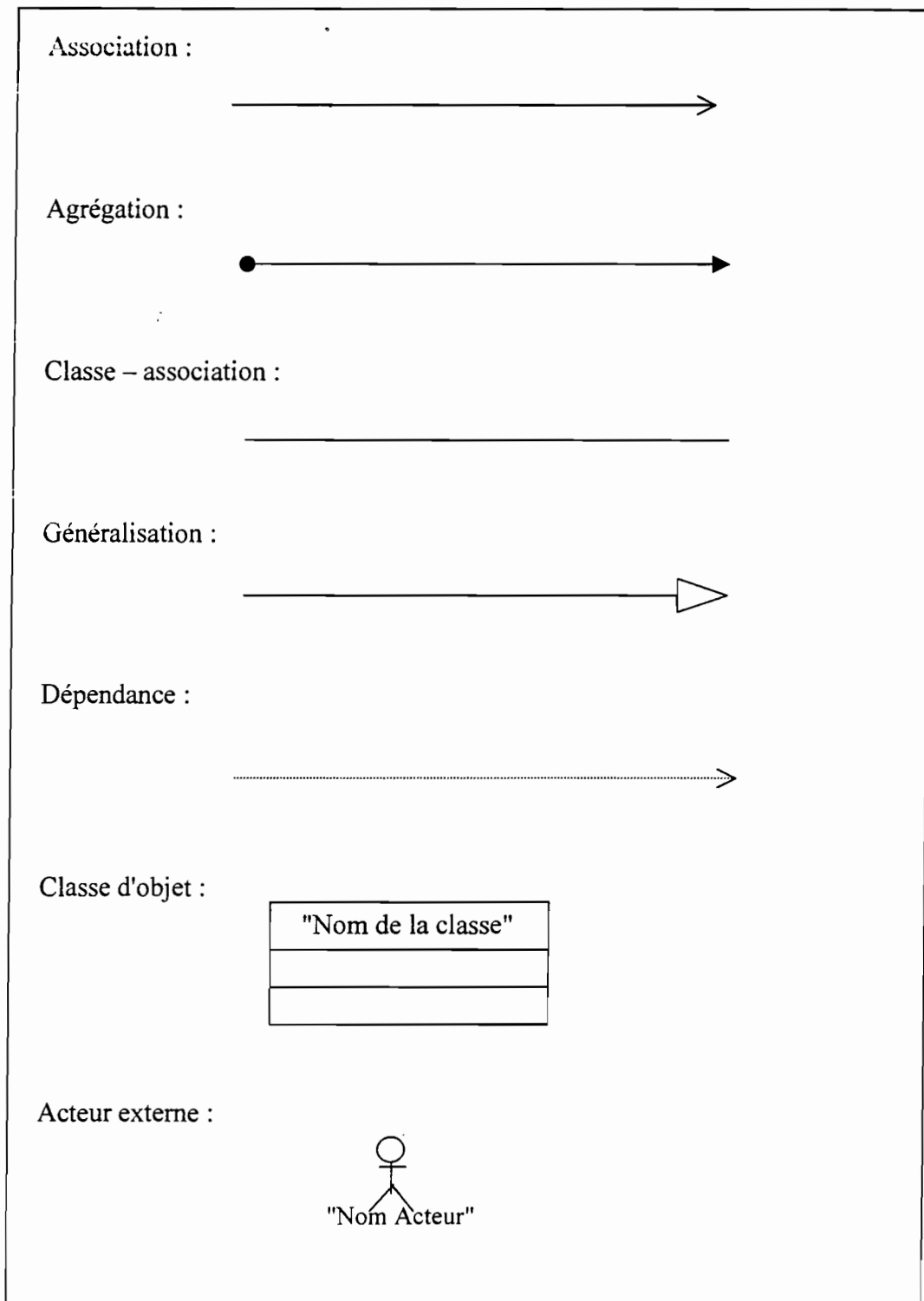


Figure 47 : Eléments de représentation des relations dans les modèles.



Un diagramme de classes est une collection d'éléments statiques de modèle, tels que des classes, des interfaces et leurs relations, connectées entre elles comme un graphe.

Un diagramme de classes montre uniquement les aspects statiques du modèle et fait abstraction des aspects dynamiques ou temporels, même si les éléments du diagramme de classe peuvent avoir un comportement dynamique important.

Un diagramme de classes peut être considéré comme une coupe particulière dans la perspective logique de l'architecture du système.

L'illustration de ses représentations par fonctionnalité au niveau de notre système est présentée à l'aide des figures 48 à 65.

✓ **Administration de l'entrepôt de données**

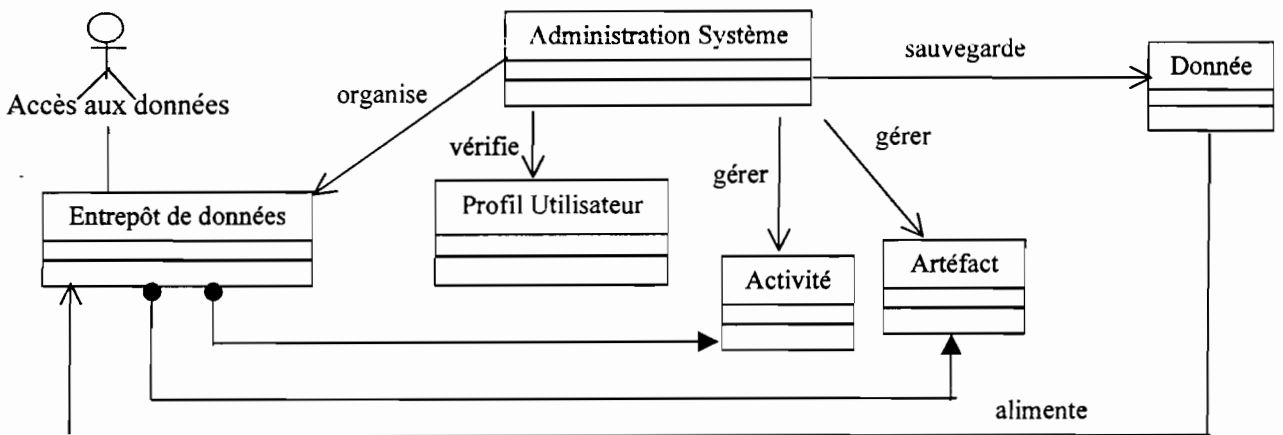


Figure 48 : Diagramme de classes pour l'Administration de l'entrepôt de données.

✓ **Administration d'une application**

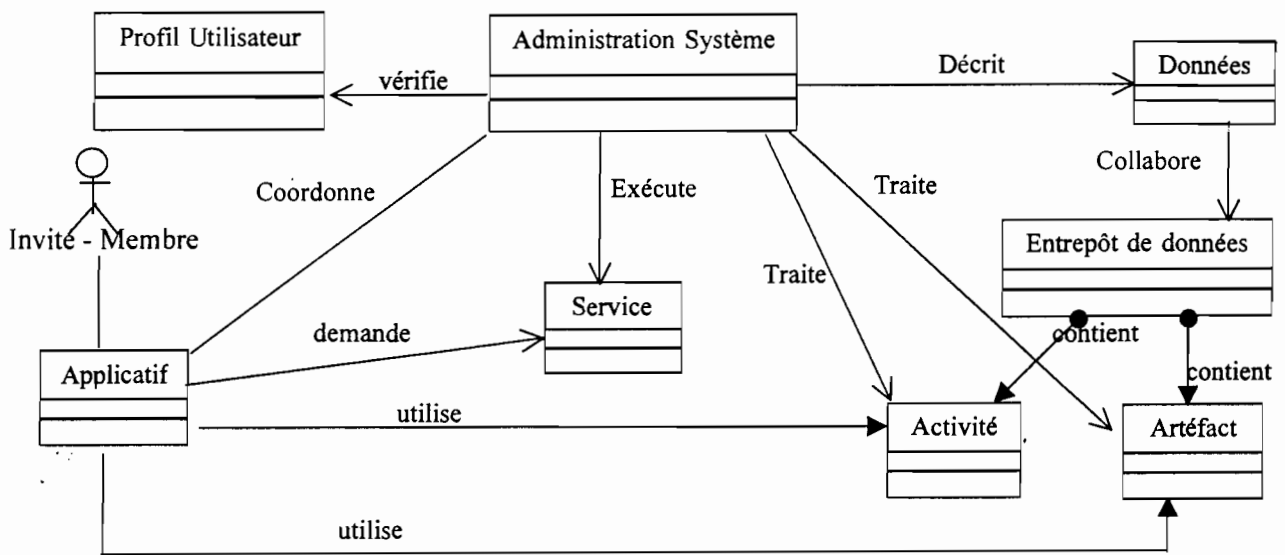


Figure 49 : Diagramme de classes pour l'Administration d'une application.

✓ **Alimentation de l'entrepôt de données**

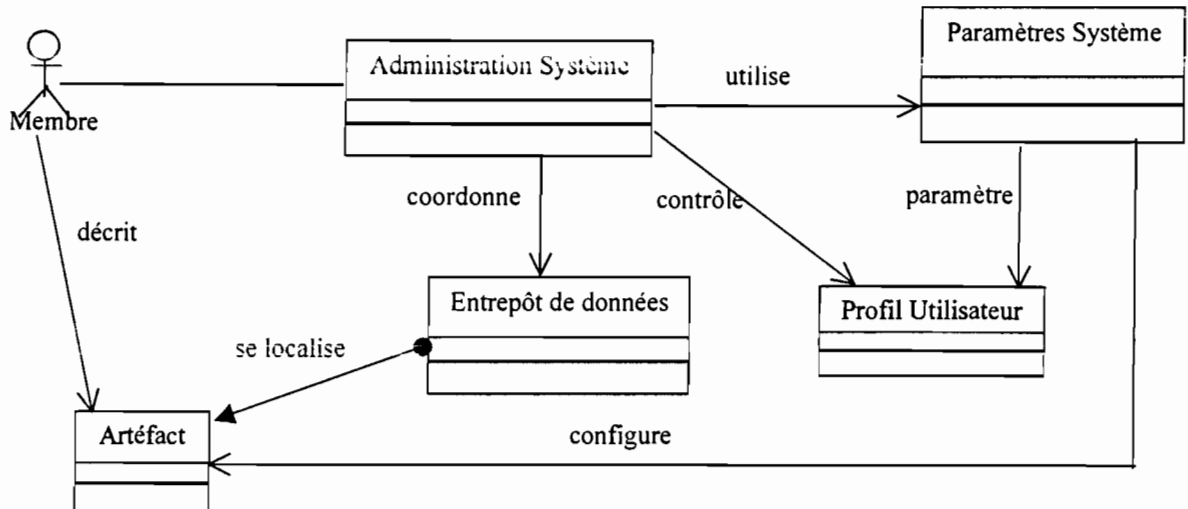


Figure 50 : Diagramme de classes pour l'Alimentation de l'entrepôt de données.

✓ **Visualisation de l'entrepôt de données**

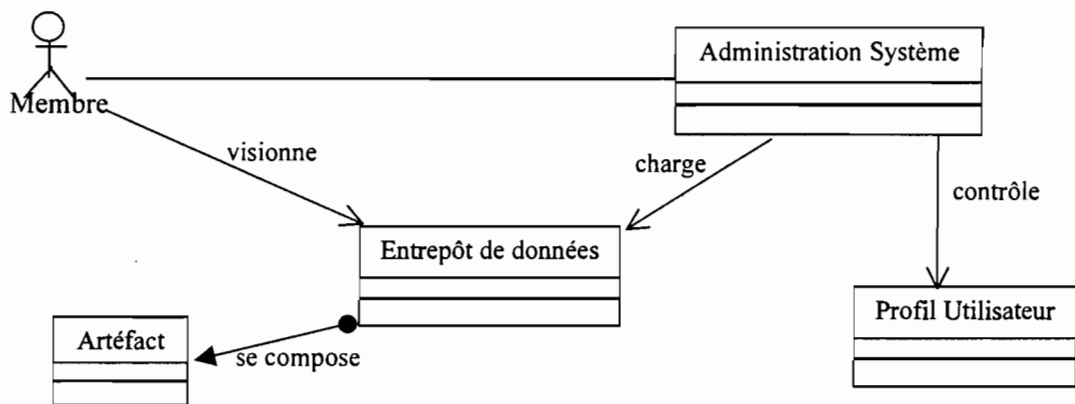


Figure 51 : Diagramme de classes pour la Visualisation de l'entrepôt de données.

✓ **Nommage d'un ordinateur**

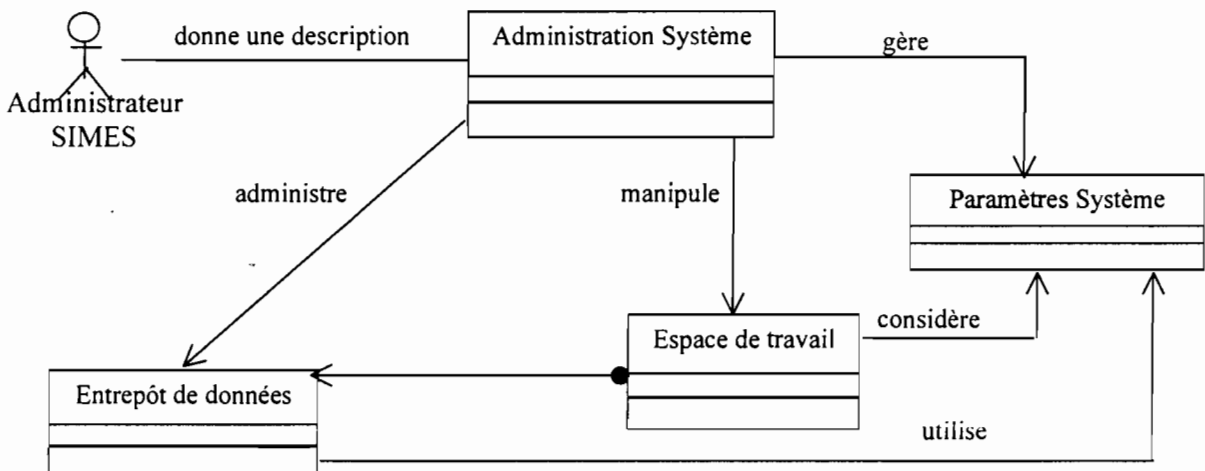


Figure 52 : Diagramme de classes pour le Nommage d'un ordinateur.

✓ **Nommage d'un utilisateur**

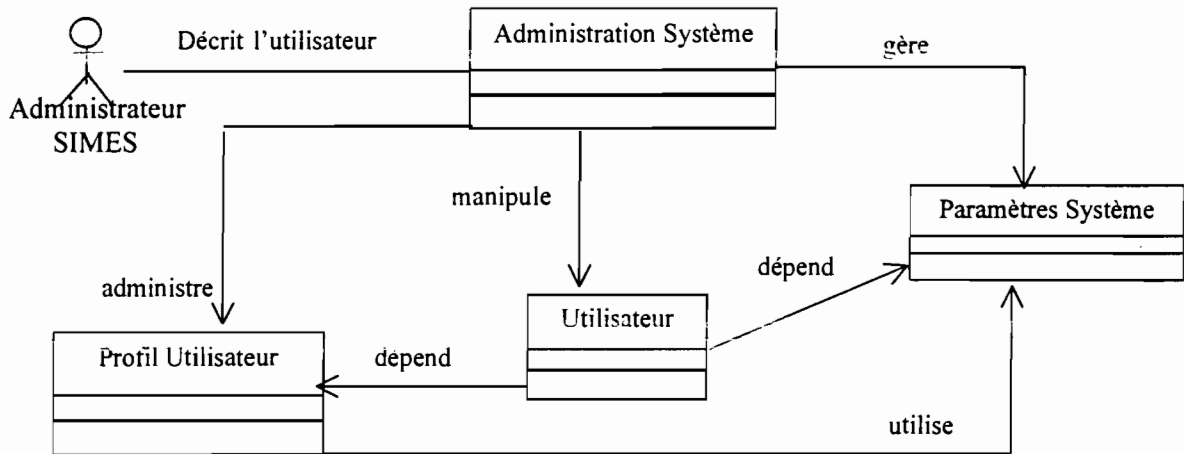


Figure 53 : Diagramme de classes pour le Nommage d'un utilisateur.

✓ **Nommage d'un artéfact**

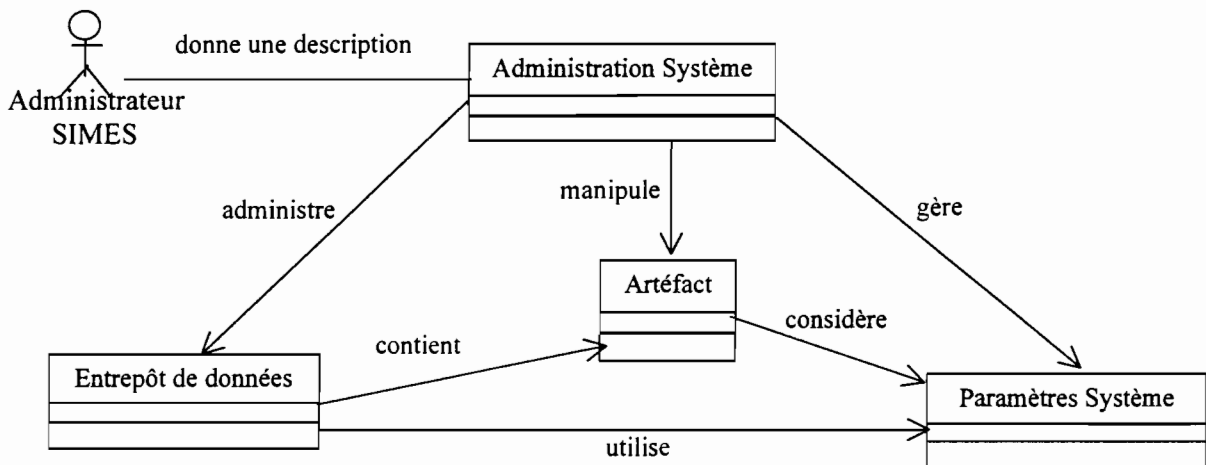


Figure 54 : Diagramme de classes pour le Nommage d'un artéfact.

✓ **Traitement d'une commande**

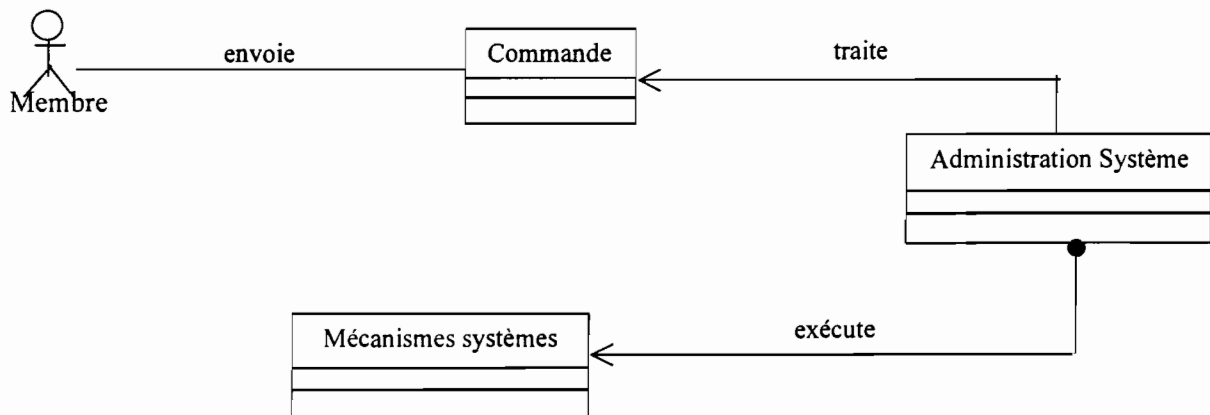


Figure 55 : Diagramme de classes pour le Traitement d'une commande.

✓ **Stockage d'un objet**

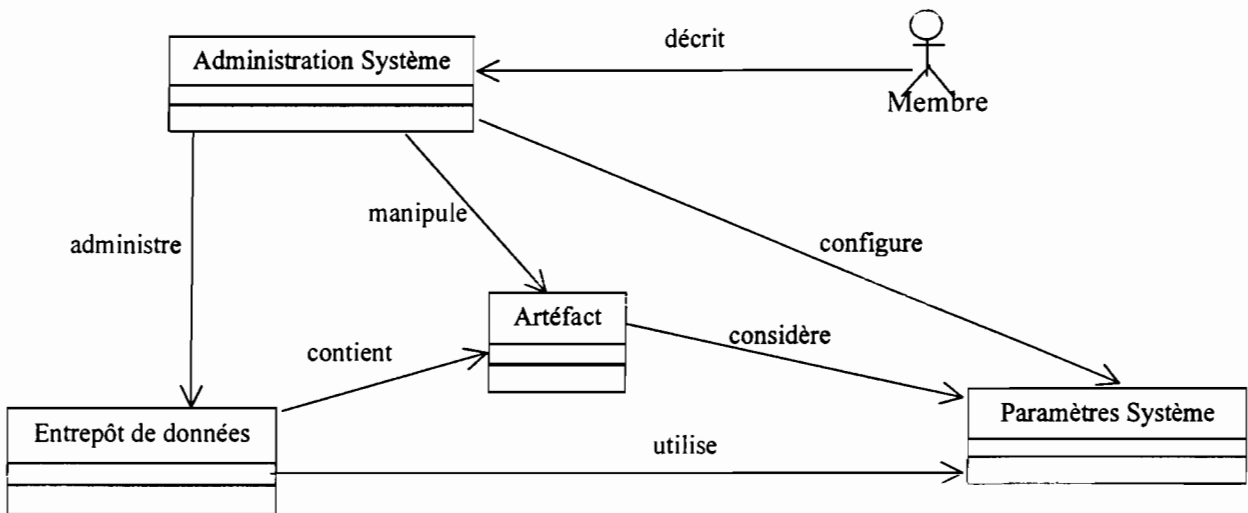


Figure 56 : Diagramme de classes pour le Stockage d'un objet.

✓ **Consultation d'un objet**

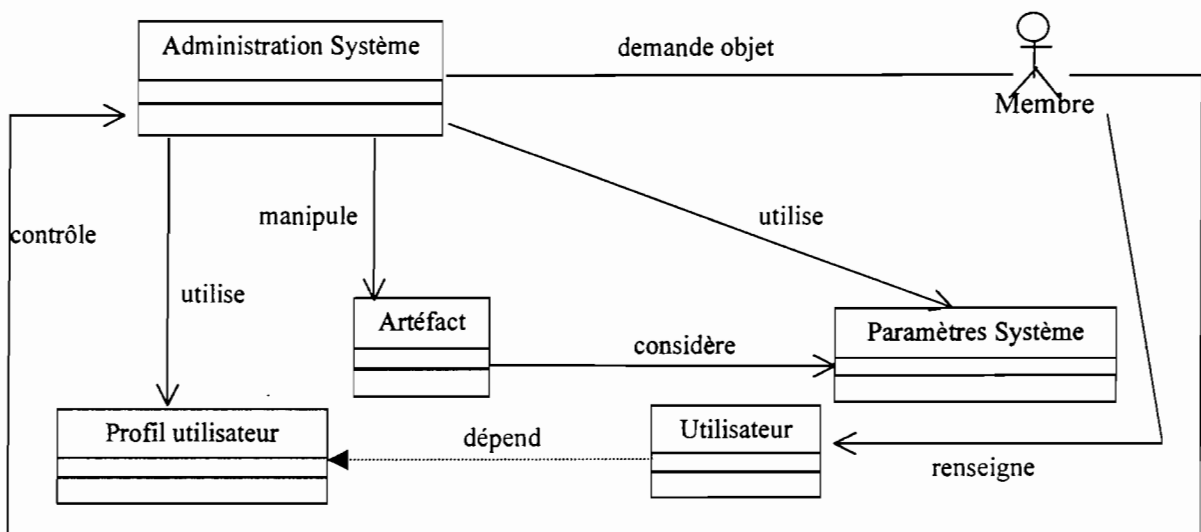


Figure 57 : Diagramme de classes pour la Consultation d'un objet.

✓ **Exécution d'un outil**

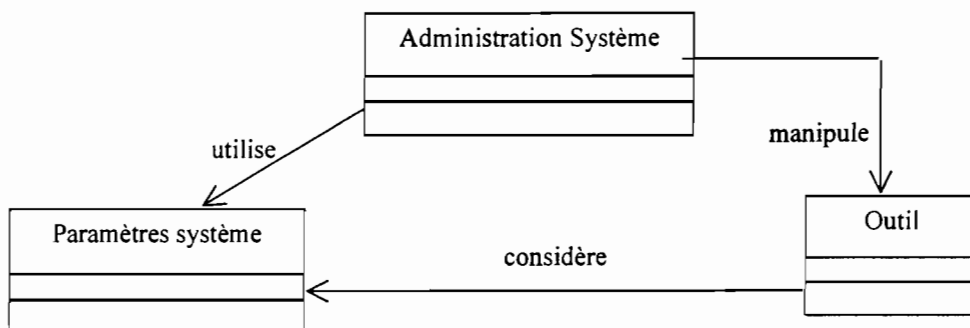


Figure 58 : Diagramme de classes pour l'Exécution d'un outil.

✓ Exécution d'une activité

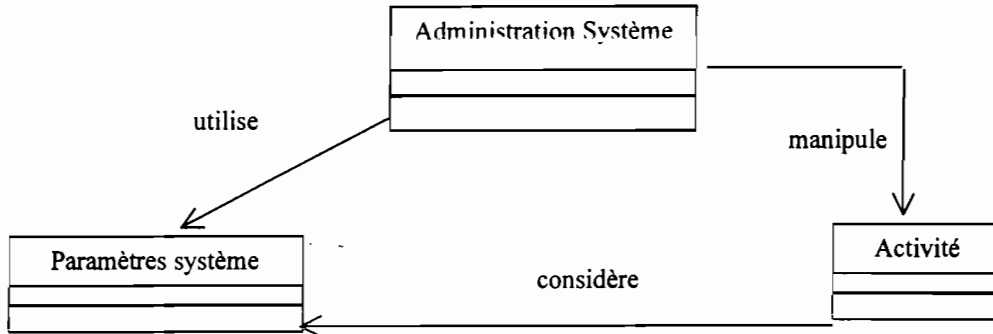


Figure 59 : Diagramme de classes pour l'Exécution d'une activité.

✓ Traitement d'une donnée

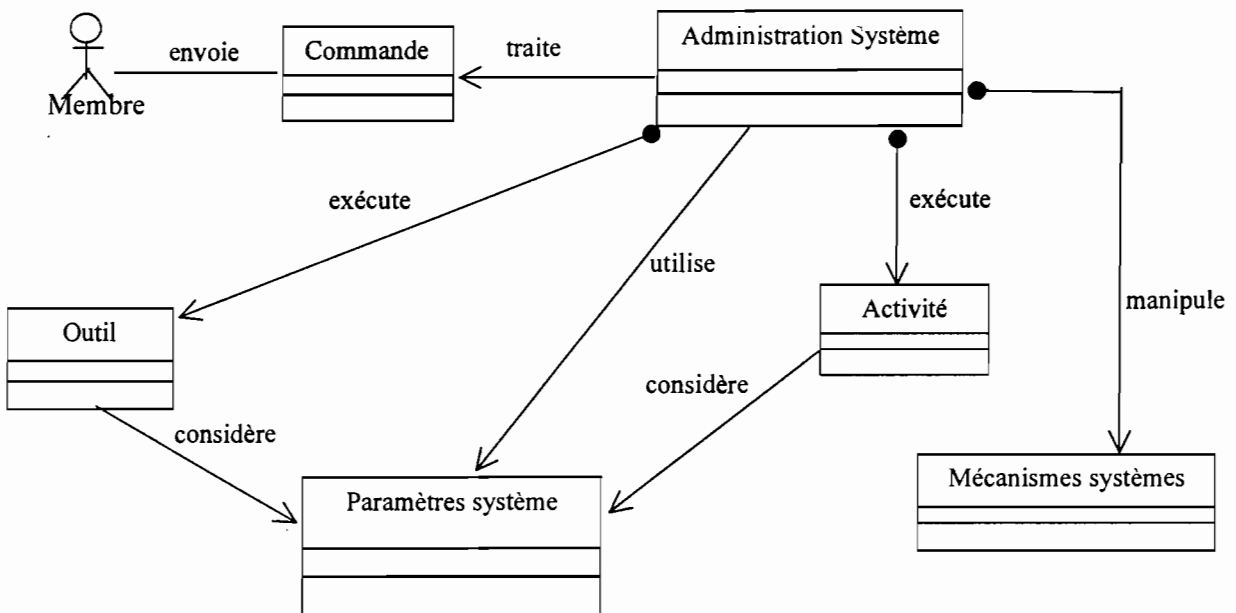


Figure 60 : Diagramme de classes pour le Traitement d'une donnée.

✓ Gestion d'une session

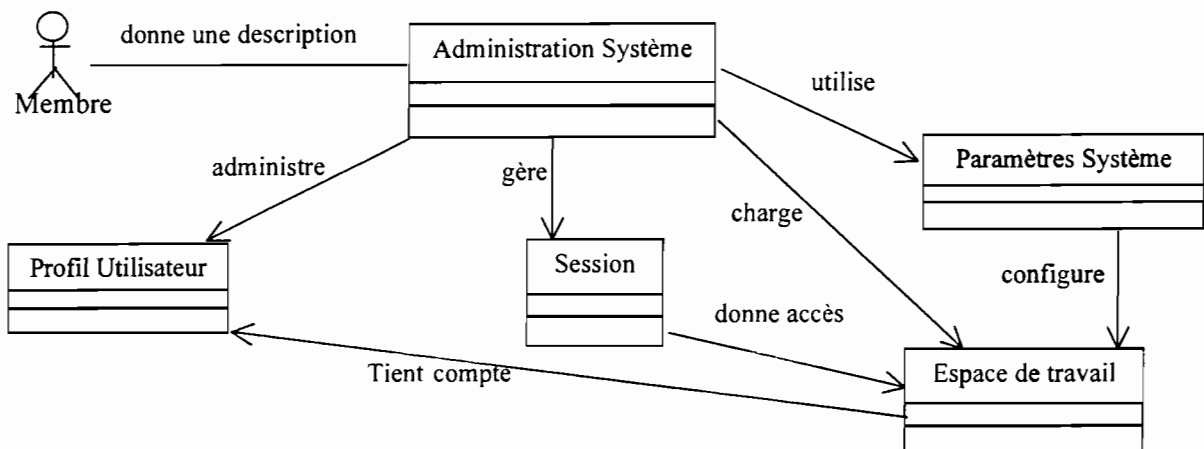


Figure 61 : Diagramme de classes pour la Gestion d'une session.

✓ **Administration du système d'information**

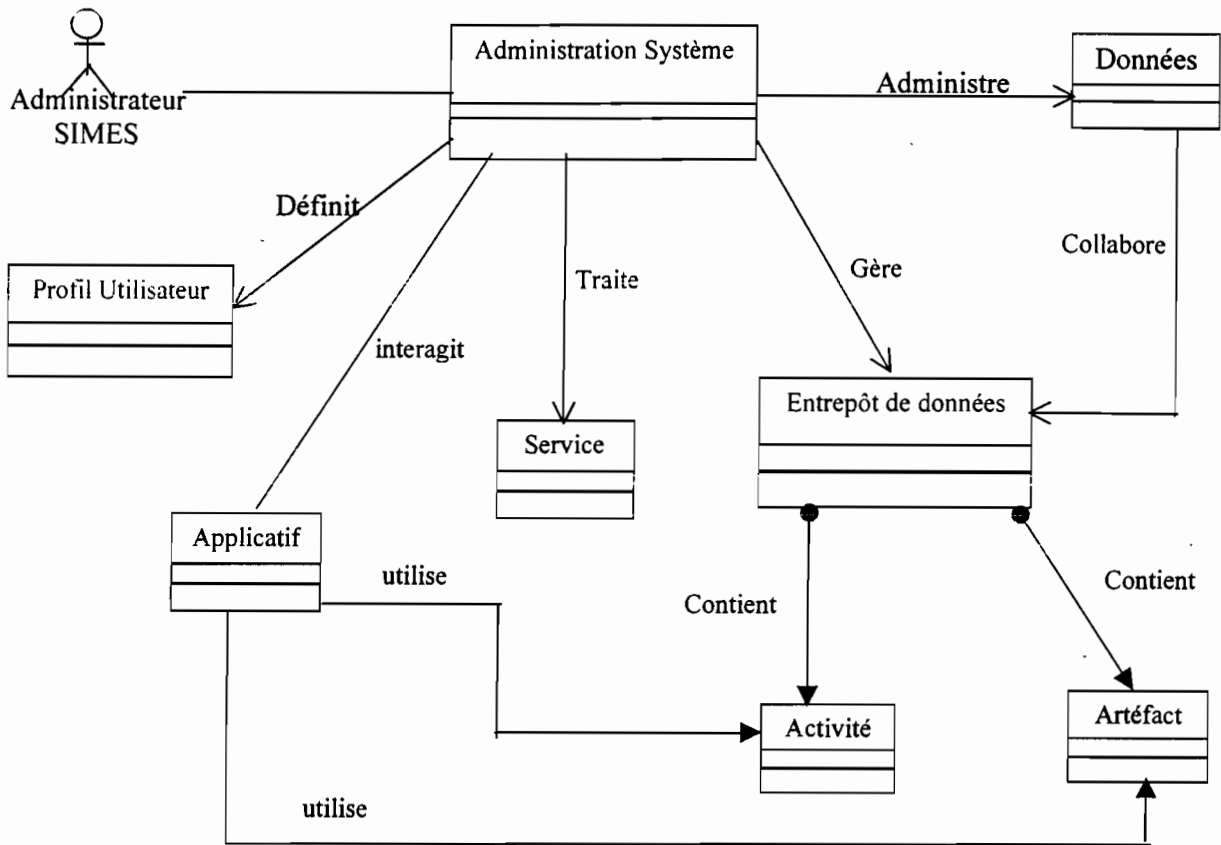


Figure 62 : Diagramme de classes pour l'Administration du système d'information.

✓ **Information d'un utilisateur**

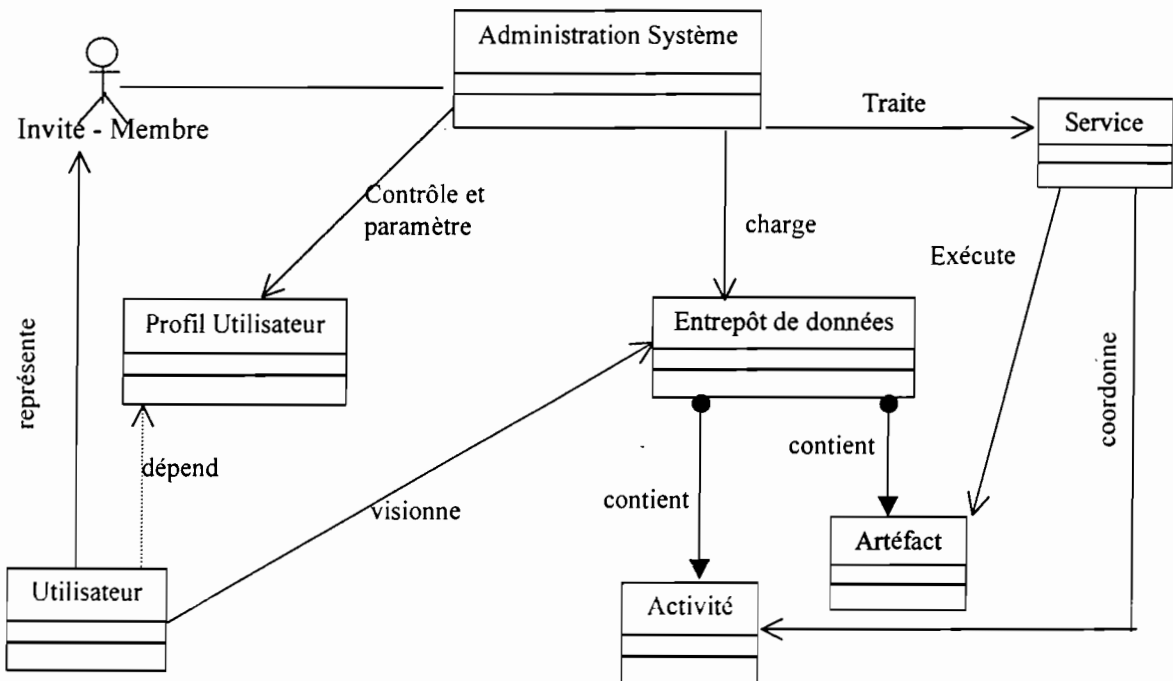


Figure 63 : Diagramme de classes pour l'Information d'un utilisateur.

✓ **Accès utilisateur**

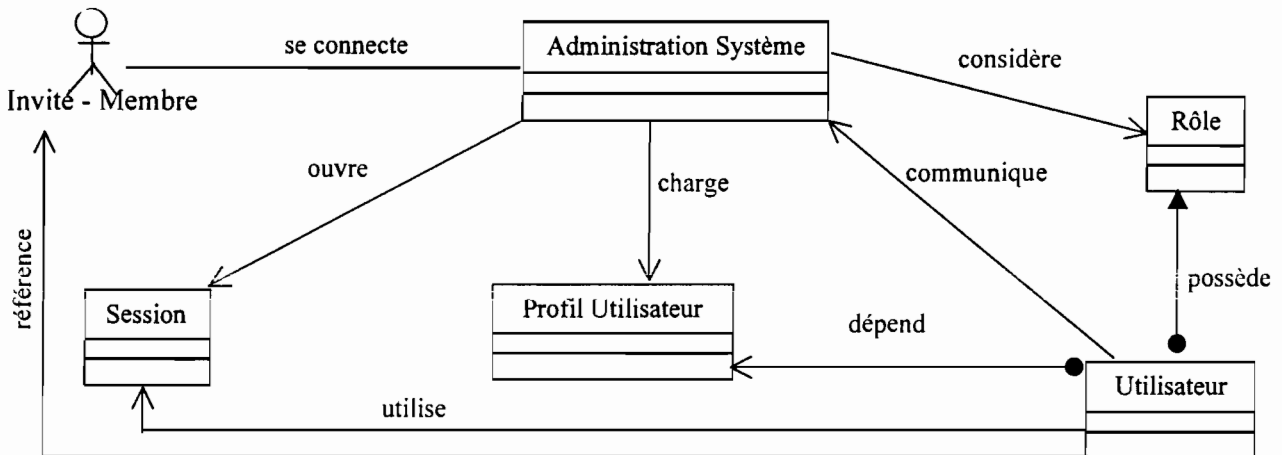


Figure 64 : Diagramme de classes pour l'Accès utilisateur.

✓ **Administration des droits des utilisateurs et groupes**

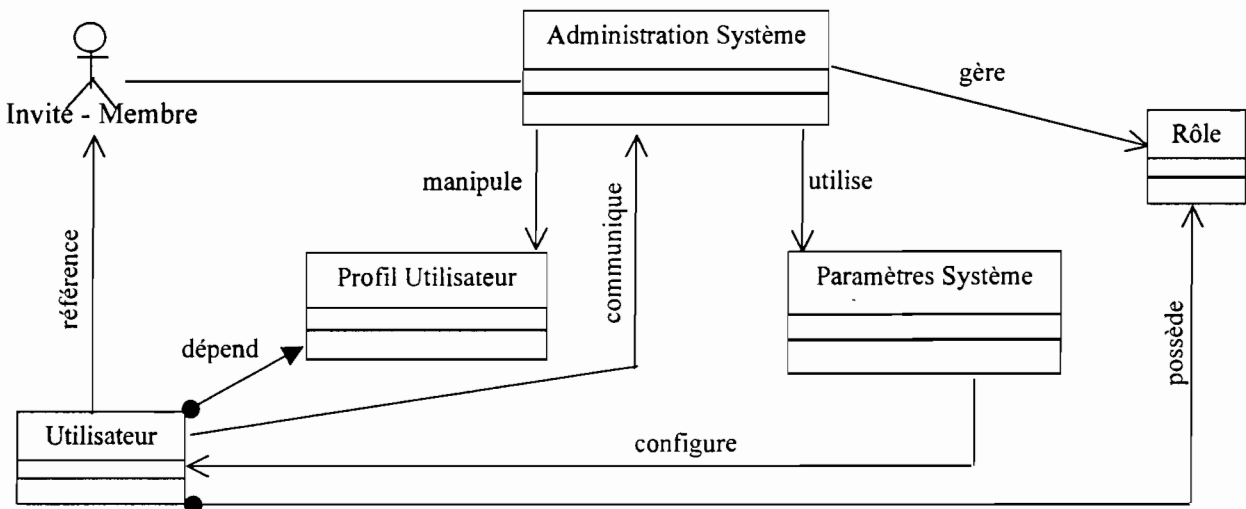


Figure 65 : Diagramme de classes pour l'Administration des droits des utilisateurs et groupes.

**II – 2 – 3 - 5 – Etude des structures organisationnelles**

La structuration organisationnelle d'un système consiste à regrouper les travailleurs et les entités en unités organisationnelles : c'est le paquetage.

Un paquetage est un groupement d'éléments de modélisation. Un paquetage peut contenir aussi bien des paquetages emboîtés que des éléments de modélisation ordinaires. Le système entier peut être pensé comme un unique paquetage de haut niveau comprenant l'ensemble. Tous les éléments de modélisation d'UML, y compris les diagrammes, peuvent être organisés en paquetages.

Une classe qui se trouve dans un paquetage appartient à celui-ci et ne peut être partagée par d'autres paquetages. En revanche, elle peut avoir des relations avec des classes qui appartiennent à

un autre paquetage. Elle peut aussi apparaître dans des diagrammes qui appartiennent à d'autres paquetages.

En ce qui concerne le système SIMES, les collaborations et les interactions entre les différentes classes et entités nous permettent de définir les paquetages suivants schématisés par la figure 66 :

- ⇒ Le paquet *Utilisateur* composé des classes :
  - Utilisateur (classe)
  - Profil utilisateur
  - Rôle
- ⇒ Le paquet *Espace de travail* composé des classes :
  - Bureau (sous paquet)
  - Service
- ⇒ Le paquet *Fenêtre* composé des classes :
  - Activité
  - Explorateur
- ⇒ Le paquet *Bureau* composé des classes :
  - Bureau (classe)
  - Fenêtre
- ⇒ Le paquet *Session* composé des classes :
  - Session
  - Utilisateur (paquet)
  - Espace de travail

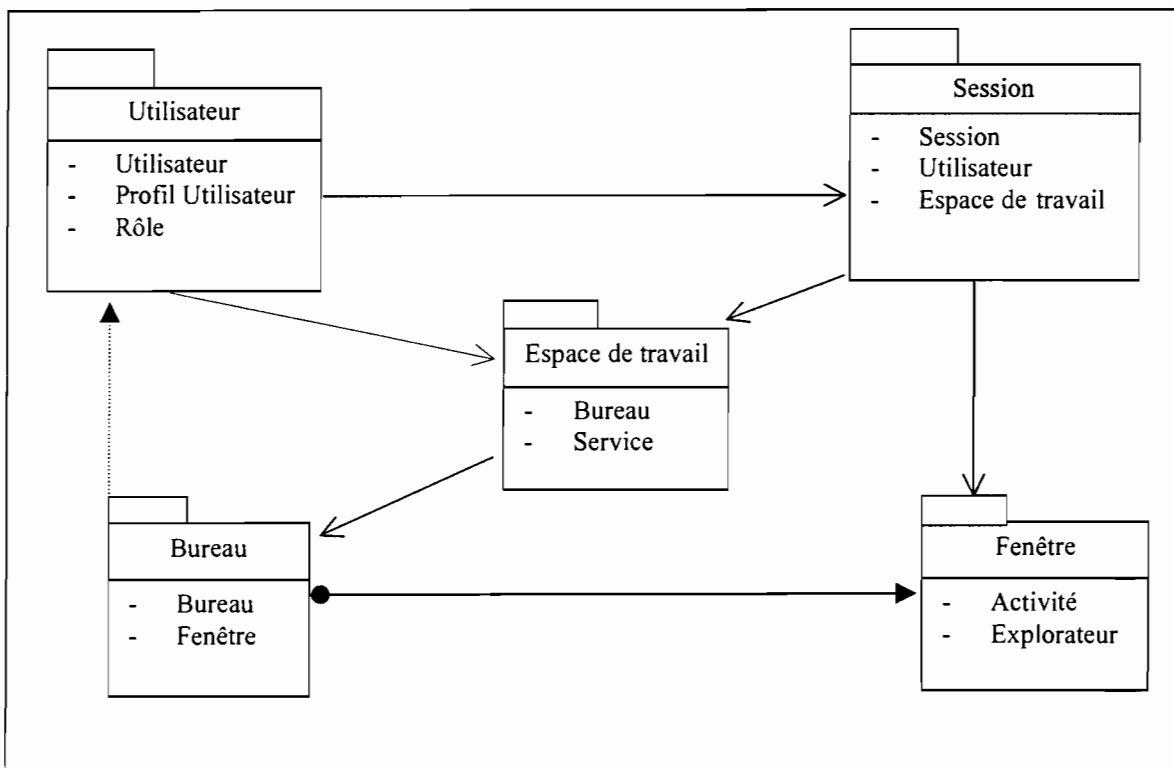


Figure 66 : Diagramme structurel statique du système SIMES.



## CONCLUSION PARTIELLE sur la modélisation

Cette partie de l'étude a permis de mettre en place le modèle logique appelé aussi en UML modèle logique du métier du système. Les différentes parties du système ont été décomposées en des unités de traitement qui pourront sans doute être vues comme des applications du système.

Les différents travailleurs et entités du système ont été identifiés de même que les activités menées par chacun d'eux. Ils ont été classifiés en catégories ou en groupes d'appartenance, ce qui a permis d'identifier les classes possibles du système.

Dans notre système informatique, un travailleur devient un utilisateur (au sens courant) du système informatique, ses activités deviennent des processus informatiques et les entités deviennent des objets entités du système informatique. Une illustration détaillée est présentée par la figure 67.

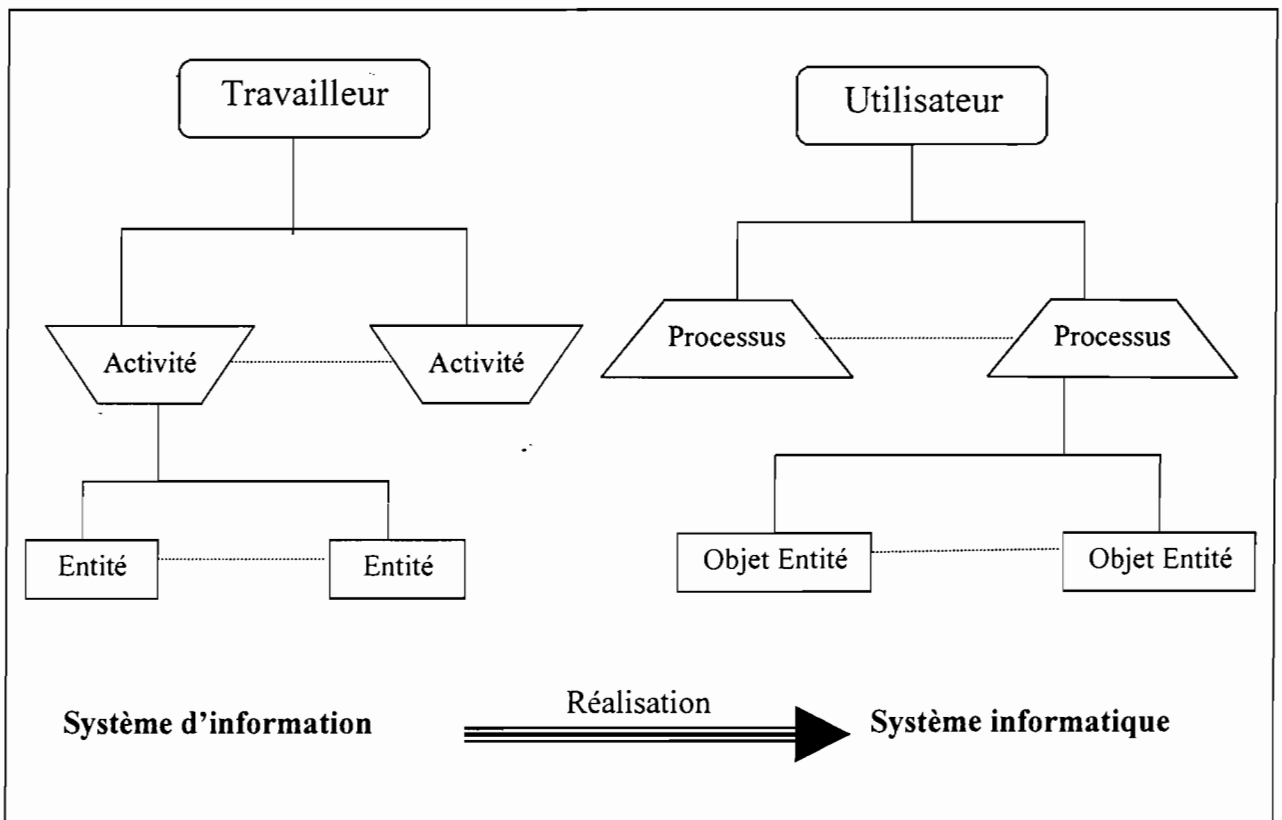


Figure 67 : Correspondance entre Système d'information et Système informatique en UML

# ***IMPLEMENTATION***

### III – IMPLEMENTATION DES OUTILS DU SYSTEME

#### III - 1 - Architecture fonctionnelle du SIMES

Le système est un ensemble de composants logiciels. Ils se situent à différents niveaux de la vision analytique du système SIMES et plus particulièrement dans la catégorie des composants outils (acquisition de l'information).

Le système SIMES fonctionnera sur des sites distants. La vision globale architecturale du système peut être perçue à travers l'architecture fonctionnelle de la figure 68 extraite du *Compte rendu des travaux des journées du 22 au 25 Mai 2000 du SIMES [Ali B. KABA 2000]* suivante :

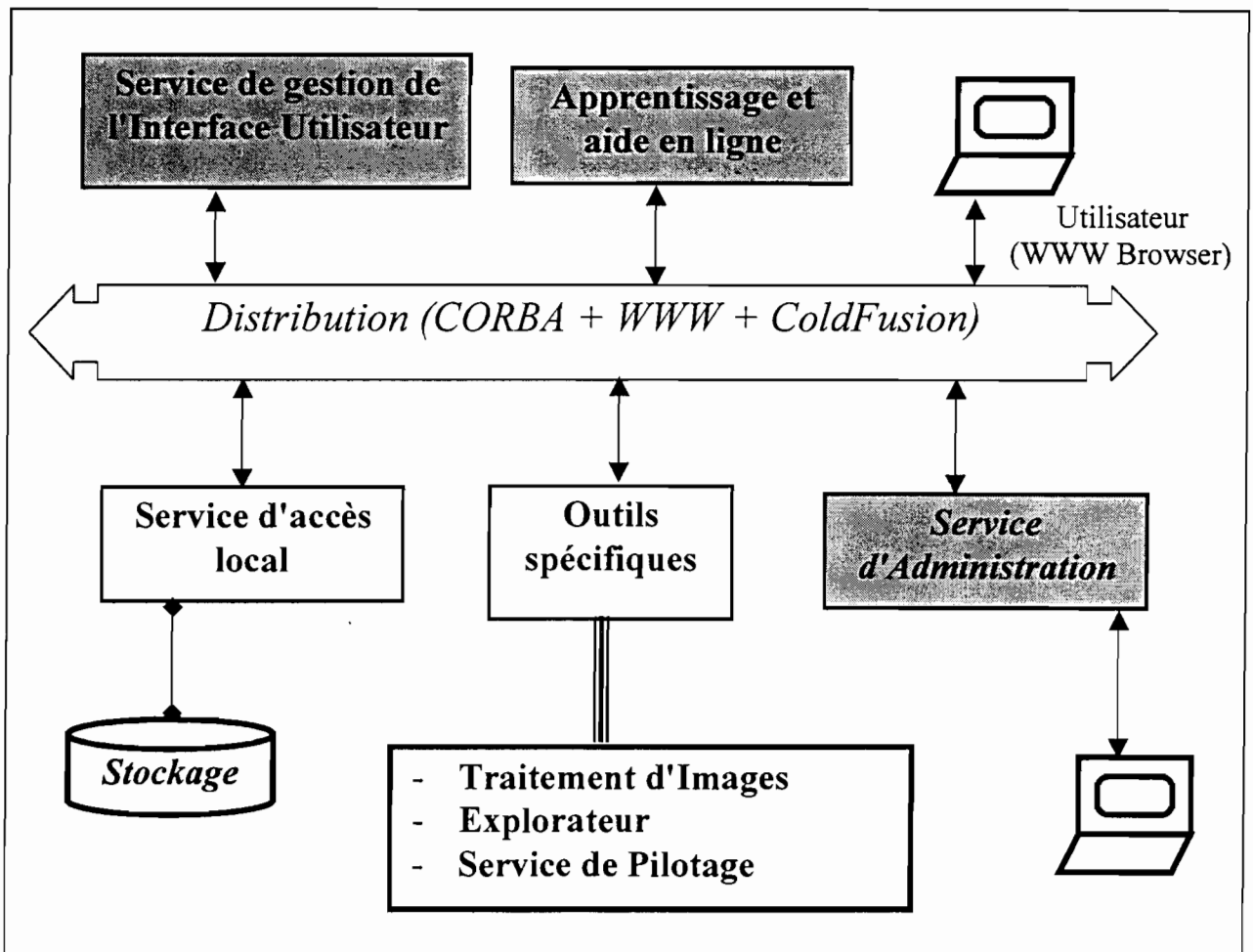


Figure 68 : Architecture fonctionnelle de SIMES

### III - 2 - Implémentation fonctionnelle

Les concepts émergents des facettes d'intégration dans SIMES sont d'une part l'interface qui doit être présente partout, commode et permettre l'accès à des objets distribués, et d'autre part les technologies WWW, CORBA, ColdFusion, C++ et Java.

CORBA (Common Object Request Broker Architecture) est intéressant pour accéder à des objets distants, mais lourd pour des objets locaux. L'administration est lourde à mettre en œuvre parce qu'il faut développer des coquilles et les déployer chez les clients. L'avantage est qu'il permet l'intégration des systèmes d'informations distribués. Le couple CORBA + WEB est une solution possible pour une interface conviviale et générale.

ColdFusion, C++ et Java permettront la réalisation (programmation de code) des applications du système (agents de pilotage et les outils d'intégration).

#### III - 2 - 1 - Architecture des couches : Vue fonctionnelle

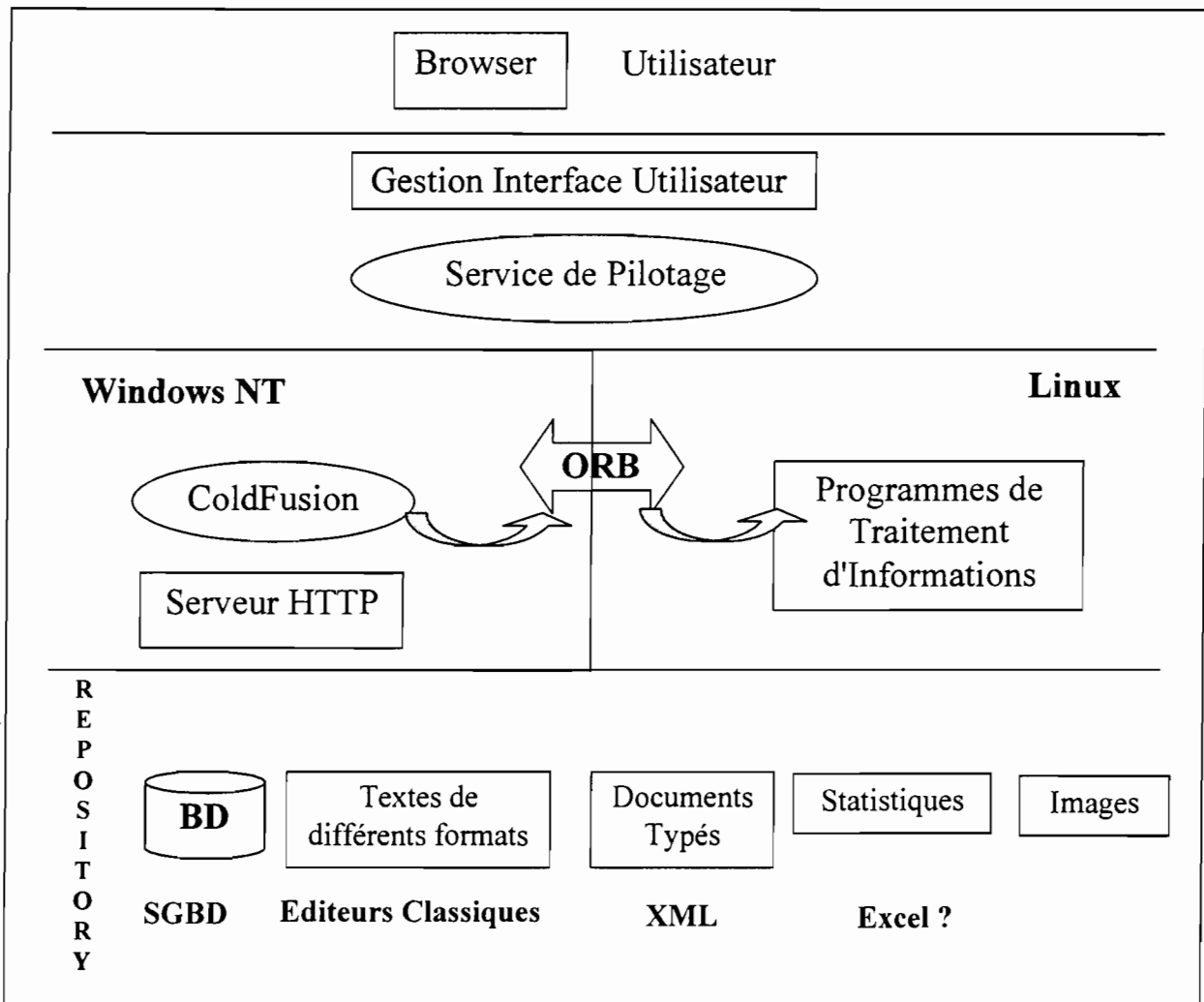


Figure 69 : Vue hiérarchique des facettes d'intégration dans SIMES

On s'appuie sur le Web pour la réalisation de l'interface utilisateur. L'échange entre deux (2) outils portera sur les données. A ce niveau se posent des problèmes de conversion de formats de données et la réalisation des convertisseurs nécessaires. d'où la nécessité d'identification des différents formats utilisés. La synchronisation et la gestion des exceptions seront au niveau du contrôle. Quand à la plate forme, il faudra parler de serveur de données (Windows NT) et de serveur d'outils (Linux). L'illustration graphique est faite par la figure 69.

Les services de base du système sont l'interrogation (Base de Données, Documents typés du genre XML, image), la transformation (appuyée sur les outils de traitement d'images), l'alimentation et la mise à jour, l'organisation (service de pilotage).

### III - 2 - 2 - ColdFusion : Présentation et usage dans le système

ColdFusion est un système de développement rapide d'applications destiné aux développeurs professionnels désireux de créer des applications dynamiques et des sites Web interactifs. ColdFusion offre un moyen plus rapide d'intégrer des technologies de navigateur, de serveur et de base de données à de puissantes applications Web.

Pour développer une application avec ColdFusion, le codage dans un langage de programmation traditionnel n'est pas nécessaire. On créera les applications en combinant du HTML standard avec un langage de balisage côté serveur de ColdFusion, le langage CFML (ColdFusion Markup Language).

L'environnement ColdFusion comporte ColdFusion Studio et ColdFusion Serveur.

#### III - 2 - 2 - 1 - Configuration requise pour l'installation

ColdFusion Studio est un environnement qui permet de développer les applications ColdFusion. Il existe uniquement sous l'environnement Windows NT et Windows 95 - 98. ColdFusion Serveur se greffe à un serveur Web existant comme Netscape. Il est disponible sur NT.

ColdFusion est donc pris en charge par l'environnement Microsoft Windows.

##### ◆ *Windows 95/98 et Windows NT*

Le système doit satisfaire aux exigences suivantes pour permettre l'installation et l'exploitation de ColdFusion :

- **Système d'exploitation** : Windows NT 3.51 (ou version supérieure) ou Windows 95-98
- **Processeur** : Intel 486 ou supérieur (processeur Pentium recommandé)
- **Espace disque** : 30 Mo
- **Mémoire** : 32 Mo
- **Serveur Web** : Un serveur World Wide Web (prenant en charge les normes NSAPI, ISAPI ou WSAPI recommandé)
- **Périphériques** : Un lecteur de CD-ROM

### III - 2 - 2 - 2 - Quelques caractéristiques de ColdFusion

Les fonctionnalités de ColdFusion se regroupent en quatre (4) domaines principaux : un développement rapide, un déploiement évolutif, une intégration ouverte et une sécurité totale.

Les fonctionnalités utilisées pour l'implémentation du système sont beaucoup plus celles du développement rapide et de l'intégration ouverte. Ce sont :

◆ **Développement rapide**

On a intégré ColdFusion Studio avec le serveur ColdFusion, en développant des outils visuels. Le langage de rédaction de scripts de serveur est à base de marqueurs CFML. On a utilisé entre autres les fonctionnalités de la programmation visuelle bidirectionnelle, de l'assurance qualité des pages dynamiques, de la Visualisation de site et de Script CFML de ColdFusion 4.0.

◆ **Intégration ouverte**

On a intégré ColdFusion avec les systèmes de serveur (messageries, serveurs Web et répertoires). La prise en charge du langage XML (Extensible Markup Language) et des technologies d'entreprise par ColdFusion a été exploitée.

ColdFusion est un environnement de développement basé sur HomeSite, un éditeur HTML. HomeSite inclut un outil de visualisation de bases de données, un debugger et bien d'autres fonctionnalités.

## III - 3 - Applications Web avec ColdFusion

Le système SIMES est une application utilisée sur le réseau mondial à travers le Web. Le site SIMES est un site interactif. Le système peut être vu comme un site de pilotage qui ne serait composé que d'outils intégrables. Ces outils peuvent être considérés comme des applications Web. La plupart de ces outils seront développés à partir de ColdFusion.

### III - 3 - 1 - Description d'application

Une application Web avec ColdFusion est formée :

- ⇒ d'un ensemble de pages HTML caractérisées par une extension spécifique : CFM.
- ⇒ de pages contenant des éléments de programmation supplémentaires : le CFML.

La figure 70 illustre la description graphique d'une application Web avec ColdFusion. Les pages constituant une application communiquent entre elles par échanges d'informations.

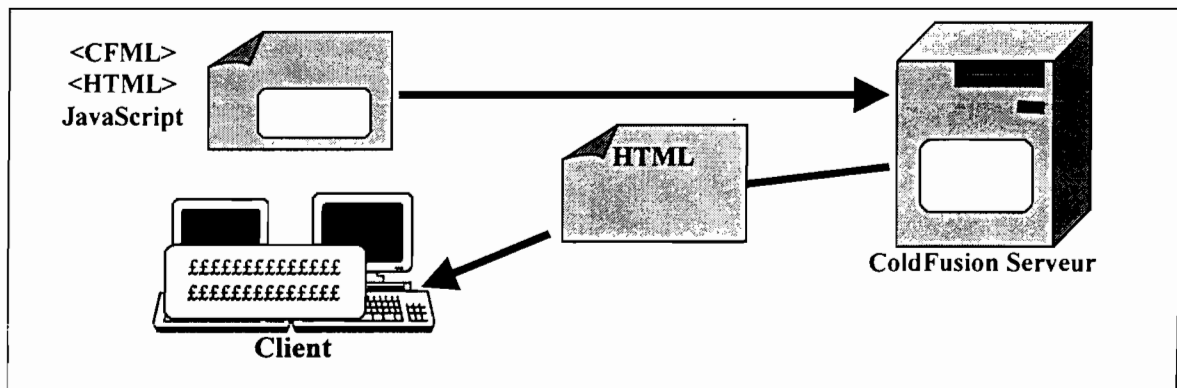


Figure 70 : Schéma descriptif d'une application Web avec ColdFusion

### III - 3 - 2 - Schémas fonctionnels

Le fonctionnement de toute activité à mener sur le site de pilotage du projet SIMES, peut se résumer à deux (2) niveaux communiquant entre-eux. Le premier niveau est celui du côté Serveur et le deuxième du côté du client c'est-à-dire sur la machine de l'utilisateur (figure 71 et 72). Pour l'étude pratique menée, Windows NT a été utilisé comme système d'exploitation pour le serveur et l'explorer comme navigateur. Quant à la machine de l'utilisateur, elle a comme système d'exploitation le Windows 95.

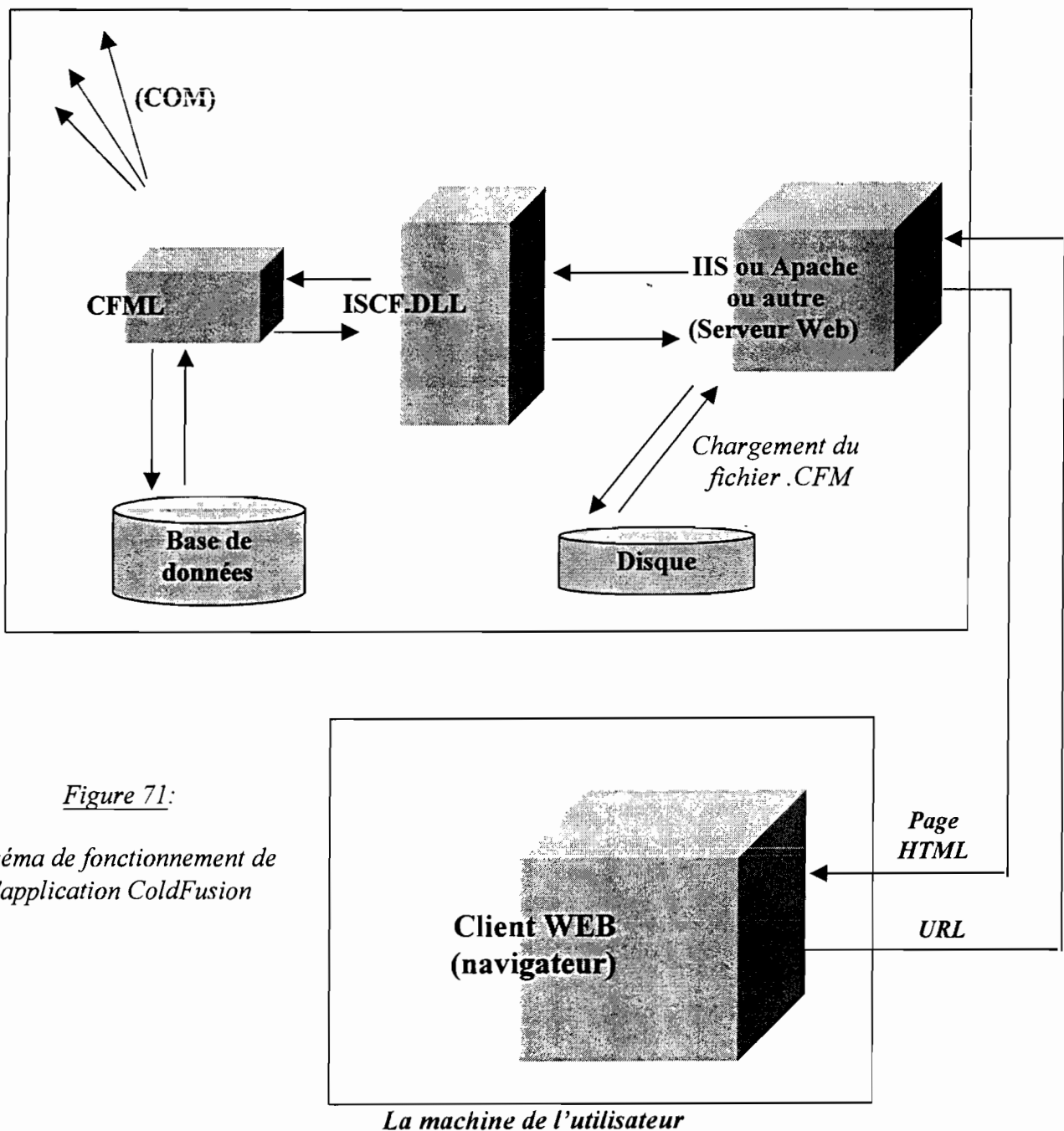


Figure 71:

Schéma de fonctionnement de l'application ColdFusion

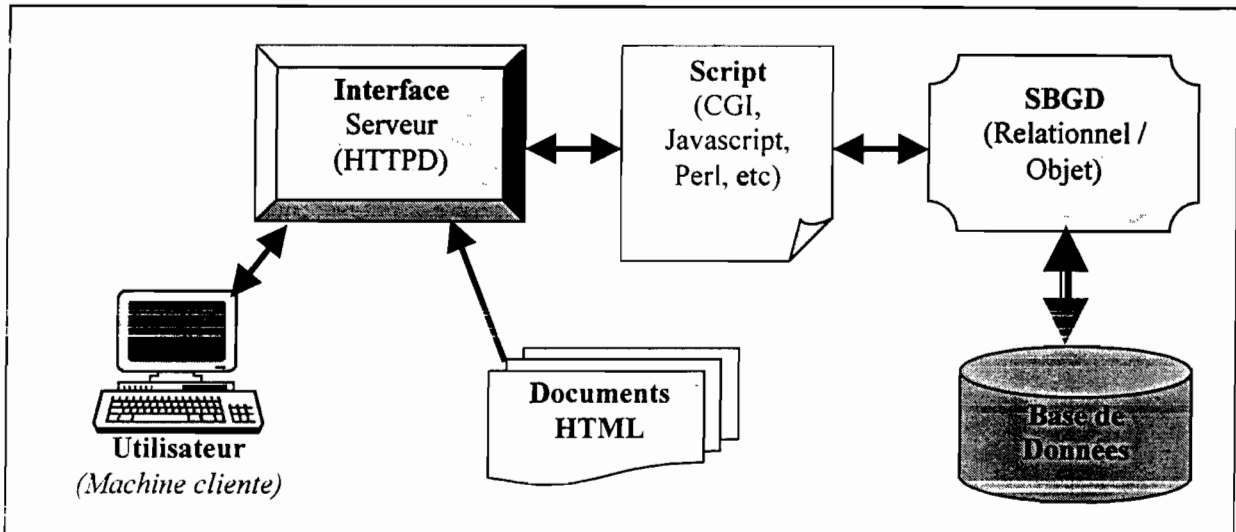


Figure 72: Schéma fonctionnel d'une page HTML avec des scripts.

### III - 3 - 3 - Informations partagées par les applications du système

Les applications du système seront formées par un ensemble de pages qui partagent des informations communes se regroupant en quatre (04) catégories à savoir les paramètres, les variables, l'authentification et les messages.

Les paramètres permettent la transmission des données aux fonctions. Ils sont également appelés arguments. Bien que les fonctions n'acceptent pas toutes des paramètres, la plupart des fonctions acceptent au moins un argument. Les paramètres sont constitués par l'ensemble des expressions valides de ColdFusion. On peut citer entre autres les nombres, les chaînes de caractères, les valeurs booléennes (logiques), les listes, les tableaux, les requêtes et même des fonctions.

Une variable est tout simplement un paramètre auquel une valeur est attribuée. Les variables sont utilisées dans des applications ColdFusion de plusieurs façons, par exemple dans les formulaires ColdFusion afin qu'elles contiennent des données soumises à une application. Les variables sont essentielles pour manipuler les entrées de formulaire, pour faire passer des données de chacun des champs d'un formulaire vers la page ACTION de l'application. Ces variables sont également appelées paramètres dynamiques. Les types de variables utilisés par les applications sont des chaînes de caractères, des tableaux, des structures, des tableaux à clés et des variables base de données. Elles sont repérées à l'aide du dièse (#) dans les applications. Pour leur affichage on utilise deux (2) dièses (##).

L'authentification concerne en générale les utilisateurs du système. Il consiste en la déclaration des utilisateurs du système et à leur reconnaissance lors de la connexion au système. L'authentification ColdFusion intègre celle de Windows NT.

Les messages peuvent être des résultats de requêtes ou des messages d'erreurs. Ils peuvent être créés, modifiés, affichés, supprimés ou des messages existant dans le système de base.



### III - 3 - 4 - Enchaînement des pages

Les pages constituant les applications sont en même temps les écrans des applications. Leur enchaînement n'est donc qu'une question de lien hypertexte puisqu'il s'agit aussi de pages HTML. Le lien dans une page se fait à l'aide de la balise `<A HREF>`. Sa syntaxe est la suivante :

```
<A HREF = "chemin d'accès + nom de la page">
```

Le chemin d'accès est considéré comme l'Url de la machine hébergeant la page. Au niveau du formulaire ColdFusion, le passage de variable se fait à l'aide de la balise suivante

```
<CFFORM ACTION> — </CFFORM>.
```

```
<CFFORM ACTION = "exemple.cfm" METHOD = "Post">
  <INPUT TYPE = "Hidden" NAME = "nom_type" VALUE = "24">
  <INPUT TYPE = "Submit" VALUE = "Entrer">
</CFFORM>
```

La plupart des pages de l'application contiennent des formulaires qui rendent aisée la manipulation des données (des bases de données). ColdFusion met un accent particulier sur l'utilisation des formulaires.

### III - 3 - 5 - Les pages entraînées par des bases de données

Des outils du système utilisent des données provenant de bases de données. L'alimentation de ces outils se fait par la manipulation de bases de données.

L'interrogation d'une table se fait de la façon suivante :

```
<!-- Début de l'interrogation de la base de données --->
<CFQUERY NAME= "Nom" DATASOURCE= "NomBD">
SELECT * FROM la table
</CFQUERY>
<!-- Fin de l'interrogation --->
```

L'affichage du résultat de la requête sous le format simple sans cadre est structuré comme suit :

```
<!-- Début du bloc de sortie --->
<CFOUTPUT QUERY="Nom">
(#Attribut1#)</B> #Attribut2# </B> #Attributn# </B> <BR>
</CFOUTPUT>
<!-- Fin du bloc de sortie --->
```

### III - 4 - Réalisation de quelques éléments du système

Quelques éléments du système ont été réalisés par prototypage. Il s'agit de :

- la structuration des fonctionnalités du système par rubrique ;
- la connexion d'utilisateur et l'ouverture / fin de session ;
- les nouveautés et les événements du projet ;
- les textes et publications ;
- les liens utiles du projet ;
- le gestionnaire des documents ;
- la mise en ligne d'un document.

### III - 4 - 1 - La structuration des fonctionnalités par rubrique

L'ensemble des fonctionnalités du système se répartit en six (6) rubriques facilitant l'exploitation du site. Ces rubriques sont : la rubrique de synthèse, la référence bibliographique, les forums, la messagerie, les documents partagés et l'interface d'administration.

Nom de la rubrique	Services offerts
Rubrique de synthèse	<ul style="list-style-type: none"> <li>- Nouveautés</li> <li>- Evènements</li> <li>- Contacts</li> <li>- Partenaires</li> <li>- Plan du site</li> </ul>
Référence bibliographique	<ul style="list-style-type: none"> <li>- Textes et publications</li> <li>- Bibliographie</li> <li>- Liens utiles</li> </ul>
Forum	<ul style="list-style-type: none"> <li>- Diffusion de forum</li> <li>- Ajout des liens</li> </ul>
Messagerie	<ul style="list-style-type: none"> <li>- La liste de diffusion</li> <li>- Client mail</li> </ul>
Documents partagés	<ul style="list-style-type: none"> <li>- Réunions plénières</li> <li>- Groupes de travail</li> <li>- Administration du projet</li> </ul>
Interface Administration	<ul style="list-style-type: none"> <li>- Gestionnaire de documents</li> <li>- Mise en ligne de document</li> <li>- Modification de l'organisation</li> </ul>

Ces rubriques sont présentées sous la forme du menu principal d'une page d'accueil. Du HTML pur a été employé pour la réalisation de cette page. Des liens ont été introduits afin de la rendre simple et dynamique.

### III - 4 - 2 - Connexion d'un utilisateur et ouverture / fin d'une session

Avant toute opération sur le site, il faut d'abord se connecter. Si le système authentifie l'utilisateur, il l'accepte et lui fournit les informations sur sa dernière session. Il vérifie tous ses droits à partir de son profil.

La classe utilisateur de notre système informatique est la première à être utilisée. Le tag CFAUTHENTICATE permet d'authentifier l'utilisateur et facilite le contrôle des droits à partir de la configuration de la classe Profil utilisateur. Dès cet instant, le système fournit la liste des nouveautés (confère III – 4 – 3).

Après ces opérations, la classe Session est mise à jour par activation de l'opération « Démarrer ». Les propriétés de la classe session sont mises à jour par CFAUTHENTICATE qui utilise les propriétés des classes utilisateur et Profil utilisateur. CFSTOREDPROC spécifie les autres informations de connexion à la base de données et identifie les procédures stockées à exécuter pour la mise à jour des autres classes. Le schéma d'une connexion au système est présenté par la figure 73.

La session ainsi ouverte, active l'opération « Afficher » de la classe Bureau qui, en fonction des paramètres d'environnement et des données du profil, alimente les propriétés « Liste de fenêtres » et « Liste des objets ».

Ces différentes actions s'effectuent dans le système informatique à l'aide de CFHTTP (pour effectuer les opérations GET et POST) afin d'envoyer des variables, des requêtes, des cookies ou CGI directement au niveau d'un serveur. CFHTTPPARAM spécifie les paramètres nécessaires aux différentes opérations.

La configuration de l'espace de travail est effectuée après l'activation de l'opération « Définir ». Cette configuration est terminée après que les paramètres d'environnement soient mis à jour et que les propriétés des fenêtres et objets soient décrites.

Ainsi, l'utilisateur a la possibilité d'initier toute sorte d'opération qui sera contrôlée en fonction de son rôle sur les objets. Ces travaux seront possibles si l'opération « lancer » de la classe service est activée et que ces différentes propriétés sont renseignées à l'aide des tags CFAPPLICATION, CFCOOKIE, CFOBJET, CFPARAM, CFPROCPARAM, CFPROCRESULT et quelques tags d'instruction sur les scripts.

En activant l'opération « Déconnecter » de la classe Session à l'aide de CFSTOREDPROC permettant de lancer la procédure de déconnexion, l'utilisateur met fin à sa session. Cette fin de session est décrite à l'aide de la figure 74.

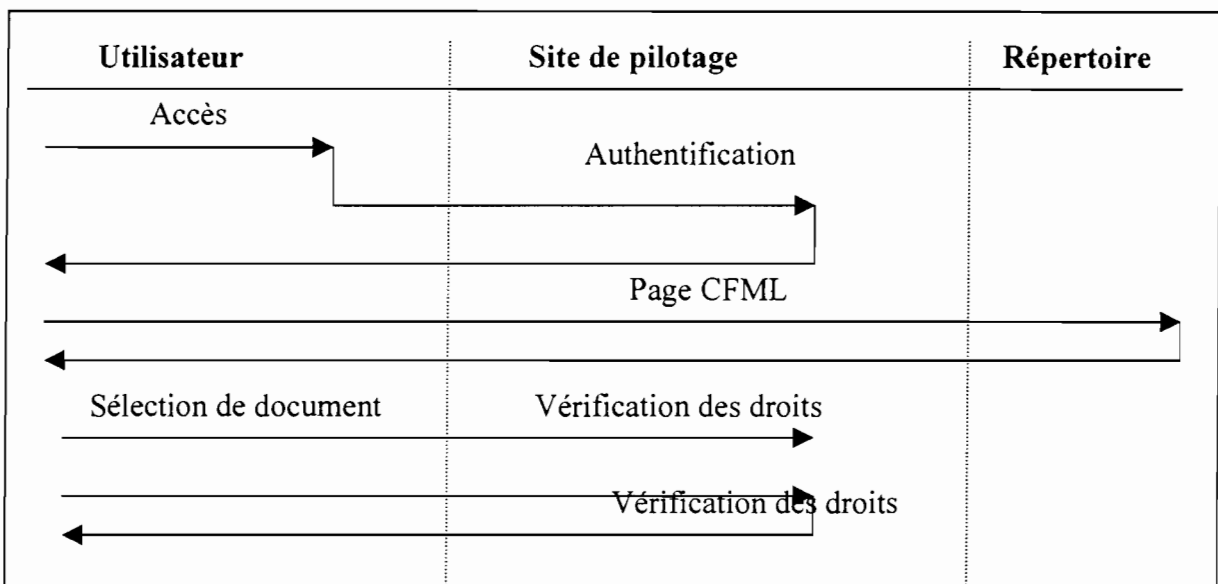


Figure 73 : Schéma d'une connexion

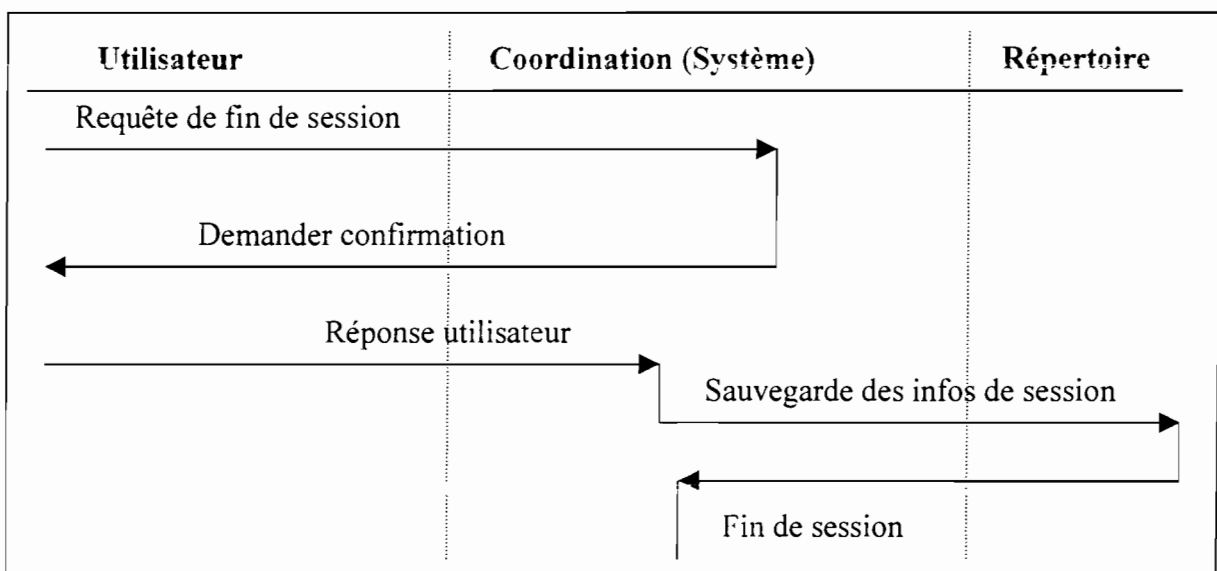


Figure 74 : Schéma d'une fin de session

Gérer l'entrepôt de données ou gérer les outils du système se résume à l'invocation d'un outil du système informatique. Cette invocation peut avoir comme résultat plusieurs services tels que l'identification, le stockage, la consultation, le traitement d'une commande, l'exécution d'une activité, etc. Le schéma de la figure 75 résume au mieux les grandes lignes de l'invocation d'un outil du système SIMES.

Un outil du système réalise un ensemble d'activités et d'artéfacts. Au début de l'invocation de l'outil, ses activités sont à l'état « décrit » par l'exécution de l'opération « Décrire » de la classe Activité. Les changements d'états successifs des activités constituent les étapes d'exécution de l'outil. Ces changements d'états s'effectuent grâce aux différentes opérations de la classe.

Les propriétés des classes sont mises à jour au fur et à mesure que les activités ou les artéfacts arrivent en fin d'utilisation.

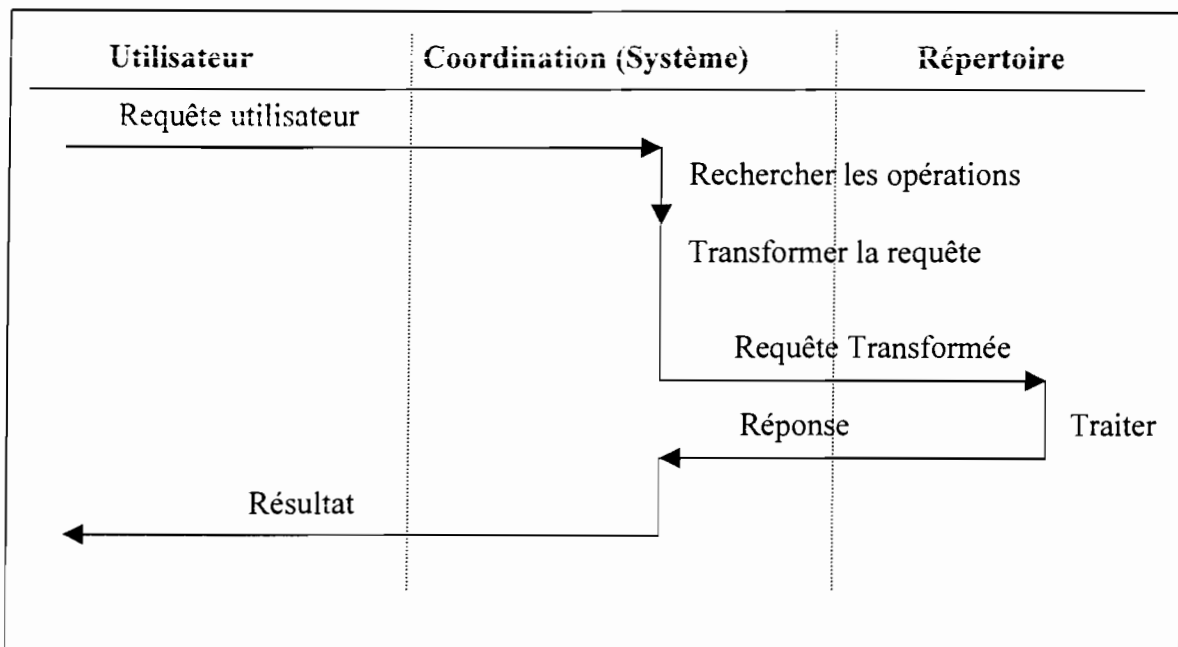


Figure 75 : Schéma d'invocation d'un outil

### III - 4 - 3 - Les nouveautés et les évènements du projet

Ces services consistent à offrir à l'utilisateur l'ensemble des évènements ou des nouveautés concernant le projet. L'utilisateur a la possibilité de lister tous les évènements ou seulement les nouveautés, de charger un ou dix (10) éléments ou d'ajouter un nouvel élément. Il s'agit de conférences, de définition de liens, de mise en ligne de documents, de rencontres, etc.

On effectue une recherche sur l'ensemble du site en retenant les évènements de date récente à l'aide du moteur de recherche VERITY de ColdFusion. A cet niveau trois (3) tags sont utilisés :

- CFCOLLECTION qui permet de définir l'emplacement des indexes et des éléments de travail ;
- CFINDEX qui permet de lancer une indexation ;
- CFSEARCH qui lance une recherche sur des indexs précédemment créés.

L'ajout se fait par utilisation du tag CFINSERT.

La réalisation de cet élément du système a conduit à la mise en œuvre d'un certain nombre d'écrans parmi lesquels un présente toutes les rubriques du système. La figure 76 montre cet interface en mode exécution. Il s'agit de la fonction « Nouveauté » de la rubrique de synthèse.

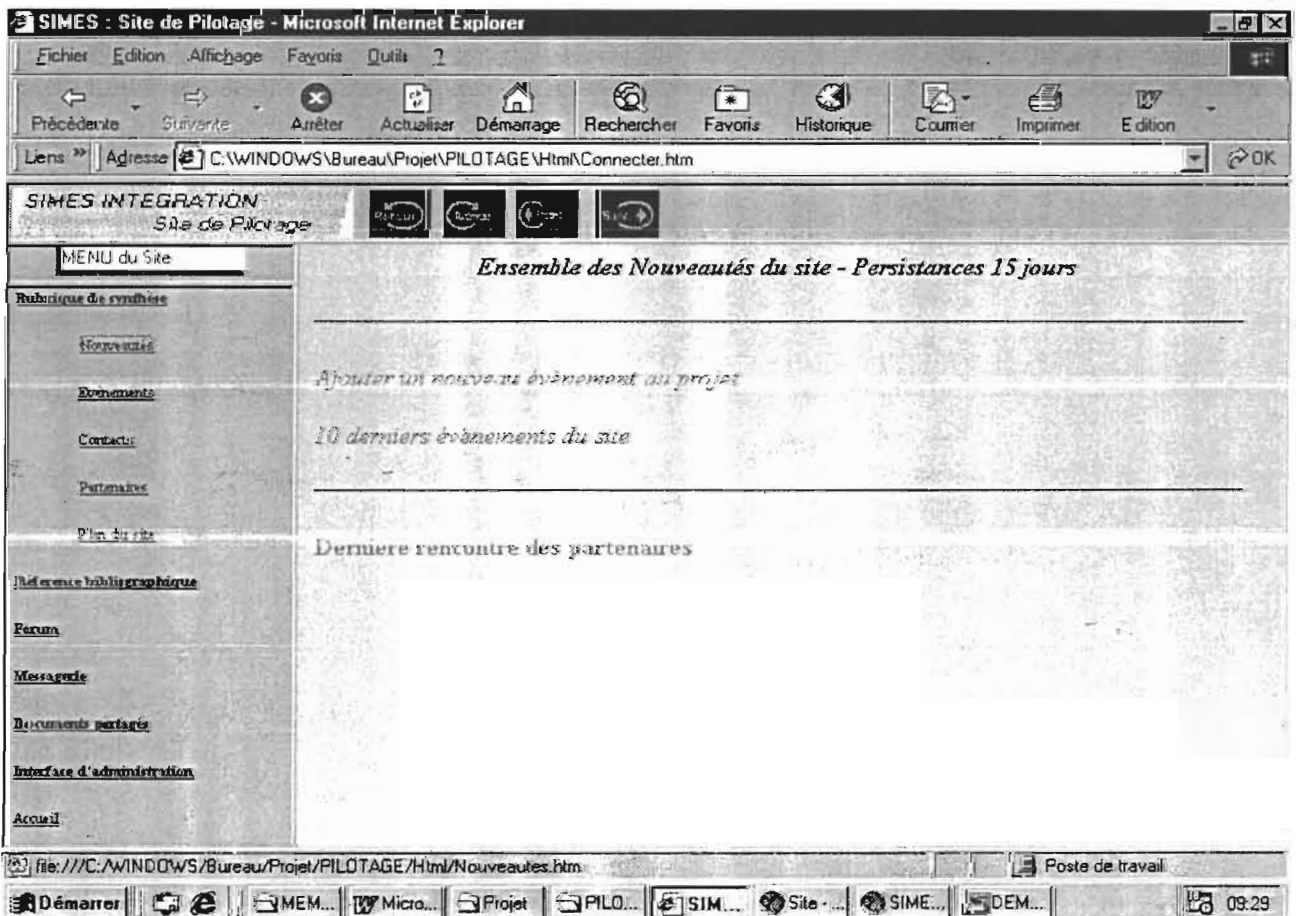


Figure 76 : Ecran Rubrique de synthèse - Nouveautés

### III - 4 - 4 - Les textes et publications

Cette fonctionnalité permet de retrouver sur le site un texte de travail ou une publication. La recherche de ces éléments se fait en fonction de leur titre qui est considéré comme un paramètre au niveau de la page (le formulaire de saisie). Ce paramètre est transmis à la base par référence à l'aide de champs de formulaire (form.variablename) ou de Cookies HTTP (Cookie.variablename).

Le lieu d'affichage doit être précisé et il est possible de lister tous les textes et publications du site. Cela s'effectue à l'aide du moteur de recherche de ColdFusion sur les sources de données et de CFQUERY pour la transaction par groupe.

L'écran permettant l'exécution de ces fonctionnalités du système est présenté à la figure 77.

Dans le même ordre d'idées, on effectue des recherches bibliographiques sur le site. Cela se réalise par utilisation de CFOUTPUT, CFHTTP, CFFILE et VERITY de ColdFusion.

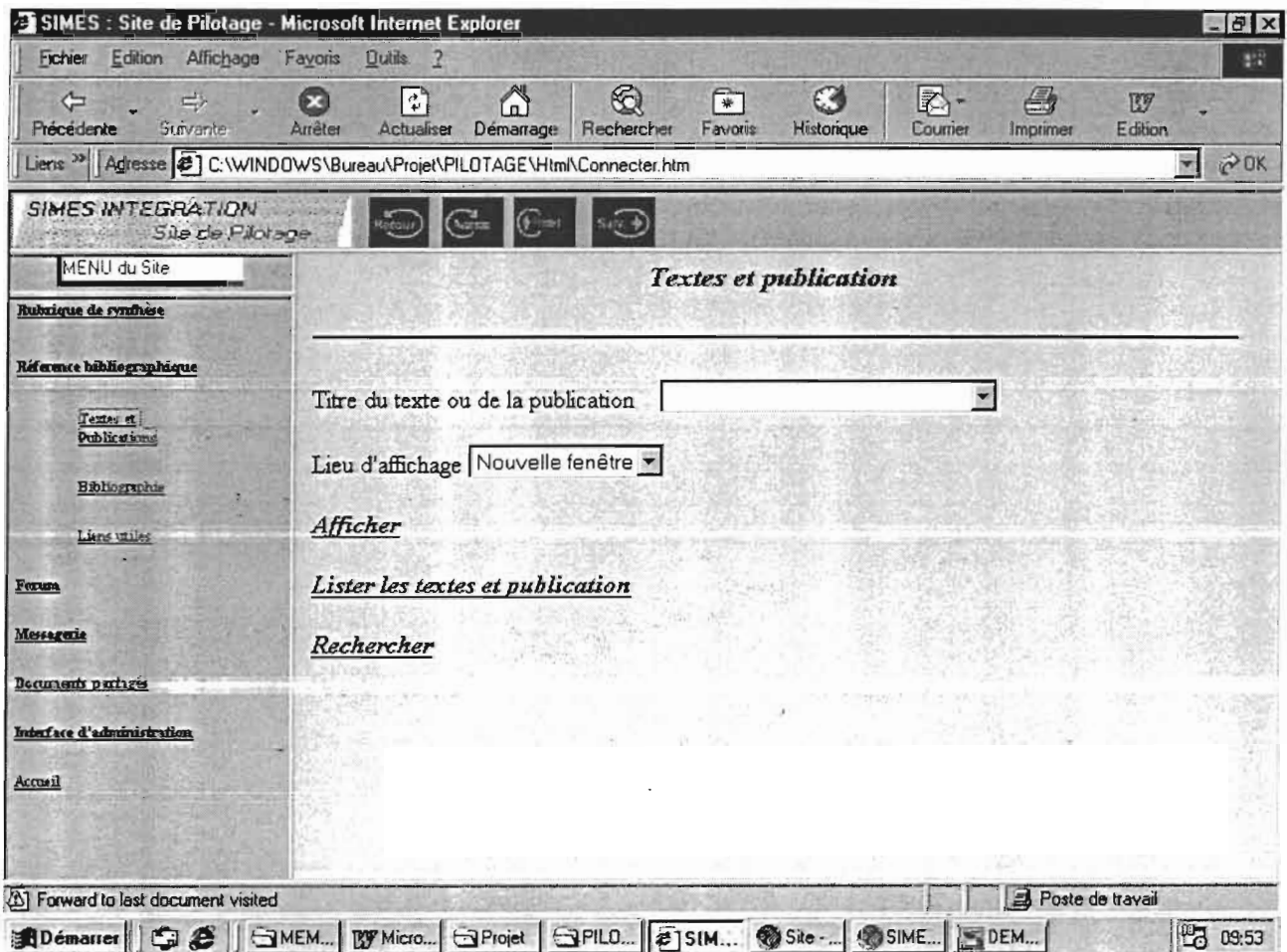


Figure 77 : Ecran des textes et publications

### III - 4 - 5 - Le gestionnaire de document

Il permet l'administration de l'entrepôt de données et l'organisation du site. On définit des liens à partir de la page d'application vers des textes, des opérations (sur les activités, les artéfacts), les fenêtres ou le nommage de certains outils du système.

Ces activités réalisées sont concrétisées à l'aide des écrans de gestion de documents et d'organisation du site présentés au niveau des figures 79 et 80.

Le tag CFOBJECT crée et utilise des objets COM ou CORBA. Encapsulé (le tag CFOBJECT) à CFMODULE on fait appel à des marqueurs personnalisés par écriture de scripts CFML. CFTREE permet la création d'arborescence.

Le transfert de document entre serveurs s'effectue par CFFILE qui permet aussi de télécharger des fichiers sur le disque de la machine cliente. L'échange des structures de données complexes (applications, données structurées, etc) se fait par le tag CFWDDX. Cet transfert est illustré à l'aide de la figure 78.

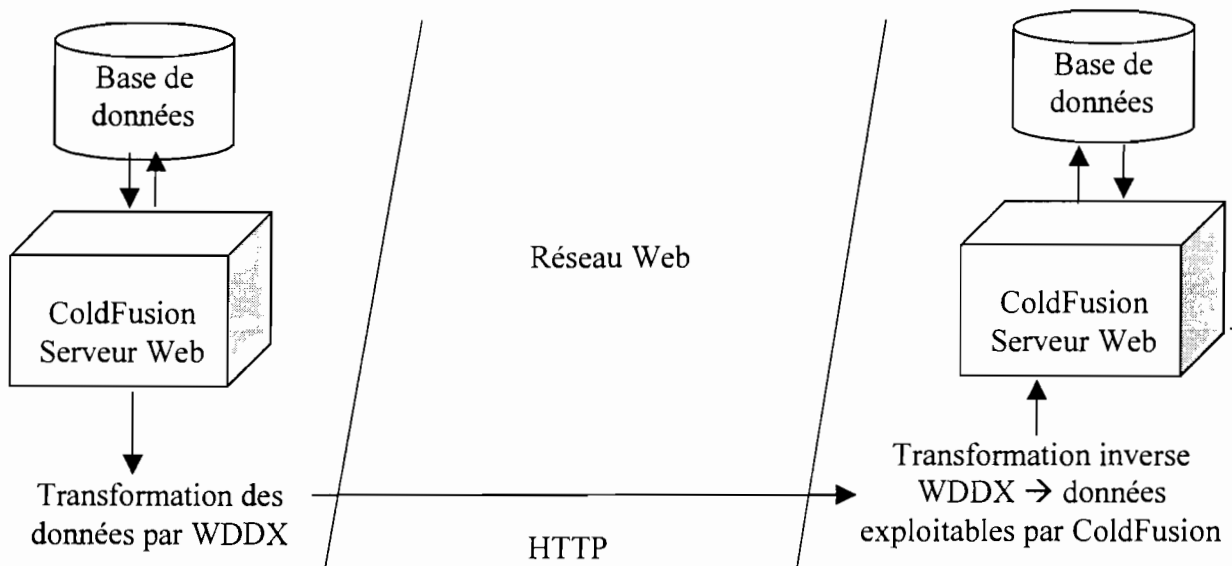


Figure 78 : Schéma de déplacement ou de transfert de données ou de document entre serveurs.

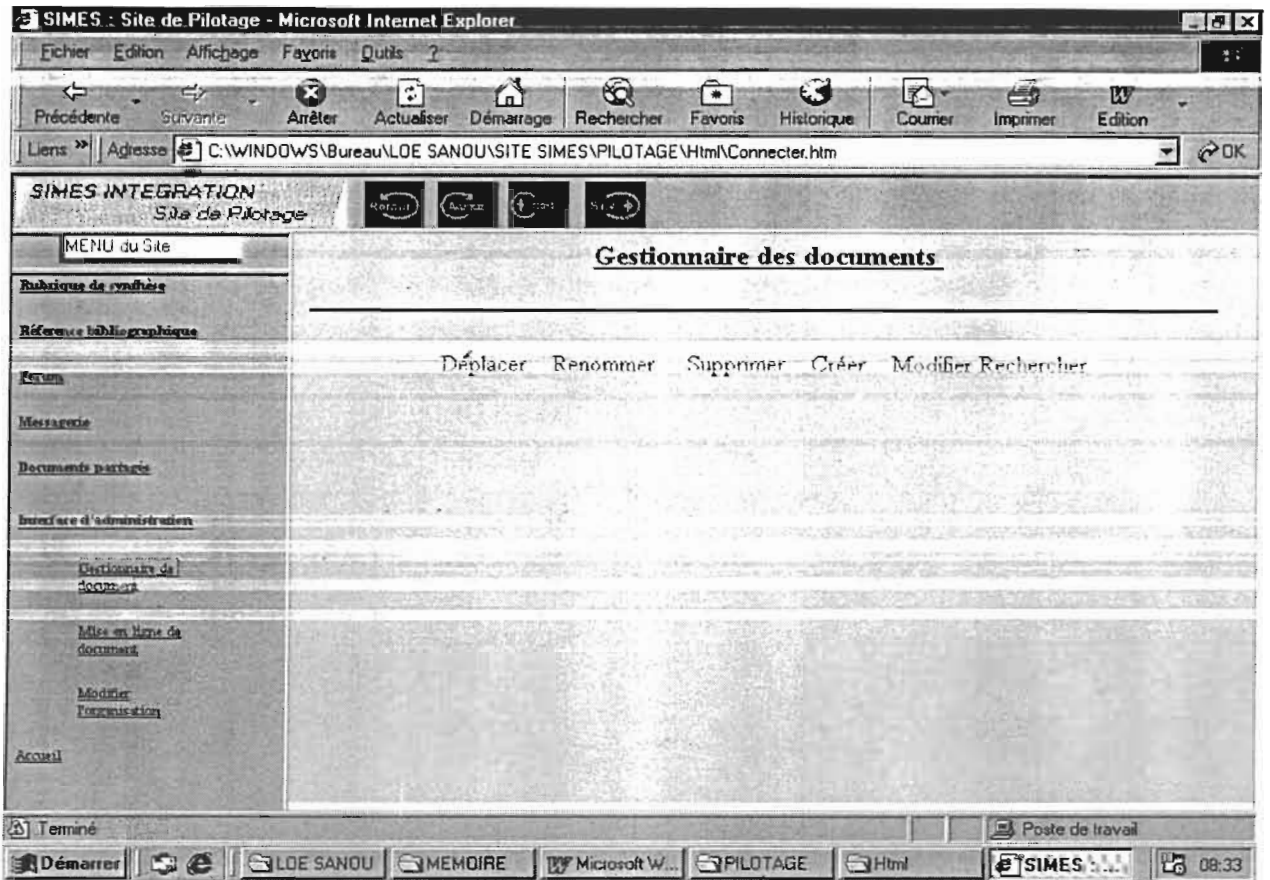


Figure 79 : Ecran du gestionnaire de document.

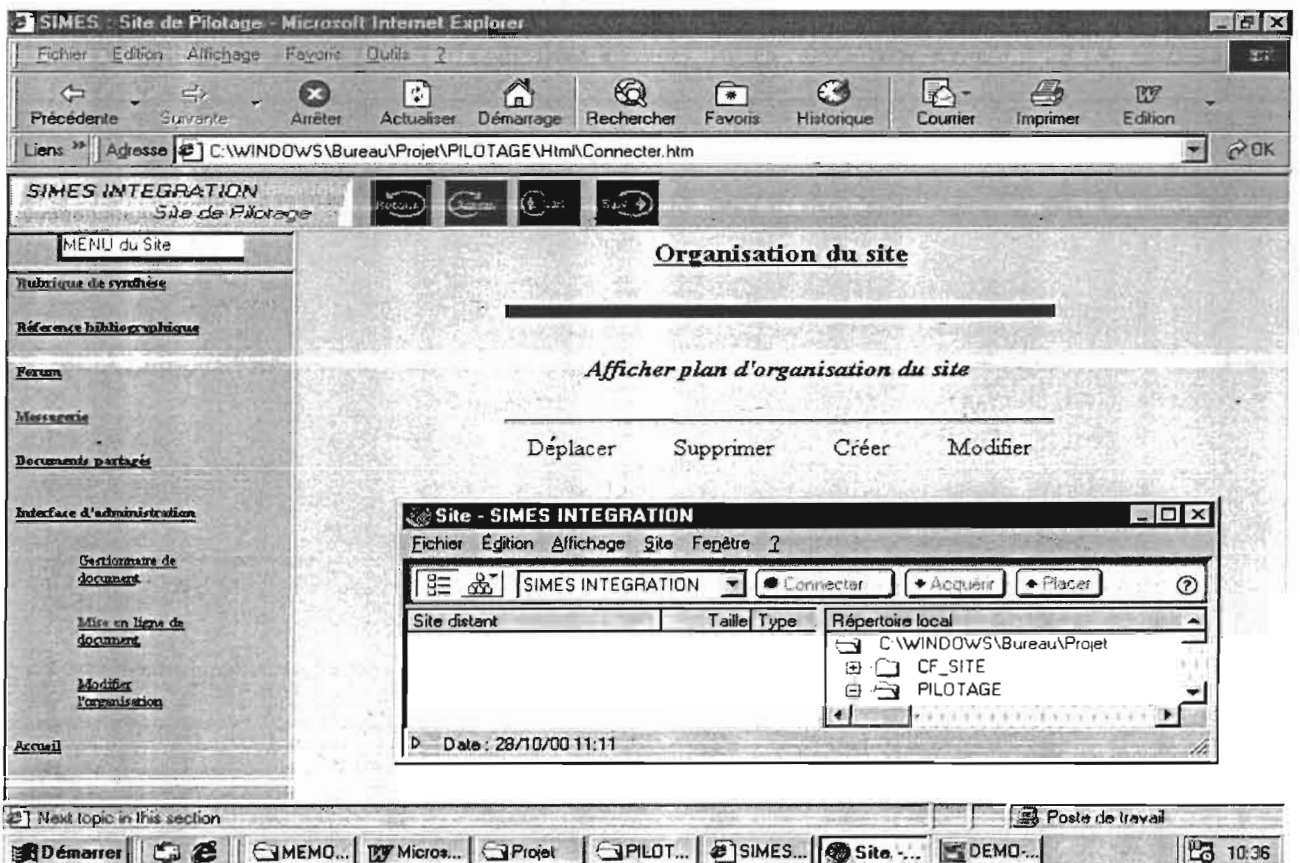


Figure 80 : Ecran d'organisation du site.



### III - 4 - 6 - La mise en ligne d'un document

Elle consiste à alimenter l'entrepôt de données et à administrer partiellement des paramètres système, à savoir les droits d'accès aux objets et leur visibilité. Ces objets concernés peuvent être des documents, des outils, des activités à placer dans un espace de stockage partagé destiné aux utilisateurs du système SIMES.

Pour un document, on utilisera un formulaire de saisie comportant l'auteur, l'appartenance, le titre du document, la catégorie, le mail de l'auteur, la date de dernière modification et le résumé du document. La figure 81 est une présentation de cet formulaire sous la forme d'un écran.

La création de nouveaux emplacements se fait à l'aide de CFTREE (création d'arborescence pour affichage hiérarchique de données en passant par des scripts : <CFSCRIPT> ..... </CFSCRIPT>). Avec CFFILE, on déplace, on copie ou on change le nom de fichier sur un serveur. Il permet aussi la lecture, la modification et la création de fichiers.

Figure 81 : Ecran de mise en ligne de document.

## CONCLUSION

Le système devrait permettre d'acquérir des données ou informations de différents types dans un environnement composé de différents systèmes de gestion de bases de données contenant diverses données (fichiers textes, images cartographiques, ...). Il devrait faciliter les échanges (de données, d'informations, de ressources, de documents) et la communication entre les entités du système. Le système devrait être administrable et pouvoir évoluer. Le site SIMES devrait être un site interactif et par conséquent les différents outils du système devraient être exécutés grâce aux navigateurs.

Afin d'y parvenir, nous avons dans notre travail pu effectuer la modélisation des structures statiques, la description des fonctionnalités à mettre en place et la réalisation partielle des outils de gestion de l'entrepôt de données et de ceux de gestion même du système.

Notre étude pourra être poursuivie afin de rendre fonctionnel le site en définissant les bases de données et en développant les scripts CFML pour les différentes fonctionnalités.

## BIBLIOGRAPHIE

### Ouvrages

- [Clavel 1996] G. Clavel, N. Mirouze, E. Pichon et M. Soukal *Java la synthèse*  
InterEditions 1996
- [Macary 1996] J-F. Macary et C. Nicolas *Programmation Java*  
InterEditions 1997
- [Pascal 1998] P. Pascal N. Kettani, D. Mignet, C. Rosenthal *De Merise à UML*  
Edition EYROLLES Octobre 1998
- [Sommerville 1998] I. Sommerville *Le génie logiciel* (Le best-seller du génie logiciel)  
Addison-Wesley 1998
- [Spaindour 1997] S. Spaindour & V. Quercia *WEBMASTER* Traduction de Eric Dumas  
Edition O'REILLY 1997
- [Theron 1988] P. Theron *Guide Pratique du Génie Logiciel*  
Edition Eyrolles 1988

### Rapports du projet SIMES

- [Kaba 2000] Ali B. KABA *Compte rendu des travaux des journées  
du 22 au 25 Mai 2000*  
SIMES 2000
- [Kaba 2000] Ali B. KABA *Rapport de réflexion introductive sur l'intégration  
dans le projet SIMES.*  
ESI – Bobo-Dioulasso (B.F.) 2000
- [Monga 1996] O. Monga *SIMES : Project Programme*  
France 1996
- [Monga 1997] O. Monga *Les livrables du projet SIMES*  
SIMES 961620 Tome I 1997

### Support de cours

- [Camille 2000] S. Camille *Copie des transparents du cours sur UML*  
Université Paris 1 – Sorbonne Février 2000
- [Grosz 2000] G. Grosz *Support de cours sur les Méthodes Orientées Objets*  
3ème année CICI – ESI – UPB Juin 2000