

# Conception des systèmes d'information des observatoires environnementaux : Une architecture de médiation

---

*Deliverable number : D1.2*

*Nature:P*

*Contractual Date of Delivery: 14 March 1998*

Task WP3.1 : Specification of the common software bases

*Nom du rédacteur*

*Patricia Dzeakou*

*Orstom*

*dzeakou@pophost.orleans.orstom.fr*

*JeanClaude DERNIAME*

*LORIA*

*derniame@loria.fr.*

## **Abstract:**

Ce document décrit une proposition de conception détaillée de la partie “interface utilisateur” et outils d'accès aux informations du projet SIMES. Il s'inscrit à la suite du document intitulé “Spécification de l'Architecture du Système d' Information SIMES” qu'il précise . Cette partie de l'architecture se place au dessus de la partie détaillée dans le document “Une Architecture pour les Environnements Hétérogènes Distribués: Procédés d'intégration” dont elle utilise les services.

This document describes a detailed proposal for the Man Machine Interface of SIMES project. It follows the document entitled “Specification of the Information System SIMES”. This part of the architecture takes place above the detailed section “An architecture for heterogeneous distributed environments” using its services.

## **Keywords:**

Système d'Informations, Intreface utilisateur, observatoire,



# Conception des systèmes d'information des observatoires environnementaux :

## Une architecture de médiation

<b>1. UNE APPROCHE DU TERME OBSERVATOIRE : LA STRATÉGIE "NAVIGATION - RECHERCHE - OBSERVATION" .....</b>	<b>5</b>
1.1.1. <i>La navigation</i> :.....	5
1.1.2. <i>La recherche</i> :.....	5
1.1.3. <i>L'observation</i> :.....	5
<b>ARCHITECTURE FONCTIONNELLE .....</b>	<b>6</b>
2.1. SOURCES D'INFORMATIONS .....	7
2.2. LES MÉDIATEURS WRAPPERS ET CONVERTISSEURS .....	7
2.3. LES BASES DE STOCKAGE D'INFORMATIONS .....	7
2.4. LES MÉDIATEURS OUTILS .....	7
<b>3. ARCHITECTURE LOGICIELLE .....</b>	<b>8</b>
<b>4. CONCEPTS DU SYSTÈME DE MÉDIATION .....</b>	<b>10</b>
4.1.1. <i>Le contexte</i> .....	10
4.1.2. <i>La session</i> .....	10
<b>5. LES CONCEPTS DE L'INTERFACE UTILISATEUR.....</b>	<b>11</b>
5.1.1. <i>La notion de métaphore</i> .....	11
5.1.2. <i>La notion de vue</i> .....	12
5.1.3. <i>Le profil</i> .....	12
<b>6. ADMINISTRATION .....</b>	<b>12</b>
6.1. ADMINISTRATION DU SYSTÈME D'INFORMATION.....	12
6.2. ADMINISTRATION DU RÉSEAU .....	12
6.3. ADMINISTRATION DE LA PLATE-FORME .....	12
6.4. ADMINISTRATION DES UTILISATEURS .....	13
<b>7. LES OUTILS DE DÉVELOPPEMENT .....</b>	<b>14</b>
<b>8. SCÉNARIO.....</b>	<b>14</b>
<b>9. PLANNING .....</b>	<b>14</b>



## **1. Une approche du terme observatoire : la stratégie "Navigation - Recherche - Observation"**

Le terme "observatoire" est utilisé dans de nombreuses disciplines et domaines de la vie courante avec diverses interprétations. Dans cette thèse, le cadre d'utilisation de ce terme est celui de la conception de systèmes d'information environnementaux. Plusieurs types de systèmes informatiques ont été utilisés jusqu'ici par les projets d'observatoires pour atteindre leurs objectifs (tableur, BD, SIG, outil de simulation, site Web, etc.).

Il se dégage de cette discussion que la notion d'observatoire en tant que système d'information laisse apparaître deux aspects :

- Un aspect "*conservatoire*" qui exprime le fait que dans un observatoire, il y a acquisition et conservation (accumulation) de données ou de documents et que cette information n'est pas mise à jour mais archivée. Ensuite, il y a restitution de cette information à l'utilisateur qui doit par conséquent avoir la possibilité d'*explorer l'information* archivée. Ce processus d'acquisition/restitution de l'information est également présent dans les problématiques de bases de données.
- Un aspect "*observatoire*" qui exprime le fait que les utilisateurs peuvent utiliser ces systèmes pour *effectuer des observations* (mesures, extractions, synthèses, etc.) sur les informations qui ont été archivées.

La principale contrainte du système étant qu'il soit accessible via le Web, la fonction d'exploration peut se décliner en fonctions de navigation et de recherche. La stratégie que nous suivons dans la thèse est donc celle de la conception d'une architecture et d'une interface, véritable charpente pour la pratique des trois fonctions essentielles d'un observatoire qui sont : la navigation, la recherche et l'observation.

Il apparaît également de souligner la différence entre ces trois fonctionnalités d'un "observatoire" :

### **1.1.1. La navigation :**

- l'utilisateur qui prend en charge la recherche, il peut ne pas connaître avec précision ce qu'il cherche
- l'utilisateur prend les décisions au fur et à mesure de la lecture
- Intérêt : interactivité de la lecture

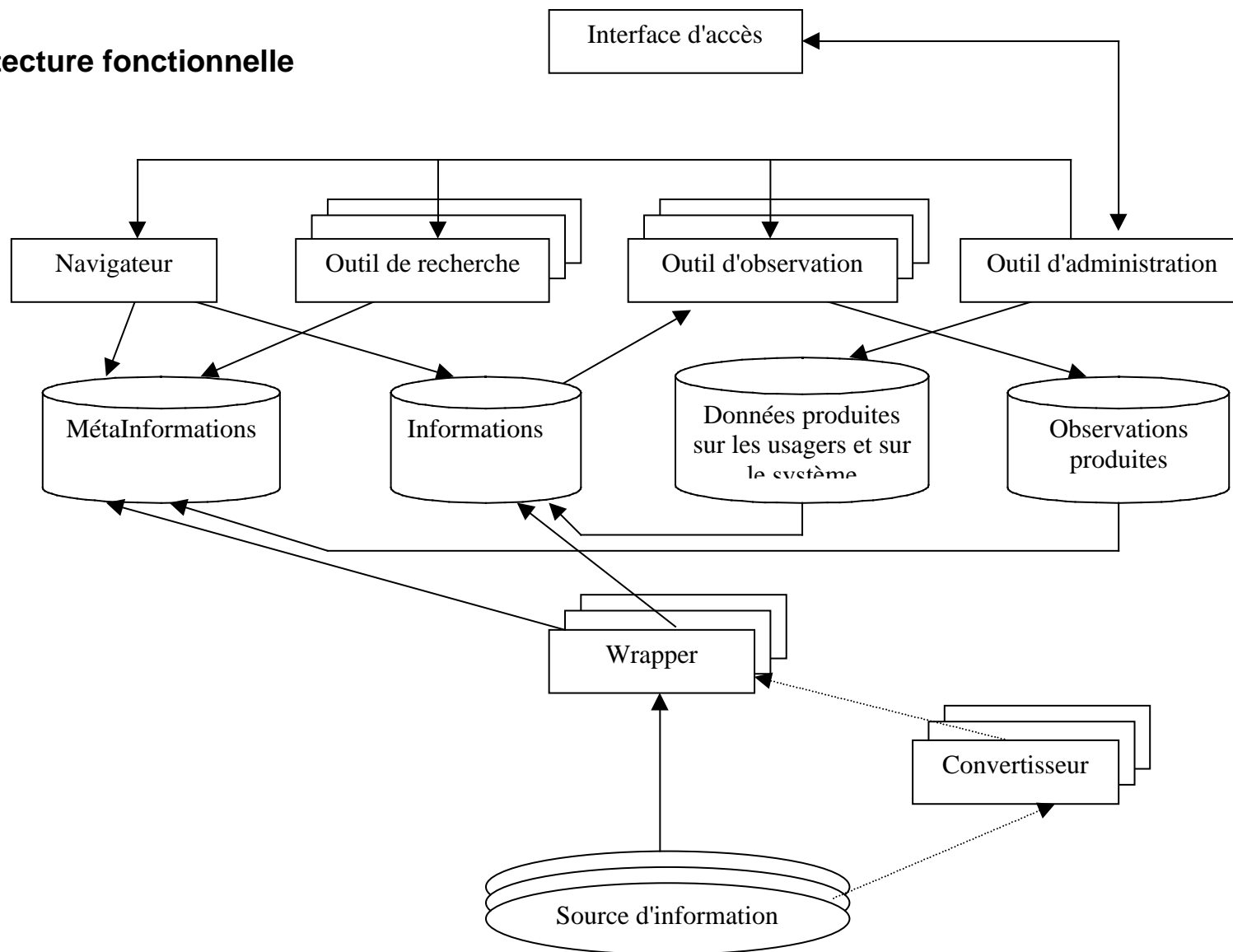
### **1.1.2. La recherche :**

- exigence de savoir caractériser ce que l'on cherche
- en gros, vous donnez les caractéristiques de ce que vous cherchez et un moteur se charge de faire la recherche pour vous
- Intérêt : on est dispensé du travail d'exploration

### **1.1.3. L'observation :**

On possède un corpus de "documents" ou d'informations et l'idée c'est de produire de nouvelles informations à partir des informations contenues dans ce corpus et d'enrichir le système.

## 2. Architecture fonctionnelle



L'architecture fonctionnelle que nous présentons est une architecture de médiation comportant 4 niveaux :

### **2.1. Sources d'informations**

Elles sont de divers types, pour l'instant on se restreindra aux types de données suivants : Tables de bases de données relationnelles, fichiers texte (Word et utilisation de styles), fichiers HTML et fichiers d'images. On ne souhaite pas trop contraindre ce niveau

### **2.2. Les médiateurs Wrappers et convertisseurs**

Ce sont des médiateurs dont l'objectif est de permettre une intégration de l'information provenant du niveau précédent et qui alimentent de ce fait les bases d'informations et de méta informations

### **2.3. Les bases de stockage d'informations**

Pour l'instant on distingue fonctionnellement 4 types d'informations qui seront stockées à ce niveau :

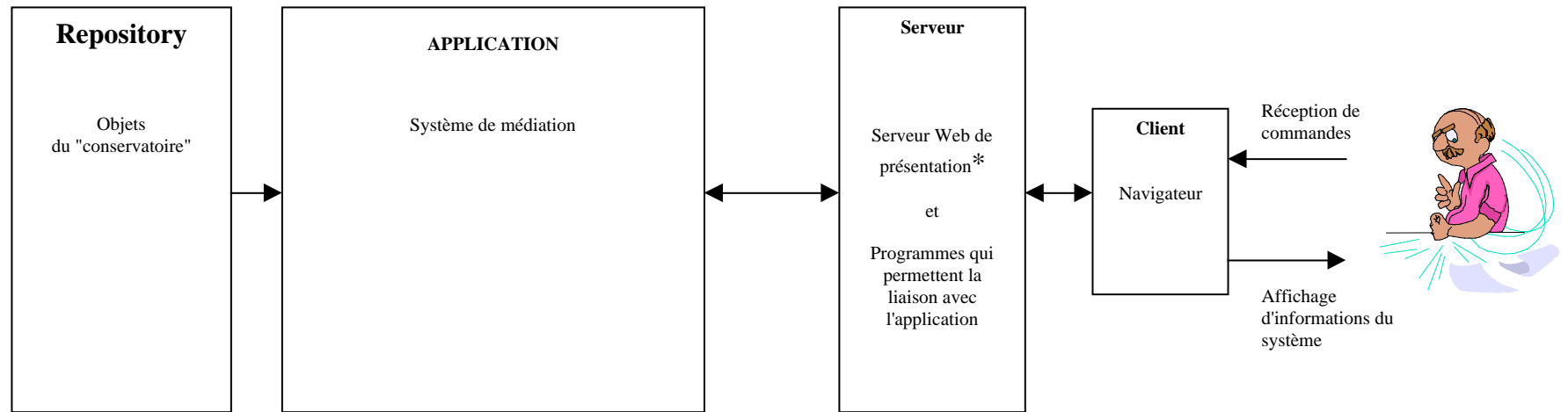
- La base d'informations : il s'agit de documents (ou de pointeurs sur des documents) avec des contenus provenant des sources d'informations. Elles sont intégrées via un médiateur wrapper. Pour l'instant, nous ne sommes pas précis sur la forme dans laquelle ils seront intégrés.
- La base de métainformations : il s'agit d'informations nécessaires à la gestion des informations stockées dans le réceptacle précédent et intégrées également via un wrapper.
- Les données sur les usagers et sur le système : ce sont des informations recueillies par l'outil d'administration. En tant que données sur l'exploitation du systèmes et sur la consultation des contenus, elles font également partie de l'information qu'il est nécessaire de rendre accessible dans un système d'information de type observatoire.
- Les observations produites : il s'agit de données (résultats) issues des observations (faites via un outil d'observation) et qui alimenteront les bases d'informations et de métainformations.

### **2.4. Les médiateurs outils**

On distingue à ce niveau deux types de médiateurs :

- L'outil d'administration qui est un outil dédié à la gestion proprement dite du système (enregistrement des usagers, des outils, coordination, etc.)
- Les outils tournés vers les besoins de l'utilisateur. Fonctionnellement, on en distingue 3 :
  - Le navigateur qui proposera à l'utilisateur une interface lui permettant de naviguer (en réduisant le phénomène de désorientation) dans les informations archivées
  - Les outils de recherche de recherche qui pourront être des moteurs de recherche classique (Exemple : Excite) ou plus spécialisés
  - Les outils d'observation qui permettront à l'utilisateur d'effectuer des observations sur les données archivées via des traitements (statistiques, traitements d'images, traitement de textes, etc.).

### 3. Architecture logicielle



\* Le serveur Web met essentiellement à la disposition de l'utilisateur des pages de dialogue avec le système





## 4. Concepts du système de médiation

Deux concepts principaux se dégagent de cette approche de médiation :

### 4.1.1. Le contexte

Un contexte est défini comme un ensemble de liens sur un sous-ensemble (de documents au sens large) du système d'information. Le contexte peut être un ensemble de liens sur tout le système d'information, on peut imaginer par exemple que lorsqu'un utilisateur se logge, le contexte qui lui est attribué soit le système d'information tout entier. On peut le caractériser de la manière suivante :

Attributs : identifiant du contexte précédent, liste de pointeurs sur des "documents", représentant le contexte, opérations qui ont permis l'obtention de ce contexte

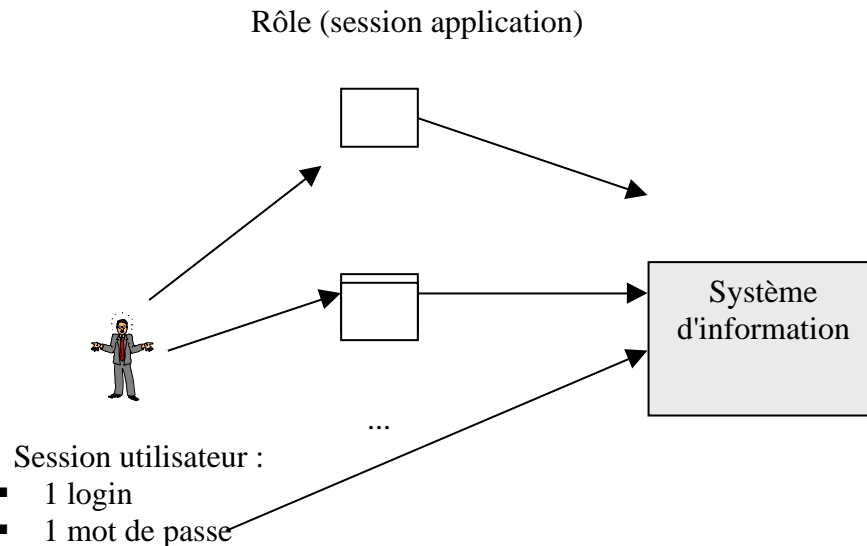
Opérations :

- Créer()
- Nommer(nom\_contexte) : attribue un nom (identifiant) au contexte
- Sélectionner(document) -> contexte  
Nous avons considéré l'action élémentaire de la navigation qui est celle de la sélection d'un document. La navigation se matérialise dans le système par une succession de contextes.
- Rechercher(critères) -> contexte
- Observer(outil, paramètres) -> Informations  
Observer signifie appeler un outil qui va travailler sur le contexte courant, qui va extraire ou produire des informations et qui va éventuellement les stocker dans le système d'information. Il doit donc être possible au niveau de l'architecture d'appeler des outils extérieurs.  
Les "outils observateurs" prennent en entrée un contexte qui peut être spécifique de l'outil. Par exemple à un outil de traitement d'images correspond un contexte contenant des liens sur des document images. Mais on peut tout à fait imaginer qu'il y ait des outils qui utilisent un contexte contenant des liens sur des documents de types différents.  
Conséquence : Nécessité de pouvoir ouvrir une fenêtre pour l'outil sans quitter le navigateur et de prévoir à la fin le retour au navigateur et le rangement éventuel des observations faites (relatif au travail de G. Kouamou)
- Sauver() : sauvegarde du contexte et des caractéristiques qui ont permis d'y aboutir
- Détruire() : destruction du contexte
- AfficherContexte(Interface, type de vue) : provoque la demande - à l'interface - d'affichage du contexte. Similaire à l'opération Voir ?

### 4.1.2. La session

La session peut être définie comme la durée pendant laquelle on maintient la connexion entre un utilisateur et le système d'information. C'est cette permanence de la connexion qui donne la possibilité de conserver des informations sur le déroulement de la session. Nous avons fait les choix suivants :

- Une session par utilisateur à un instant donné
- Des rôles pourront éventuellement être définis, permettant à un utilisateurs de se connecter avec le même login et mot de passe (donc une même session utilisateur) sous plusieurs rôles différents. Un rôle jouant ici le rôle d'une session application.
-



A l'ouverture d'une session un utilisateur choisit le *mode d'utilisation* (différent du rôle) du système. On peut distinguer trois modes d'utilisation du système : le mode administrateur (administration du système), le mode observateur (auquel on s'intéresse car c'est lui qui donne sur les 3 fonctionnalités "naviguer/rechercher/observer" et le mode fournisseur qui sera utilisé par les propriétaires de sources d'information.

Une session est un objet dont nous donnons une première définition non exhaustive:

Attributs : contexteInitial, contexteCourant, utilisateur, boutons, menus

Opérations :

- Démarrer() -> initialisation de la session
- Arrêter() -> sauvegarder et arrêt de la connexion
- Sauvegarder() -> stockage d'informations relatives à la session
- Suspendre() -> entraîne la sauvegarde de la session
- Reprendre() -> restauration de la session
- Restaurer(nom\_contexte) : restauration d'un contexte de la session ou d'un contexte sauvegardé . Cette opération fait appel à l'opération  
Changer\_contexte(nom\_contexte)
- Utilisateur() -> retourne le nom de l'utilisateur auquel appartient la session
- ContexteInitial() -> retourne l'identifiant du contexte initial
- ContexteCourant() -> retourne l'identifiant du contexte courant
- Changer de contexte(nom\_contexte) -> remplace le contexte courant par le contexte identifié par nom\_contexte
- ContextePrécédent() -> Change le contexte de la session en choisissant comme nouveau contexte le contexte précédent

## 5. Les concepts de l'Interface utilisateur

### 5.1.1. La notion de métaphore

Ce concept devrait être plus précisé en fonction de ce qui se fait dans le domaine des interfaces utilisateur pour qui ce concept est familier. Il suggère à priori des modes opératoires i.e. qu'on définisse pour chaque type de métaphore les opérations correspondantes. C'est le cas par exemple dans la métaphore du bureau chère au Macintosh et à Windows où des opérations telles que classer, trier, expédier, calculer, jeter, etc. sont suggérées à l'utilisateur.

Quelques exemples précis devraient nous permettre de mieux cerner ce concept. Nous avons abordé des exemples relatifs :

- Au temps : avec des opérations telles que l'obtention d'informations relatives à une date précise, comprises dans un intervalle ou encore d'un âge approximatif.
- A la géographie : avec des opérations relatives au zoom, la restriction, l'échelle, la notion d'alentours, etc.

Nota bene : En fait, la vue présente le contexte et via l'exécution d'une de ses opération on aboutit à un nouveau contexte.

Question : Le fait d'exécuter une opération via une vue est-il considéré comme une navigation ou comme une recherche ?

### **5.1.2. La notion de vue**

La vue est une façon de présenter un contexte. Elle comporte deux aspects principaux :

- Présentation caractéristique (utilisation d'une métaphore ?)
- Catégories de vues prédéfinies

Le problème de la différence entre contexte et vue

Au départ, nous avons classé la vue comme concept du système de médiation. Puis il nous est apparu qu'en fait il n'y avait pas de grande différence entre vue et contexte. En effet, quand on passe d'un contexte à une vue, on ne change pas de contexte, la vue n'est qu'une façon d'afficher le contexte. Il en découle que qu'une vue était une notion plus proche de l'interface utilisateur en ce sens que le fait de demander à l'interface d'afficher une vue en utilisant une métaphore donnée a pour effet de d'afficher le contexte en utilisant cette métaphore.

### **5.1.3. Le profil**

Nous avons évoquer la possibilité d'avoir :

- Soit des profils avec un certain nombre de caractéristiques prédéfinies : préférences ou centres d'intérêt, contexte(s) spécifique(s), vue(s) spécifique(s)
- Soit des profils émergents qui peuvent donc évoluer au fur et à mesure de l'utilisation du système

Différent de Rôle ? : je pense que les deux notions sont proches (en termes de caractéristiques)

## **6. Administration**

On peut distinguer 4 types d'administration au sein de ce système d'information. Nous donnons un bref aperçu non exhaustif de ce à quoi elles font référence :

### **6.1. Administration du système d'information**

- Droits d'accès, protection des objets
- Ajout et suppression de documents
- Organisation de l'information et les tables d'accès

### **6.2. Administration du réseau**

- Administration des sites qui apportent de l'information (sites où sont localisées les sources d'information)
- Administration de leurs contenus
- Tables d'accès

### **6.3. Administration de la plate-forme**

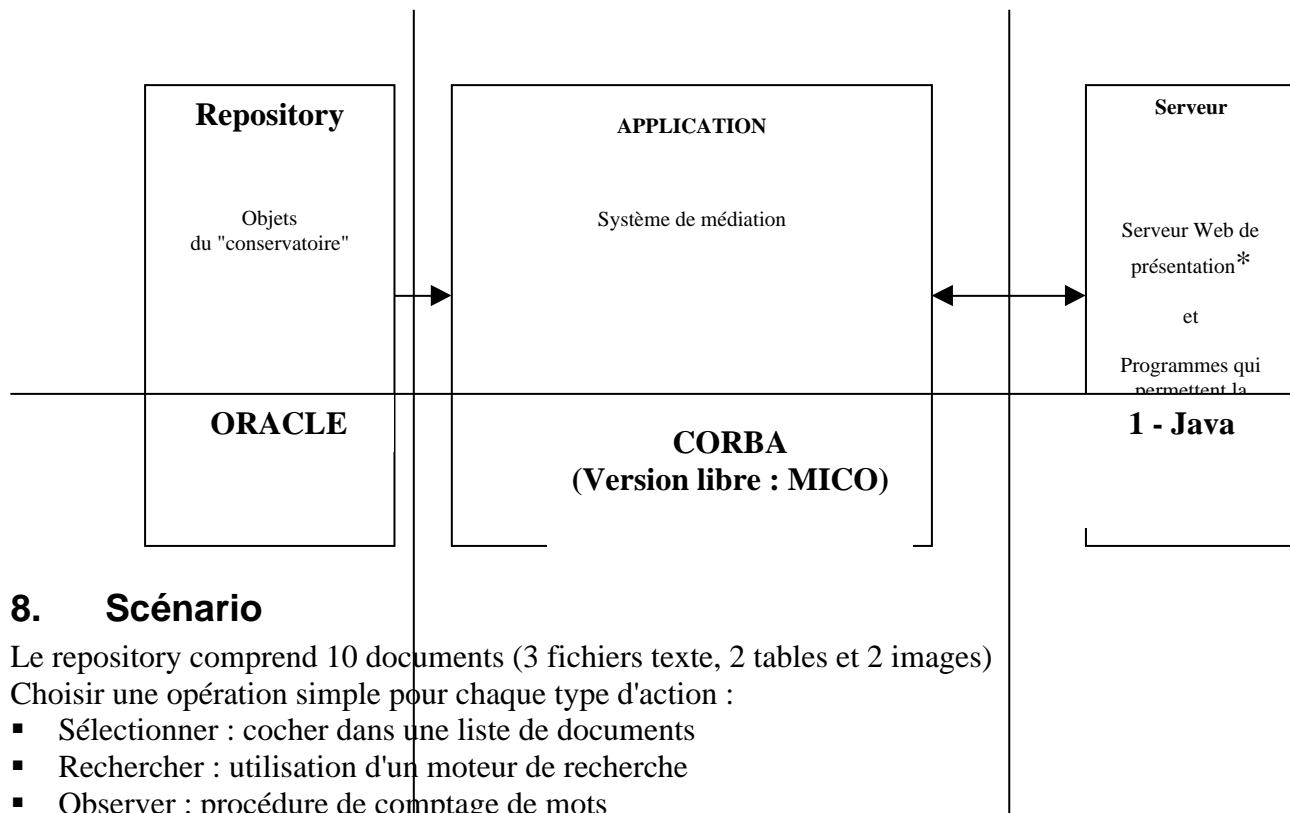
Administration des outils disponibles (enregistrement, installation, désinstallation, gestion des licences, configuration des logiciels, etc.)

#### **6.4. Administration des utilisateurs**

- Enregistrement des utilisateurs (login, mot de passe)
- Création d'un profil
  - Préférences, centres d'intérêt
  - Contexte initial / contexte par défaut
  - Vues /vue par défaut
  - Historique de la session
  - Push ?

Nota bene : Cette thèse ne s'occupe pas de tous les problèmes d'administration cités ci-dessus. Etant axée sur le système de médiation, elle ne s'intéresse qu'aux aspects de l'administration en relation avec cette architecture de médiation.

## 7. Les outils de développement



## 8. Scénario

Le repository comprend 10 documents (3 fichiers texte, 2 tables et 2 images)

Choisir une opération simple pour chaque type d'action :

- Sélectionner : cocher dans une liste de documents
- Rechercher : utilisation d'un moteur de recherche
- Observer : procédure de comptage de mots

Scénario (qui devra pouvoir être exécuté par le premier prototype) :

L'utilisateur se connecte avec comme contexte initial au départ tous les documents du repository.

1. Il indique ensuite quelle est la vue qu'il souhaite (appliquer une vue qui renvoie le même contexte).
2. Il sélectionne (en cochant dans la liste des documents correspondant au contexte initial) 3 documents (le nouveau contexte courant contient alors N-3 documents)
3. Il fait une recherche sur ce document (utilisation d'un moteur) lui permettant de réduire son contexte.
4. Il change de vue (restriction du contexte au textes)
5. Il effectue une observation sur le nouveau contexte: compter le nombre de mots ou compter le nombre de fois où le mot X apparaît

## 9. Planning

Le plan de travail pour les deux mois à venir est le suivant :

**Mardi 2 décembre 1998 au plus tard**

- remise du présent document à JCD

**Fin décembre :1998**

- UML
- Définir les "use-case" et diagrammes de transition correspondant au mode "observateur". Ne s'intéresser qu'aux composants et aux acteurs.
- Préciser les types de vues et la notion de métaphore

- Concevoir l'IDL (Interface Definition Language) i.e. pour chaque concept (classe d'objet) déterminer les signatures de ses opérations

**Fin Janvier 1999**

- Premier prototype en Java pouvant exécuter le scénario défini dans le chapitre précédent et parallèlement apprentissage de Java