SIMES - 961620
Système d'Information Multimédia
Pour l'Environnement Subsaharien

Tools integration

Deliverable number : D 20
Nature:D
Contractual Date of Delivery: 14 May 2001
Task WP3 : Tools integration within a user friendly generic system

Authors :

Georges Edouard KOUAMOU
University of Dschang
e-mail: kouamou@simes.uninet.cm

Jean-Claude Derniame
INRIA
LORIA, Campus Scientifique. B.P. 239
54506 Vandoeuvre-lès-Nancy Cedex (France)
derniame@loria.fr

Abstract :

Deliverable D20 was initially meant to cover the "Data acquisition tools integration" while D19 was to cover the "Data processing tools integration". This splitting was assessed as not adequate, since on the one hand there would have been a lot of redundancy between both documents, and on the other hand, the data integration would not have been covered properly. D19 and D20 were thus renamed respectively "Data integration" and "Tools integration" : this revised splitting ensures now a good coverage of both data and tools integration issues.

As a complement to deliverable D19, the present document D20 describes the approach which has been adopted to define the integration of software tools for data acquisition and processing, within the SIMES platform. This approach aims at incorporating the existing standards and mechanisms rather than replacing them. It is therefore based mainly on CORBA, Java and HTML technologies and we are using freeware as much as possible for the implementation.

SOMMAIRE

LISTE DES FIGURES

1 INTRODUCTION

1.1 PREAMBULE

Le document D20 était initialement intitulé "Data acquisition tools integration", tandis que le D19 était intitulé "Data processing tools integration". Ce partage a été jugé inadéquat, du fait qu'il entraînait de nombreuses répétitions et empêchait de couvrir correctement le sujet de l'intégration des données. D19 et D20 ont donc été renommés respectivement "Data integration" et "Tools integration" : ce nouveau partage assure maintenant une bonne couverture de l'ensemble des aspects touchant à l'intégration des données comme à celle des outils de traitement.

1.2 CONTENU DU DOCUMENT

Ce document décrit l'approche d'intégration d'outils logiciels adoptée pour la plate-forme SIMES. Il se situe dans le cadre du WP3 et concerne aussi bien les outils d'acquisition que de traitement de données. La plate-forme qui en résulte s'appuie sur des standards ouverts tels que HTML, Java et CORBA ; et l'implémentation utilise les logiciels libres autant que possible. Ce qui fait de SIMES une plate-forme extensible et réutilisable à moindre coût.

Ce document complète ainsi le D15 et le D19.

1.3 CONTEXTE

Le projet SIMES "Système d'Information Multimédia pour l'Environnement Subsaharien", programme INCO-DC 961620, est un projet financé conjointement par l'Union Européenne et la Banque Mondiale (projet conjoint WISE-DEV). Il se situe dans le cadre du programme Européen INCO-DC dont l'objectif est de développer un partenariat avec les pays en développement.

SIMES WISE-DEV vise à associer des partenaires africains et européens pour l'application et l'adaptation des nouvelles technologies de l'information à la compréhension et à la maîtrise de l'environnement sub-saharien. Ses objectifs scientifiques et technologiques devront favoriser les compétences particulières en informatique appliquée aux informations et problèmes touchant à l'environnement.

D'une part, l'Observatoire de la pêche dans le Delta Central du Niger et la gestion de la vallée du fleuve Sénégal sont des dispositifs de collecte de données de nature diverse et de formats variés. Ces données peuvent être par exemple des relevés de terrain, des résultats d'enquêtes, des images aériennes ou satellitales. Leur manipulation nécessite :

- Des outils de traitement d'images, et plus spécialement le mosaïquage et la segmentation.
- Des applications spécifiques à l'exemple du Modèle Intégré du Delta (MIDIN) qui prend en compte les différentes activités agro-pastorales menées par les population, la migration des pêcheurs, la biomasse, etc.

D'autre part, les Herbiers demandent une méthodologie de gestion des collectes des spécimens d'espèces végétales. Cette méthodologie s'attèle davantage à la structuration et à la synthèse des informations.

Que ce soient des applications développées expressément ou des outils acquis par ailleurs, tous correspondent aux principales fonctionnalités rencontrées dans les observatoires de l'environnement. Afin de minimiser les coûts du développement, et d'assurer l'indépendance économique et technologique de la maintenance, les logiciels libres (dans le sens de la Free Software Foundation) constituent la base des réalisations du projet, tant qu'ils sont compatibles avec les fonctionnalités du système.

La plate-forme SIMES, à l'instar des systèmes d'information sur l'environnement, est caractérisée par l'hétérogénéité, la diversité et la distribution des données et des outils. Leur manipulation (acquisition, stockage et traitement) nécessite des outils logiciels variés tels que l'analyse statistique, les algorithmes de simulation, les Systèmes de Gestion de Base de Données (SGBD) relationnels et/ou Orientés Objet, ou les Systèmes d'information Géographique (SIG). Face aux problèmes de représentation des données et du choix des outils de manipulation et de traitement des informations, il existe rarement une solution unique ; chaque équipe ayant ses habitudes de programmation, ses préférences de langage et de plate-forme.

L'idée maîtresse consiste à partir des expériences menées dans les applications pilotes pour proposer une plate-forme logicielle qui fournisse un support aux activités des observatoires sur l'environnement, et qui offre aux thématiciens un cadre unifié de travail. Une telle plate-forme doit être générique, extensible et intégrée.

1.4 STRUCTURE DU DOCUMENT

Dans la suite de ce document, nous commencerons par une présentation des concepts en insistant sur la notion d'outil et d'activité. Cette présentation est complétée par des aspects conceptuels qui s'appuient sur UML et les « Design Patterns ». Elle s'articule suivant deux étapes : la structure qui présente le concept d'outil et la dynamique qui décrit les objets actifs et les interactions entre les différentes entités de la plate-forme. La section 3 détaille l'architecture fonctionnelle qui repose sur le couplage des protocoles HTTP du Web et IIOP de CORBA. Ce couplage est complété par l'utilisation des composants qui relève de l'état de la technologie actuelle. Enfin les sections 4 et 5 portent sur les guides d'utilisation de la plate-forme à travers un outil de traitement de données.

2 LA PLATE-FORME SIMES : L'INTERFACE UNIQUE D'APPEL D'OUTILS.

2.1 LES OUTILS

Un **outil** est un programme accessible à travers la plate-forme. Ses fonctionnalités sont vues comme des services offerts aux utilisateurs. L'invocation d'un outil sera appelée une activité et elle peut être locale ou distante. Nous distinguons quatre catégories principales d'outils :

- les programmes invocables en ligne de commande (Command Line Application : en abrégé CLIA)
- les objets JAVA,
- les objets CORBA,
- les composants COM.

Pour ceux des outils disposant d'une interface de programmation (API), des efforts doivent être faits pour les conformer à l'une des catégories ci-dessus. Les outils qui ne peuvent être déconnectés de leur interface utilisateur graphique sont exclus du processus d'intégration que nous mettons en place.

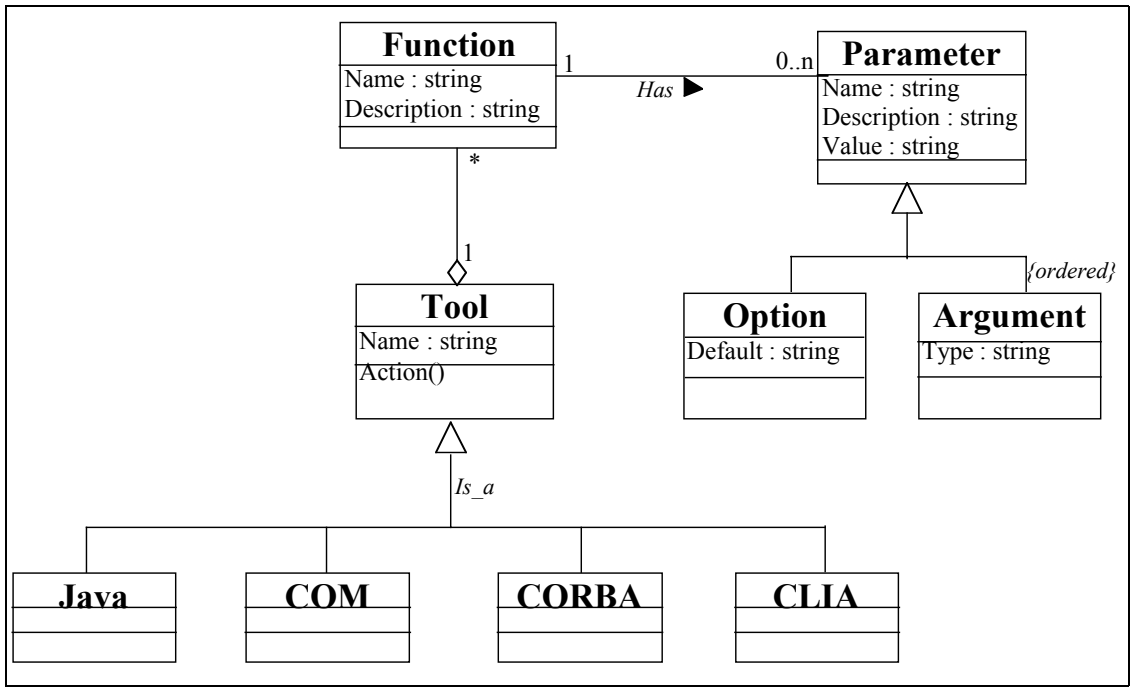


Figure 1. *Le modèle d'Outil*

2.1.1 Les outils « ligne de commande »

Les outils de la première catégorie sont constitués d'un ensemble de « *points d'entrée* » que nous appellerons fonctions. Une fonction a des paramètres qui peuvent être optionnels ou obligatoires. Les paramètres optionnels seront appelés « *options* » et les paramètres obligatoires seront appelés *« arguments »*. Ainsi, chaque fonction peut être activée par une commande ayant une forme particulière et une syntaxe précise.

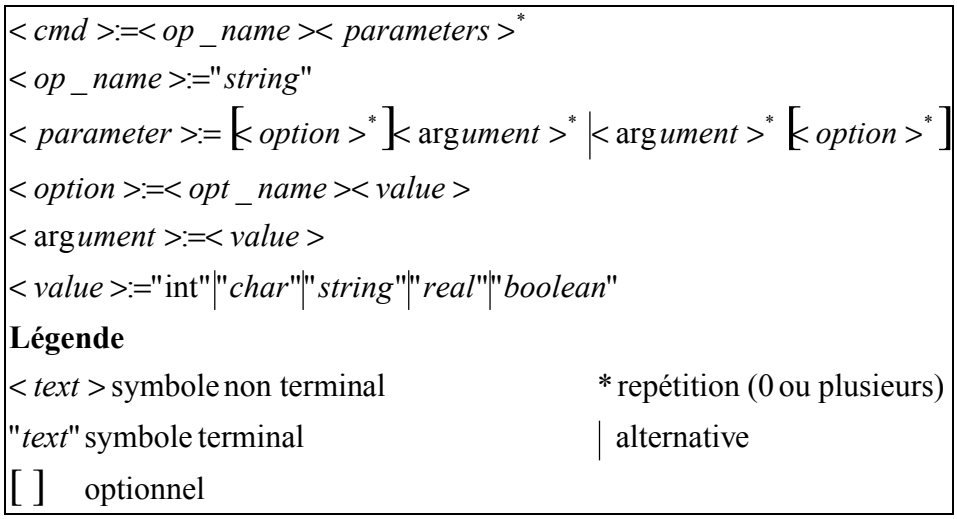


Figure 2. *Syntaxe du point d'entrée d'un outil*

2.1.2 Les objets Java, CORBA et les composants COM

Les outils appartenant à ces trois catégories disposent des interfaces. Une *interface* est une collection d'opérations (similaires aux points d'entrée des outils de la catégorie précédente) qui spécifie les services offerts par l'outil. Une interface décrit le comportement visible de l'extérieur. Les opérations sont définies au niveau des interfaces par un ensemble de spécifications qui déterminent leur signature.

< *operation* >::=< *op _ name* >"("< *parameter* >*")"
< *op _ name* >::="*string*"
< *parameter* >::=< *param _ type* >< *param _ name* >
< *param _ type* >::="*string*"|"int"|"*real*"|"*char*"|"*boolean*"|< *type _ composés* >
< *param _ name* >::="*string*"
< *type _ composés* >::="*array*"|"*struct*"
Légende
* repétition (0 ou plusieurs) | alternative
"text" symbole terminal < text > symbole nom terminal

Figure 3. *Syntaxe de la signature d'une opération d'interface*

Les points d'entrée et les opérations sont regroupés sous le terme de « fonction » afin d'unifier leur présentation au niveau de l'interface utilisateur. La connaissance de leur syntaxe permet de transformer chaque service activé au niveau de l'interface utilisateur, en une commande interne qui correspond à une fonction appartenant à un outil.

2.2 LES ACTIVITES

Une activité est une séquence d'appels d'opérations (éventuellement limitée à une seule). Une opération est une procédure élémentaire consistant au lancement d'une commande mise à la disposition de l'utilisateur, à l'invocation d'un outil en local ou à distance. Une activité peut être interactive (nécessitant la présence de l'utilisateur qui l'a lancée) ou non. Une activité peut être immédiate (sa durée n'excède pas la session de l'utilisateur qui l'a lancé) ou non. Plusieurs activités peuvent être lancées en parallèle.

Exemples :

- Exploration locale (interactive, immédiate ou médiate)
- Invocation d'un outil ou d'une suite d'outils à distance
- Opération d'administration du site

2.3 L'INTERFACE D'INVOCATION

2.3.1 Organisation des services de la plate-forme SIMES

Les services de la plate-forme SIMES sont définis pour permettre le partage de logiciels et de données dans un environnement distribué. Même si ces objets (logiciels et données) sont en général considérés comme de simples fichiers binaires ou comme une séquence d'octets, la démarche de stockage consistera aussi à fournir des descriptifs de ces objets

pour permettre leur exploitation (visualisation, exécution, exploration, …). Les services proposés se résument en des mécanismes que nous organisons en trois catégories :

- Les mécanismes de construction de l'interface utilisateur.
- Les mécanismes de base du système SIMES.
- Les mécanismes du serveur Internet et/ou Intranet.

Les interactions entre ces catégories de mécanismes sont décrites par le schéma ci-dessous :

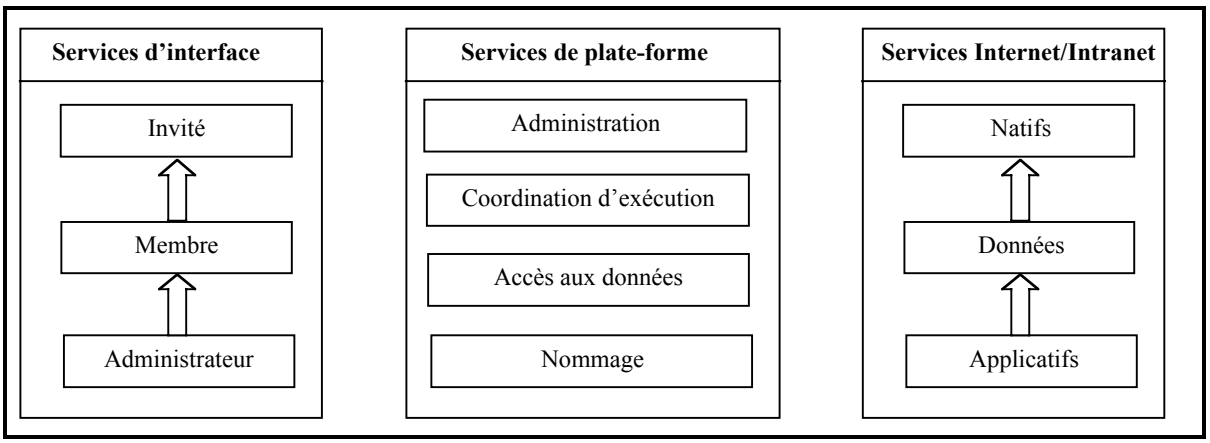


Figure 4. *Les différentes catégories de services de la plate-forme*

Les services d'interface permettent les interactions entre les utilisateurs et la plate-forme. Ils offrent toutes les fonctionnalités nécessaires pour exécuter les services disponibles au niveau de la plate forme. Bien évidemment, la visibilité de ces fonctionnalités dépend du rôle de chaque utilisateur. Ainsi les fonctions de manipulation des entrepôts qui sont offertes par le service d'accès aux données, sont accessibles pour un membre ou un administrateur mais invisibles pour un invité ; le service d'administration n'est possible que pour le rôle administrateur.

Les services de la plate-forme sont des mécanismes de haut niveau qui sont utilisés par les services d'interface. Ils comprennent :

- les services d'administration du système,
- les services de coordination d'exécution. Ces services ont trait aux éléments réactifs du système SIMES. La coordination d'exécution des différents processus consistera à gérer les états successifs des objets sur lesquels les services proposés ici sont appliqués, mais aussi à répondre aux sollicitations que les utilisateurs adressent au système. On distingue le traitement des commandes du système, l'exécution des outils, l'exécution des activités, la gestion de session.
- les services d'accès aux données. Ce sont des services utilisateurs qui concernent essentiellement les mécanismes de stockage (transférer des objets décrits par l'utilisateur sur un site SIMES) et de consultation des objets (les rendre visibles et accessibles par les autres utilisateurs s'il s'agit d'objets publics).

- le service de nommage des objets : c'est un service de nom statique qui permet d'attribuer un nom logique unique à un objet déposé sur la plate-forme SIMES, de le décrire, de le retrouver et d'y accéder.

Les services Internet et Intranet regroupent les protocoles de communication (TCP/IP, HTTP, IIOP, etc.) entre les différents sites, les mécanismes de partage et d'échange d'information et/ou de ressources (NFS, SAMBA) entre les ordinateurs, avec pour corollaire l'encapsulation implicite des services du système d'exploitation, l'utilisation de tous les mécanismes sous-jacents à savoir : la résolution des noms, la localisation des ordinateurs et des objets, etc.

2.3.2 Protocole Abstrait d'appel d'outil

Nous avons présenté précédemment les différentes catégories d'outils qui pourront cohabiter sur la plate-forme. Pour le client (qui peut être le service d'interface utilisateur, une activité prédéfinie, ou un autre outil) nous devons présenter une interface uniforme, indépendante du type d'outil et du mécanisme d'interaction sous-jacent. La description de cette structure utilise les « Design patterns ».

La structure

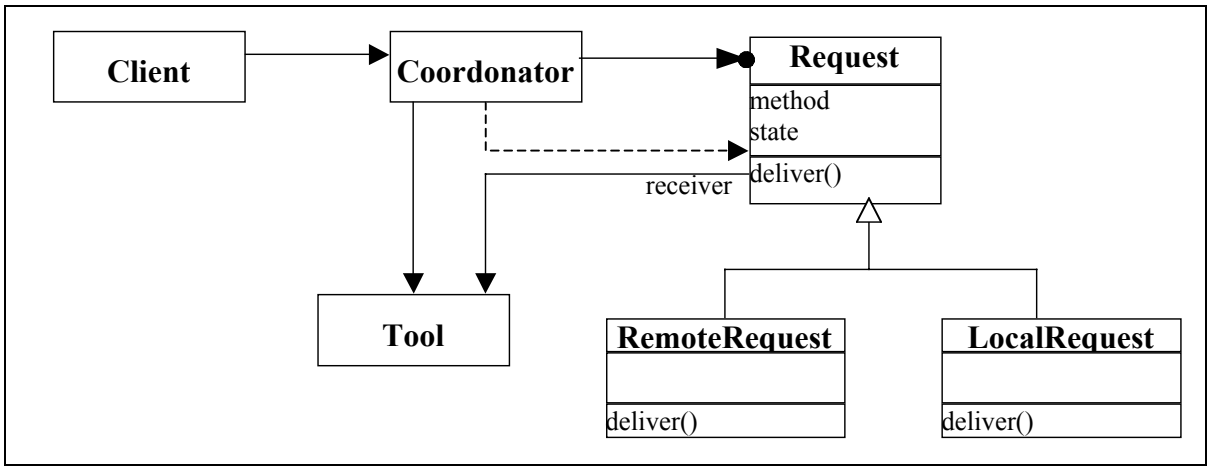


Figure 5. *Structure de l'interface abstrait d'appel d'outil*

Les objets participants

- Request : déclare l'interface d'appel. Il peut être spécialisé selon que l'outil est distant ou local. L'attribut *method* détermine le type d'action à effectuer. L'appel peut être synchrone, asynchrone ou par notification.
- Client : cette classe représente le service de l'interface utilisateur ou une activité prédéfinie.
- Serveur de Coordination (Coordinator) : il crée la requête et définit le destinataire.

- Tool : il a la capacité de traiter la demande contenue dans la requête.

Les collaborations

- Le client transmet les paramètres utiles au serveur de coordination.
- Le serveur de coordination vérifie la validité du destinataire et crée la requête.
- Le serveur de coordination émet la requête en appelant la méthode deliver().
- La requête est transmise au destinataire qui traite la demande.

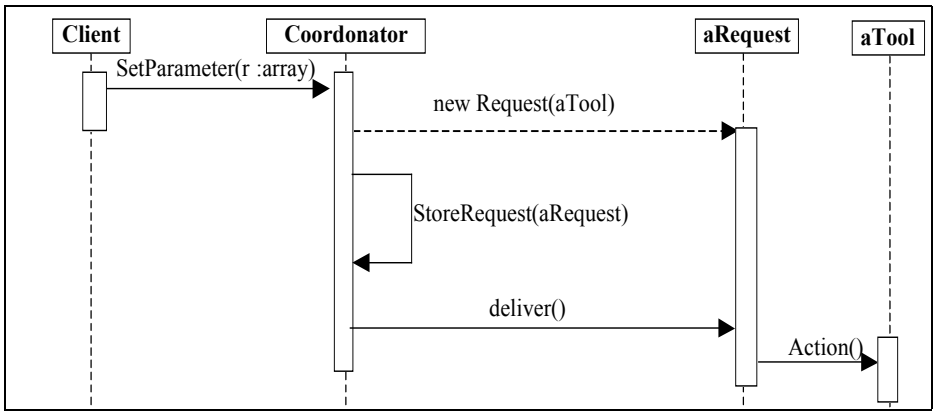


Figure 6. *Schéma des interactions*

3 L'ARCHITECTURE FONCTIONNELLE

Initialement, nous considérons qu'un site SIMES est constitué d'un Intranet composé de :

- Un serveur primaire qui dispose des données et des outils de traitement ; il est connecté directement à Internet et sert de passerelle entre l'intranet et le monde extérieur. Le système d'exploitation est supposé être Windows NT/2000, mais rien n'empêche d'utiliser tout autre système d'exploitation.
- Des serveurs secondaires et/ou des stations de travail qui disposent d'autres outils de traitement. Ils s'exécutent sur Linux, ou Windows NT/9X/2000 Pro.
- Des postes de travail (clients) qui ne disposent pas de composants utiles pour la plate-forme, mais qui permettent uniquement d'accéder aux services offerts par la plate-forme. Ils disposent d'un navigateur Web principalement.

Les machines serveurs se partagent des fichiers et répertoires par l'intermédiaire de NFS ou SAMBA. Cette considération est due tout simplement aux objectifs fixés initialement à savoir : préserver l'autonomie des applications et des données.

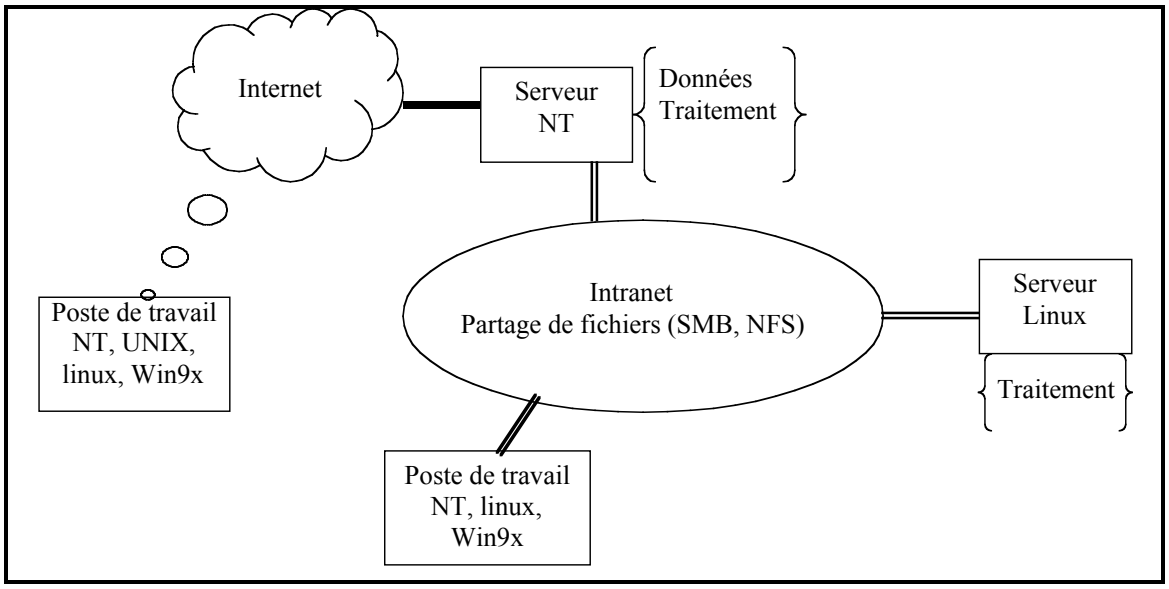


Figure 7. *Une vue spatiale d'un site serveur*

Une implémentation de la plate-forme est prévue pour être distribuée sur un réseau de type Internet, et accessible de partout. Du côté des sites serveur, les données sont :

- Soit des documents dans différents formats allant des formats propriétaires comme des documents Word (.doc), Framemaker (.doc, .fm) ; des feuilles de calculs comme Excel (.xls), des graphiques, des formats d'échange tels que .rtf.
- Soit des questionnaires dans des formats ad hoc.
- Soit des images dans différents formats tels que jpeg, gif, tiff,…
- Soit des bases de données.

Les outils sont distribués sur les sites serveurs et doivent être invocables de partout. L'interface utilisateur est simple et uniforme autant que possible. Pour satisfaire la généricité, l'ouverture et l'extensibilité de la plate-forme, son architecture s'appuie sur des standards émergents tels que HTML pour la description des échanges entre les postes clients et les sites serveur, SQL et ODBC pour l'interfaçage avec les bases de données et CORBA pour la distribution.

Pour tous ces besoins et pour la conception de l'architecture, nous avons tiré profit du projet PCIS2 principalement des recommandations touchant à la distribution et à l'organisation en couches. La plate-forme SIMES comprend les principales fonctionnalités suivantes :

- l'extraction des informations,
- l'intégration des outils,
- la gestion de l'hétérogénéité
- et des services de l'interface utilisateur.

C’est une architecture 3-tier où :

- L’interface utilisateur est complètement déployée chez le client sous forme de page Web. Elle est composée d’applications spécifiques et particulièrement d’un outil de navigation dans une base de documents.
- Les outils verticaux concernent l’administration des sites et la coordination de l’exécution, des outils de traitements et des outils spécifiques au domaine d’application tels que les extracteurs.
- La prise en compte de la distribution est basée principalement sur CORBA, une proposition de l’OMG qui s’impose comme un standard. Il permet d’accéder à tout objet installé sur un site équipé d’un ORB (Object Request Broker).

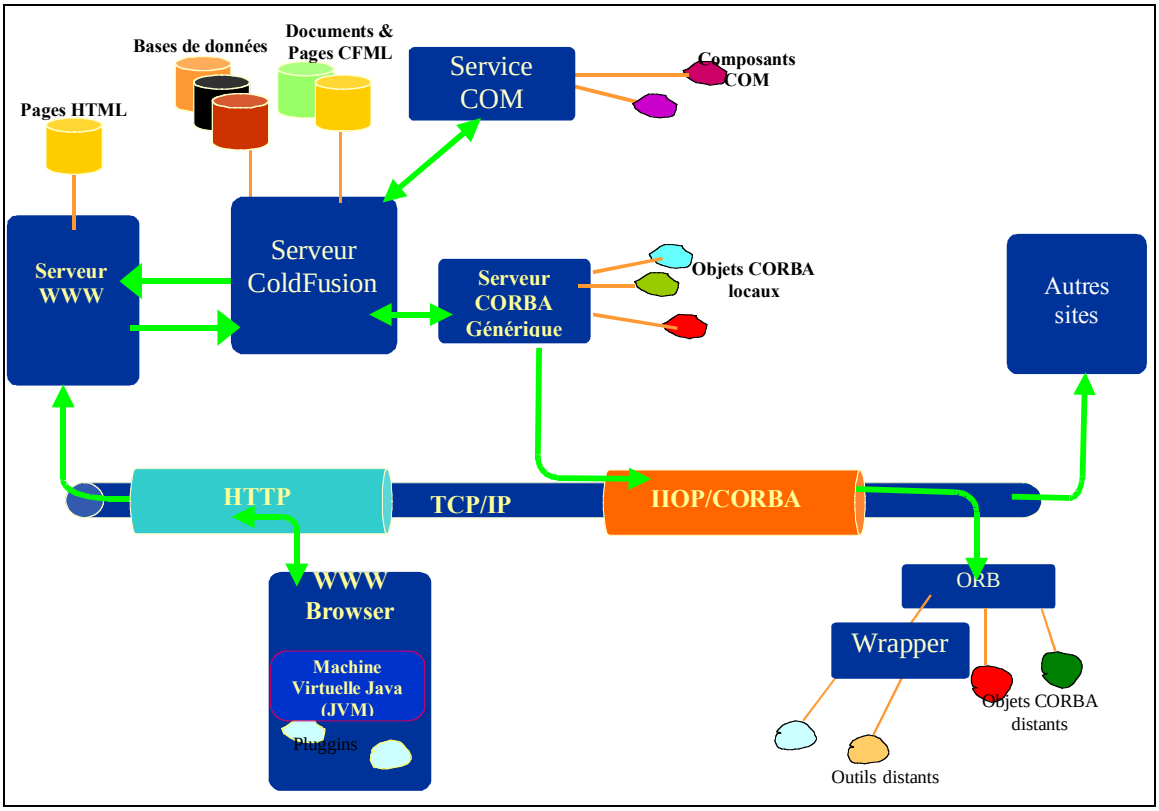


Figure 8. *Une vue topologique de l’architecture de SIMES*

Pour des raisons économiques, afin de faciliter la dissémination de la plate-forme SIMES, la mise en œuvre de l’architecture définie doit s’appuyer autant que possible sur des logiciels libres, l’achat des outils COTS (Commercial Off-The-Shelf) dans certains cas et l’utilisation de Java et C++ pour développer des fonctionnalités spécifiques.

Les services de l'interface utilisateur sont basés sur la technologie Web. Nous avons déjà étudié dans les chapitres précédents les différentes techniques de programmation qui permettent de développer des applications sur le Web. Du côté client, nous préconisons un navigateur Web comprenant une machine virtuelle Java, et des « pluggins » nécessaires pour afficher les différents types de documents manipulés. JavaScript est le principal langage utilisé pour accroître l'interactivité entre les formulaires et l'utilisateur.

Du côté serveur l'utilisation d'un firmware conduit à des solutions plus ouvertes, et complètement indépendantes des systèmes de base de données sous-jacents. En effet étendre les fonctionnalités d'une application pour y ajouter des supports Web, comme c'est le cas pour la plupart des SGBD tel que ORACLE à partir de sa version 8, est une approche certes efficace, mais propriétaire, et utilisable uniquement avec ORACLE. En adoptant une solution pareille, toutes les implémentations de SIMES devraient s'appuyer sur ORACLE, ce qui limite la dissémination. Après une étude des propositions commerciales nous avons opté pour ColdFusion[1]. Ce logiciel offre une interface entre des pages Web et des bases de données compatibles SQL ou ODBC. Il fonctionne suivant le modèle client-serveur : l'essentiel des traitements réside sur le serveur, et du côté client sont affichés uniquement des formulaires à remplir pour les requêtes.

ColdFusion présente des avantages par rapport à ses concurrents tels que ASP. Il est indépendant des bases de données et du système d'exploitation cible ; il est conçu pour fonctionner avec la plupart des serveurs web ; et l'utilisation des tags CFML est facile pour celui qui a une connaissance de HTML. De plus il présente des interfaces de connexion pour des systèmes utilisant COM, CORBA, C++ ou Java.

Pour maîtriser l'hétérogénéité et assurer la distribution des outils afin d'accéder aux outils distants installés sur d'autres sites, nous avons adopté CORBA. Nous avons évalué une implémentation en C++ : MICO, et une implémentation en Java : JacORB. L'interface ColdFusion-CORBA étant valable actuellement pour Visibroker uniquement, un serveur générique intégré à l'outil de coordination et d'exécution des outils de la plate-forme est indispensable. Cette composante permet d'accéder à l'ORB local et au delà à l'architecture CORBA, pour atteindre l'ORB distant et invoquer les outils distants.

4 INSTALLATION D'UN OUTIL SUR LA PLATE-FORME SIMES

La phase d'enregistrement consiste à fournir les informations permettant de définir la signature associée à chaque point d'entrée de l'outil, laquelle sera transmise à l'interface utilisateur pour la construction du formulaire associé à l'outil. Les informations qui décrivent un outil et ses points d'entrée sont constituées de :

[1] Produit de Allaire Corporation

- le nom externe tel qu'il sera présenté à l'utilisateur ;
- le nom interne qui sera utilisé par le service de coordination ;
- les paramètres (arguments et options) à transmettre lors de l'invocation afin que le serveur de coordination puisse construire la commande (si c'est un outil sous forme d'un programme invocable en ligne de commande du système d'exploitation cible) ou la requête (si c'est un objet CORBA) d'invocation de l'outil.
- d'autres informations complémentaires qui sont fournies en fonction de la structure de l'outil que nous pouvons intégrer à posteriori.

4.1.1 Structure d'un programme invocable en ligne de commande

Un programme invocable en ligne de commande suppose que nous disposons unique du fichier exécutable et des informations sur les formats en entrée et en sortie. Le nom de l'outil dans ce cas de figure désigne le chemin absolu d'accès au fichier binaire. Si de plus l'outil est installé sur une machine distincte de celle du serveur de coordination, l'utilisation d'un « wrapper » est indispensable.

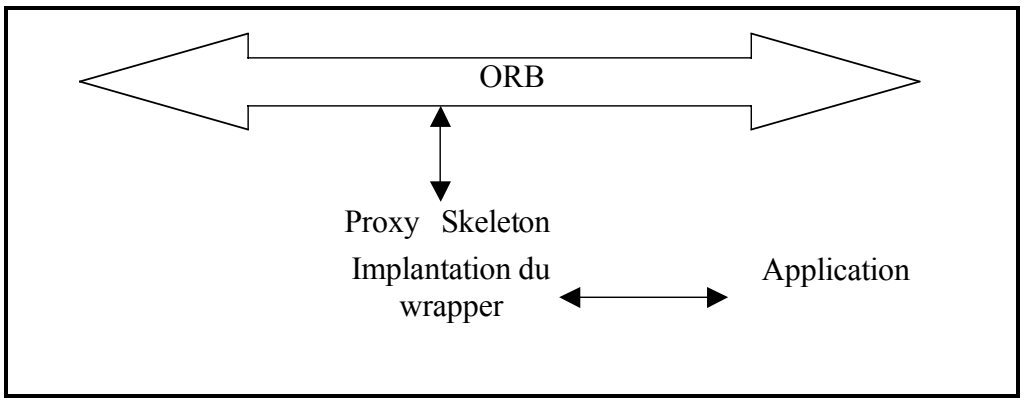


Figure 9. *Fonctionnement du « wrapper »*

Le wrapper assure la médiation entre l'environnement et l'outil : lorsque le service correspondant est sollicité au niveau de l'interface utilisateur, tous les paramètres nécessaires sont transmis au wrapper qui se charge de construire la commande appropriée. La communication entre le wrapper et l'outil s'effectue par les mécanismes offerts par le système d'exploitation cible.

4.1.2 Structure d'un objet CORBA

Pour un objet CORBA, on doit disposer au minimum d'un fichier contenant la référence d'objet (IOR) ou alors de son contexte dans le serveur de noms (Naming server). Avec ces informations, sa localisation est complètement transparente et relève de la compétence de l'ORB.

4.1.3 Structure d'un composant COM

Tout comme les objets CORBA, un composant COM dispose également d'un contexte qui détermine le type de mécanisme d'accès à mettre en oeuvre :

- In-Process : le client se connecte directement à une bibliothèque contenant le serveur et tous deux s'exécutent dans le même processus : celui du client. Ils partagent les ressources allouées aux processus client et la communication se fait à travers des appels de fonction. Cette façon de travailler fait alors des hypothèses sur le système d'exploitation et l'ensemble des logiciels disponibles.
- Serveurs locaux : le client accède à un serveur s'exécutant dans un processus différent mais sur la même machine.
- Serveurs distants : le client peut accéder à un serveur distant s'exécutant sur une autre machine.

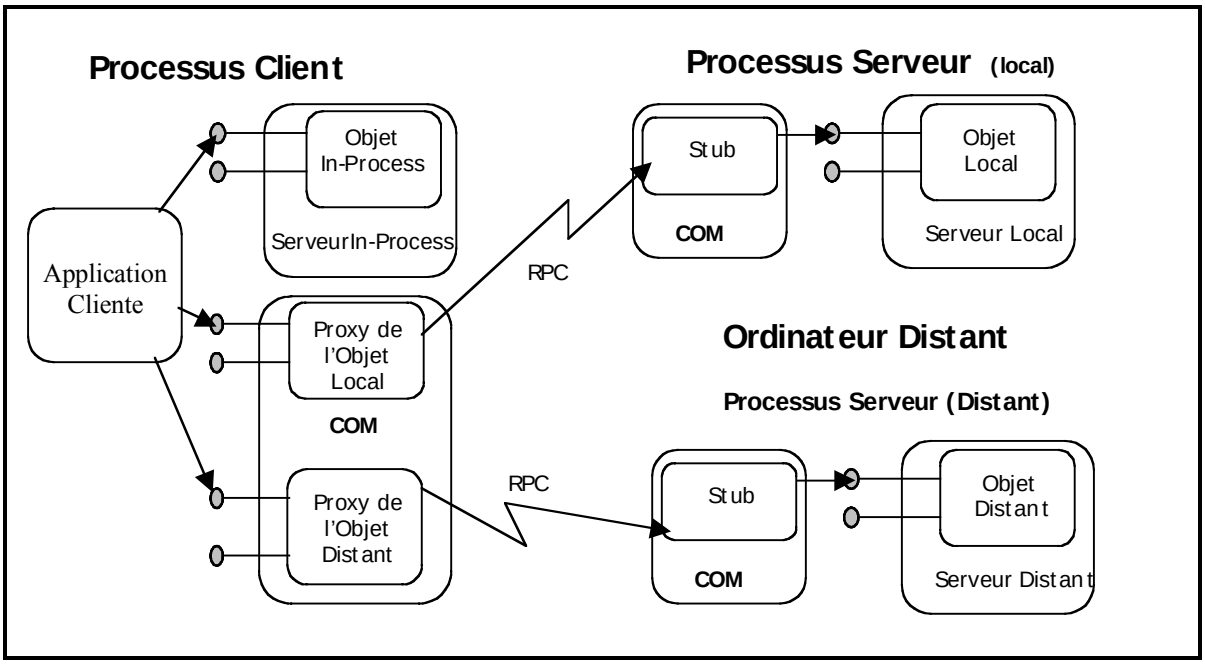


Figure 10. *Les méthodes d'accès aux objets COM*

4.1.4 Illustration : Megawave 2

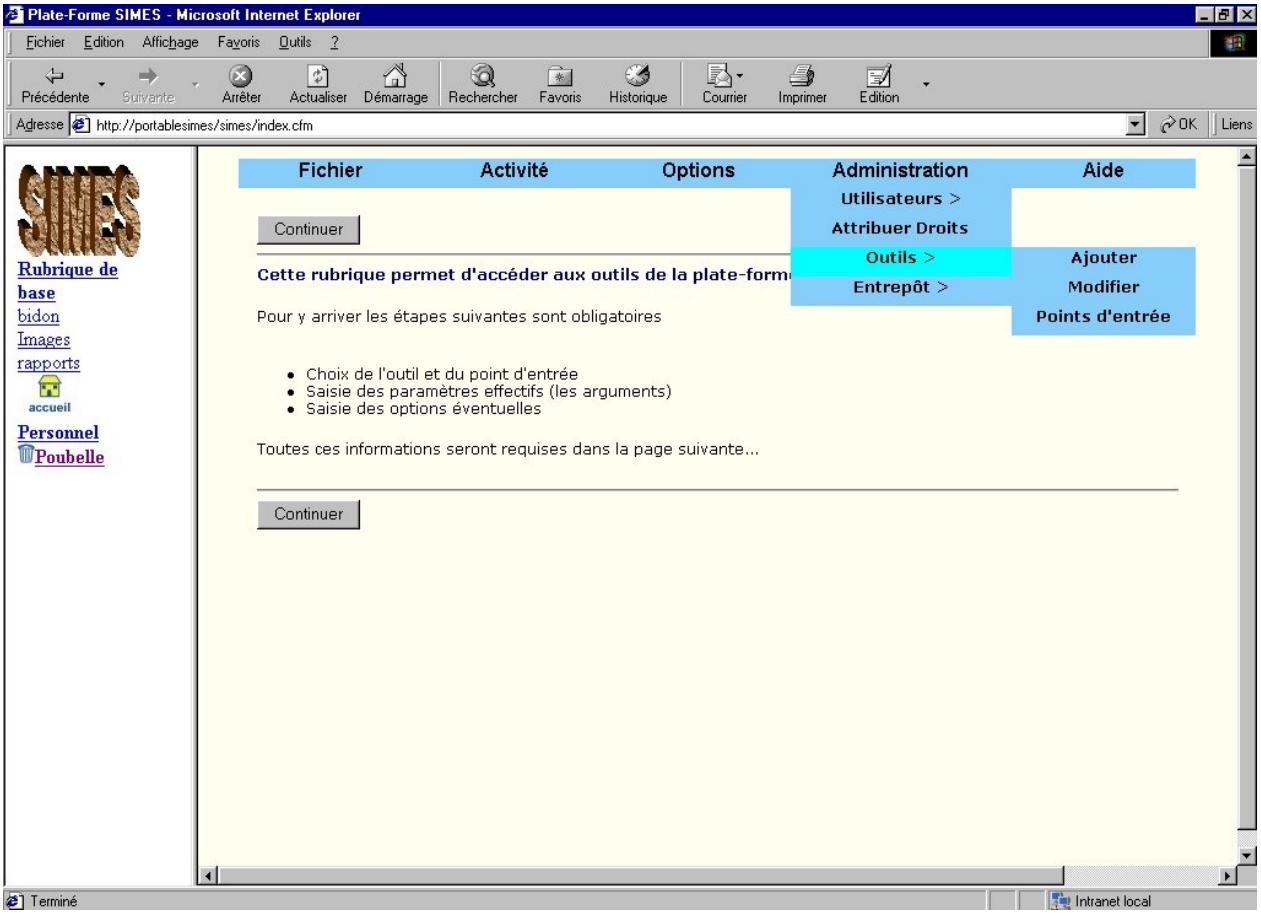
Parmi les outils disponibles actuellement figurent Mégawave2. Megawave 2 est un environnement logiciel conçu pour aider les chercheurs à écrire les algorithmes de traitement de signal et d'analyse d'images. Son intérêt réside dans la promotion des algorithmes les plus modernes de traitements d'images. Il est constitué d'un ensemble de fichiers binaires ayant chacun une fonctionnalité bien définie, que l'on peut invoquer en ligne de commande : raison pour laquelle nous le classons dans la catégorie des outils ligne de commande. C'est un outil qui s'exécute essentiellement sur des systèmes Unix et

compatibles tel que Linux. L'intérêt de cette action consiste à le mettre en ligne afin de pouvoir y accéder à partir de tout poste de travail connecté à Internet.

Au préalable, le fichier original doit être déposé sur le site SIMES. La procédure de dépôt est détaillée dans le « delivrable » D19.

Dans le menu administration, l'item Outil regroupe tous les points d'accès aux pages qui permettent de fournir les informations d'encapsulation d'un outil.

- Ajouter permet de décrire un nouvel outil.
- Modifier permet de mettre à jour la description d'un outil existant.
- Points d'entrée permet de fournir la description des points d'entrée associés à un outil.



Pour ajouter un outil :

1. Sélectionner Administration>Outils>Ajouter
2. Dans la page qui s'affiche, choisir la classe de l'outil, ou alors cliquer sur le bouton nouvelle classe définir sa classe n'est pas encore définie. Dans le second cas, il faut actualiser les informations de la page pour voir la nouvelle classe dans la liste déroulante.
3. Donner un nom à l'outil
4. Fournir sa description.
5. Sélectionner le type.
6. Sélectionner le contexte.
7. Fournir sa localisation (puisque nous sommes dans un environnement distribué)
8. Fournir le chemin.

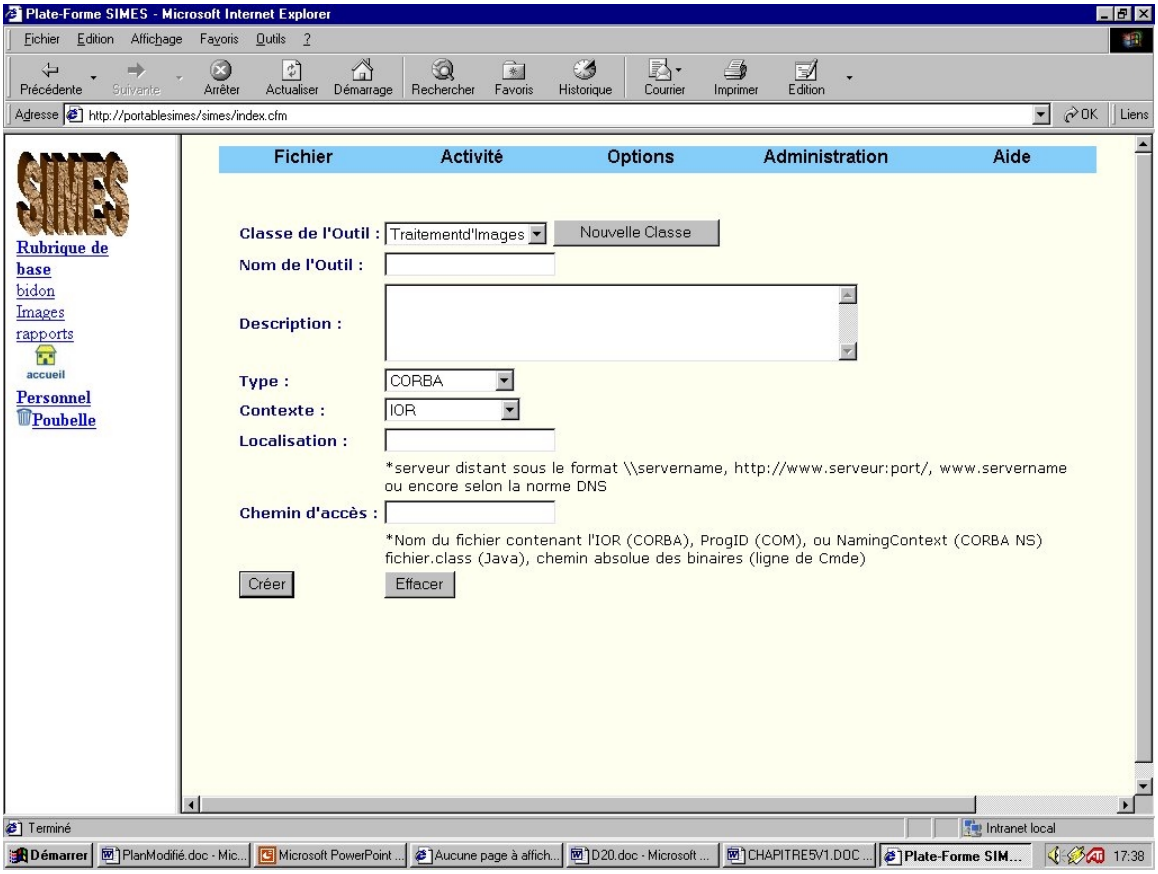


Figure 11. *Saisie des informations d'encapsulation d'un outil*

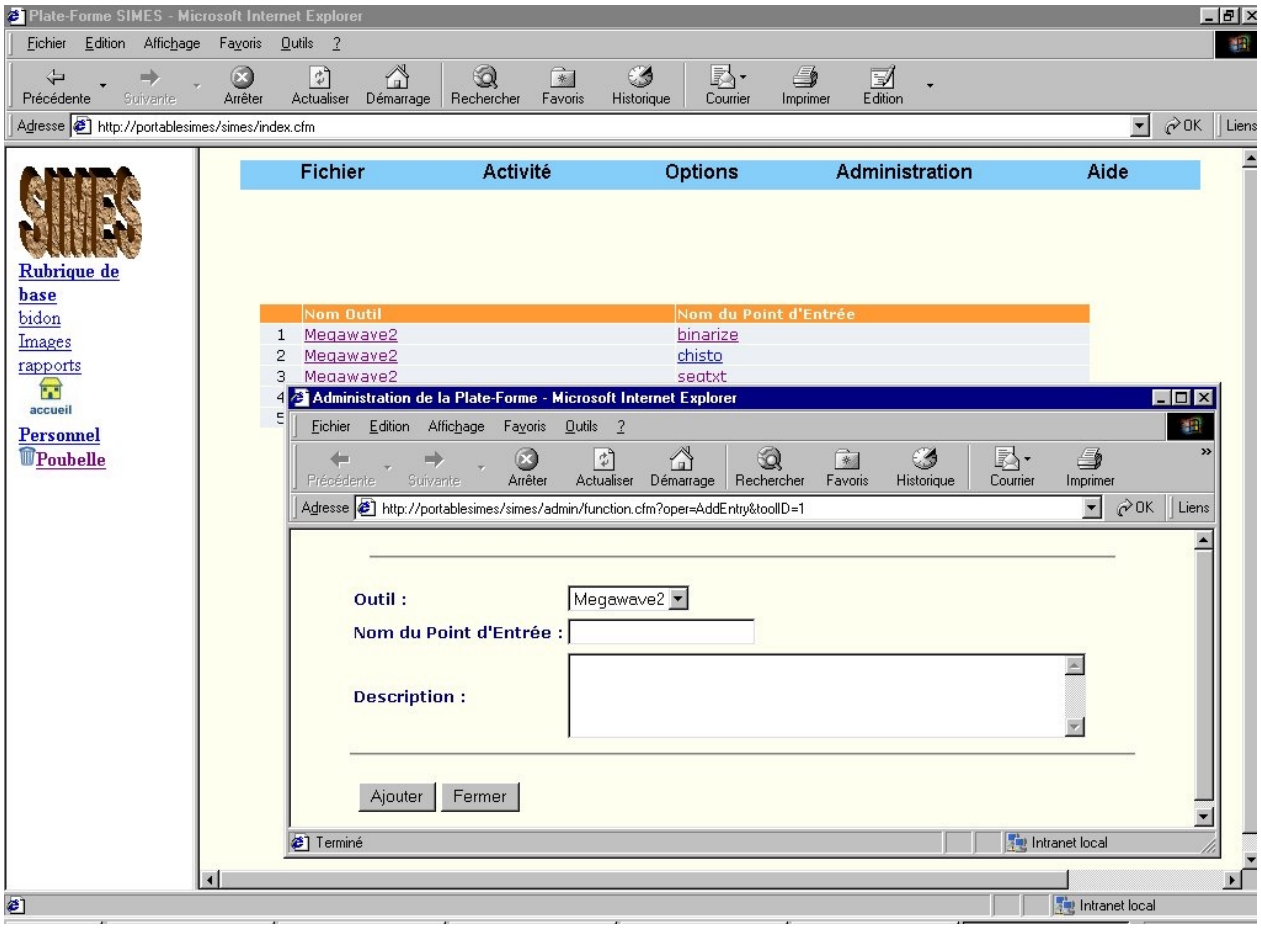
Les informations fournies par l'administrateur permettent de reconstituer les interfaces d'encapsulation de l'outil considéré. Les outils sont regroupés en classes en vue de définir une hiérarchie cohérente pour la visibilité de l'utilisateur. Selon le type d'outil, le contexte

est interprété différemment. Un outil « ligne de commande » n'a pas de contexte. Cependant sa localisation est obligatoire afin d'identifier le wrapper qui lui est associé. Le contexte d'un outil CORBA peut-être la référence de l'objet (IOR) ou le Naming Service (NS), pour un composant COM on distingue InProcess, Local et Remote.

Le chemin d'accès dépend étroitement du type de l'outil et de son contexte. Il désigne le chemin absolu des binaires pour un outil « ligne de commande », le fichier contenant l'IOR ou le Naming Context dans le NS pour un objet CORBA, le fichier .class pour Java et l'identifiant ProgID pou un composant COM.

Pour ajouter des « points d'entrée » :

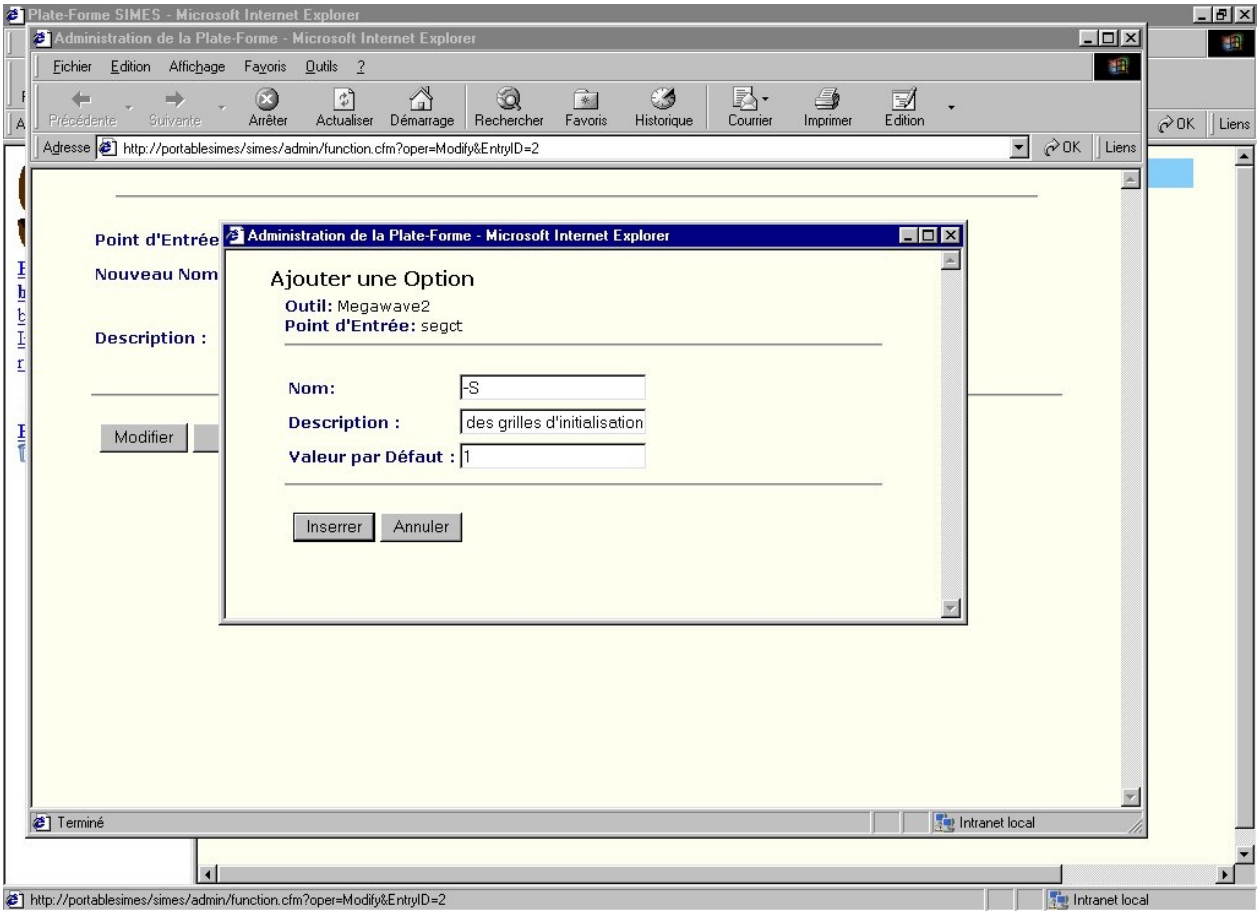
- Sélectionner Administration>Outils>Modifier
- Dans la page qui s'affiche figurent la liste des outils disponibles. Cliquer sur le nom de l'outil et une nouvelle page apparaît vous invitant à fournir :
    - Le nom du point d'entrée
    - Sa description



Une fois que le point d'entrée est défini, il faut lui associer des paramètres (options et arguments) :

1. Sélectionner Administration>Outils>Points d'entrée
2. Une page s'affiche avec la liste des points d'entrée et les outils auxquels ils sont associés. Cliquer sur le nom d'un point d'entrée pour le sélectionner.
3. Vous ensuite modifier sa description ou cliquer sur « *ajouter argument* » pour saisir ses argument et sur « *ajouter option* » pour saisir ses options.

Pour les options il faut fournir le nom, la description et les valeurs par défaut.



Le serveur de coordination récupère toutes ces données pour construire une commande interne que le wrapper transmet à l'outil à travers les services du Système d'exploitation cible.

5 UTILISATION

Pour accéder aux services qu'offre la plate-forme SIMES les utilisateurs doivent s'identifier. L'authentification de l'utilisateur marque le début de la session. A l'ouverture de session, la liste des services accessibles est établie en fonction du rôle affecté à l'utilisateur.

5.1 PRESENTATION DE LA SESSION

Une session désigne toutes les connexions à la plate-forme SIMES pouvant être effectuées par un utilisateur unique au cours de l'exploration des services disponibles. Cette vue logique d'une session commence à la première connexion établie par un utilisateur et se termine par une déconnexion normale ou anormale (après un délai d'attente spécifié). Cette vision est proche de celle des environnements multi-utilisateurs où une session se termine lorsque l'utilisateur a fini d'employer une application et la quitte pour faire autre chose.

Mais, étant donné la nature sans état du Web, il n'est pas toujours possible de définir un point précis auquel une session se termine. Dans la plupart des cas, cependant, une application Web n'a aucun moyen de savoir si l'utilisateur a fini ou s'attarde simplement sur une page.

Pour pallier cette difficulté, on peut spécifier un délai d'attente. Si l'utilisateur n'accède à aucune page de l'application durant ce délai d'attente, le serveur SIMES interprète ceci comme une suspension de la session et l'état de la session est maintenu persistant dans le référentiel (repository) pour une reprise ultérieure. Autrement l'utilisateur peut mettre volontairement fin à la session en activant le menu correspondant, dans ce cas le contexte de la session (toutes les variables associées à la session) est supprimé.

5.2 L'INVOCATION D'UN OUTIL

L'utilisation des outils de la plate-forme SIMES se fait par l'intermédiaire du menu « Activité ». On retrouve dans cette rubrique les items tels que :

- Lancer un outil : cet item permet d'accéder à l'interface générique d'invocation des outils.
- Démarrer une activité prédéfinie.
- Et une liste des services spécifiques comme l'exploration des entrepôts de la plate-forme, l'accès à des entrepôts connexes (Herbier National de Cameroun, Observatoire Péri-urbain de Ngaoundéré au Cameroun, etc.).

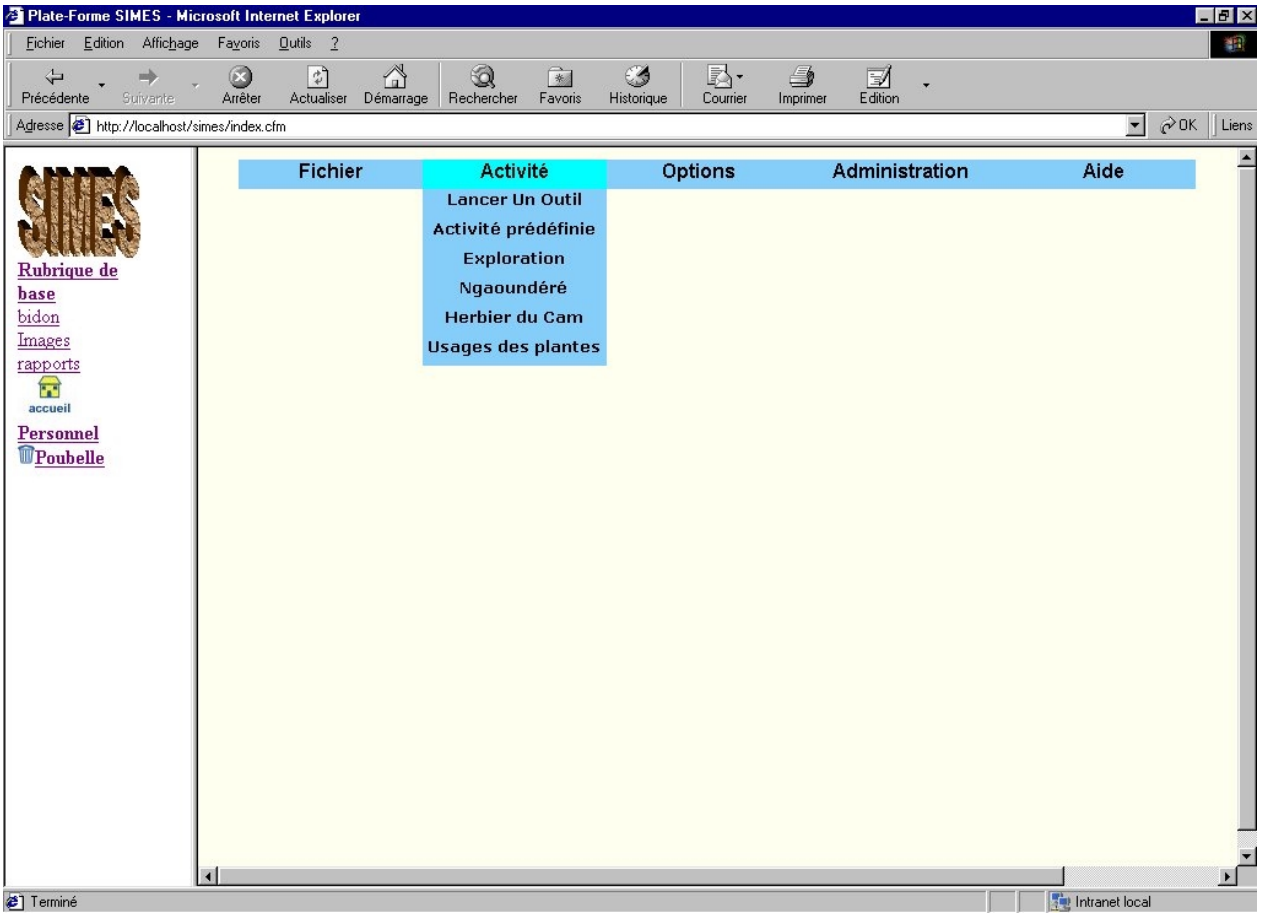


Figure 12. *Les Items de la rubrique activité*

Pour accéder à la page d'invocation d'un outil :

- Sélectionner Activité>Lancer un outil.

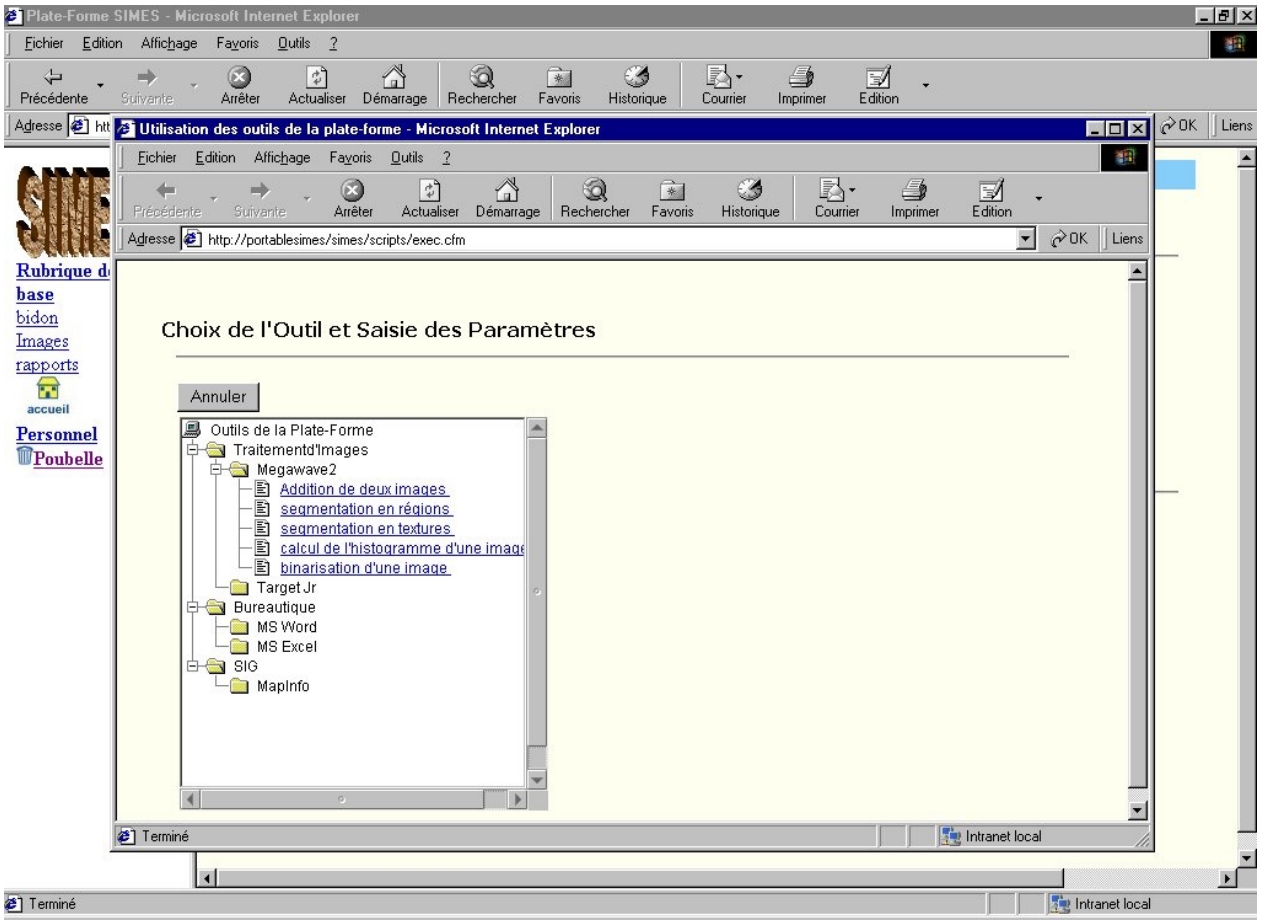


Figure 13. *Choix de l'outil à invoquer*

1. Dans la page qui apparaît, dérouler la hiérarchie et cliquer sur un point d'entrée pour le sélectionner.
2. Saisir les paramètres effectifs. Cliquer sur Exécuter pour lancer la requête
3. Construction de la requête et invocation de l'outil. Cette étape est dédiée au serveur de coordination. Elle est donc complètement transparente pour l'utilisateur.

La requête ainsi construite par le serveur de coordination est envoyé au wrapper qui réside sur le serveur de traitement qui héberge les binaires de Mégawave2. La localisation du wrapper et l'interaction entre lui et le serveur de coordination est assuré par l'ORB. Le rôle principal du wrapper est de traduire cette requête en une « *ligne de commande* » qui sera transmise à l'outil à travers les services du système d'exploitation cible.

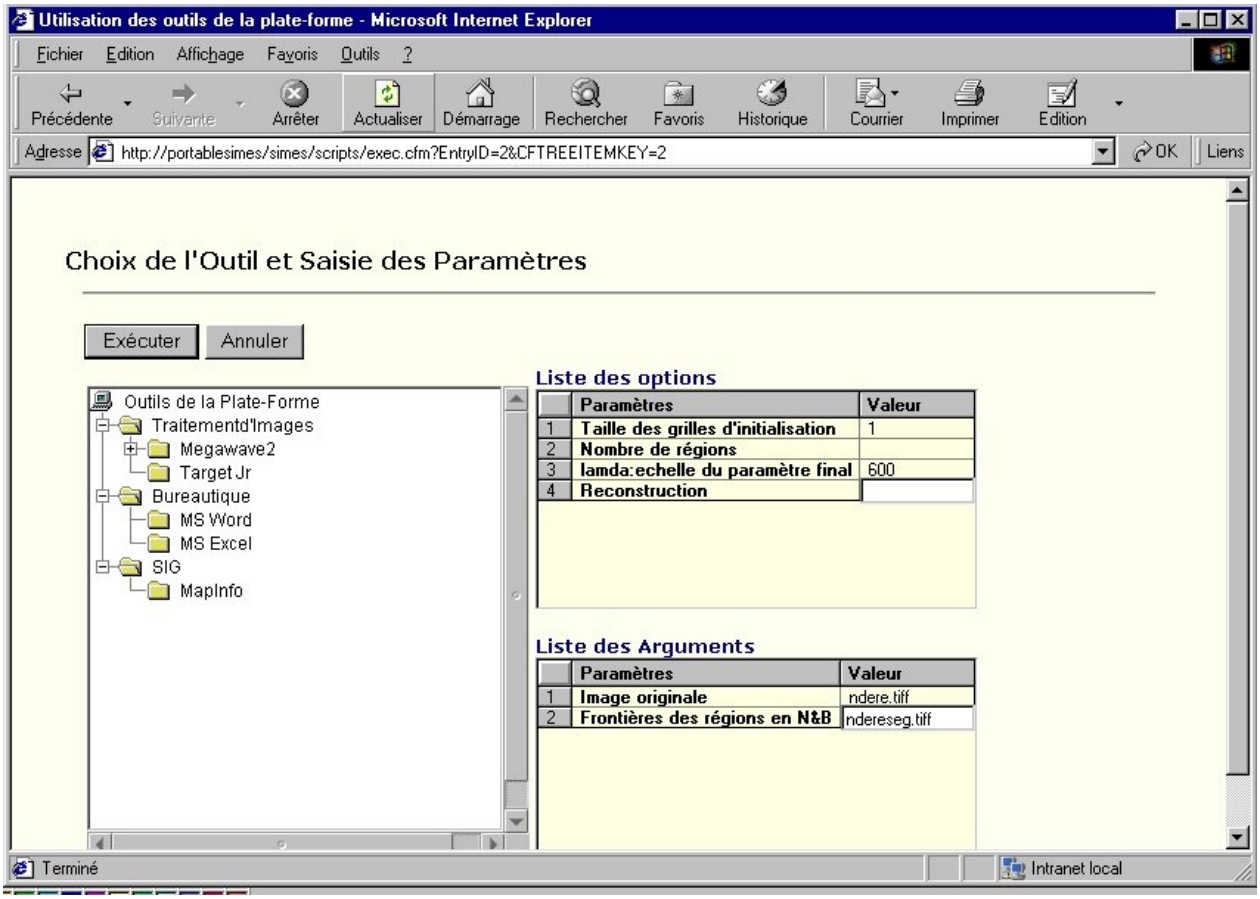


Figure 14. *Saisie des paramètres effectifs*

6 CONCLUSION

Dans sa version actuelle le système SIMES est un prototype de plate-forme logicielle intégrée qui tire profit des différentes solutions proposées aux problèmes d'intégration d'outils logiciels dans la construction des Ateliers de Génie Logiciel. Sur le plan technique, la maîtrise de l'hétérogénéité due aux langages de programmation, aux systèmes d'exploitation et aux réseaux de communication sous-jacents a conduit à la définition des architectures de référence.

Ce sont des infrastructures fondées sur le modèle client-serveur, qui offrent des interfaces d'accès à des services locaux ou distants pour les applications en garantissant l'interopérabilité, la portabilité, la flexibilité et la transparence à l'hétérogénéité. Cette solution communément appelée middleware est basée sur des concepts relativement simples (mais pas toujours simples à mettre en œuvre).

Pour satisfaire complètement l'homogénéité du service d'accès aux données et aux outils vu de l'utilisateur, le système SIMES offre une vision unifiée et une apparence commune d'accès aux données et aux outils logiciels de l'environnement. Cette propriété relève des systèmes de gestion de l'interface utilisateur. Les standards comme Motif ou Xwindows jadis utilisés dans les projets d'intégration cède progressivement la place aux navigateurs Web. L'utilisation de ColdFusion nous permet de prototyper rapidement la plate-forme tout en respectant l'indépendance vis à vis des bases de données sous-jacentes.