MASTER EN SCIENCES INFORMATIQUES

Utilisation de la technologie XML et LDAP dans la gestion de configurations d'agents

Verenne, Alain

Award date:
2003

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'informatique

Utilisation de la technologie XML et LDAP dans la gestion de configurations d'agents

Alain Verenne

Mémoire présenté en vue de l'obtention du grade

de

Maître en informatique

Année académique 2002-2003

Résumé

Depuis quelques années on voit émerger de plus en plus la technologie XML. Les domaines d'application et les outils se multiplient et se développent rapidement. Nous allons étudier ici comment construire des feuilles de configuration XML pour trois des agents d'OpenMaster, la plateforme de gestion de systèmes informatiques développés par la société française Evidian S.A., filliale du groupe BULL. Depuis les DTD jusqu'aux schémas XML, nous analyserons les avantages et inconvénients de ces deux technologies comme moyen de validation de documents XML, mais nous donnerons aussi quelques explications techniques sur les trois parseurs les plus couramment rencontrés et que nous serons amenés à utiliser : Expat, SAX et DOM.
Dans un second temps, nous essaierons de donner une ébauche de réponse de centralisation de ces configurations dans un annuaire LDAP. Ce mémoire abordera aussi un aspect plus pratique sur le façon d'implémenter ces concepts en C++, Perl et Java ; en attirant spécialement l'attention sur le problème de portabilité des différentes librairies requises pour manipuler les feuilles XML dans ces langages. Enfin nous terminerons par donner quelques suggestions concernant la mise en oeuvre d'un logiciel de gestion avec une application conçue dans le but de réaliser des projets basés sur la technologie Internet : Zope.

Abstract

For a few years, we have more and more seen the emergence of the XML technology. The application fields and the tools have been on the increase and have extended quickly. We are going to look here how to build XML configuration sheets for three agents of OpenMaster, the computer network system manager platform developed by the french company Evidian S.A., a subsidiary company of Bull group. From DTD to XML schemas, we will analyse the advantages and the inconvenients of both of these technologies like a way to validate XML documents, but we will give you also some technical explanations about the most encountered three parsers we shall use : Expat, SAX and DOM.
In a second time, we will try to give you a drafted answer about the way to centralize these configurations in a LDAP directory service. This thesis will tackle also a more practical angle about the way to implement these concepts in C++, Perl and Java ; in attracting specially attention on portability problems of the different libraries needed for manipulating XML sheets in these languages. Finally we will end in giving some suggestions concerning the design of a manager software with a application created in order to realize projects based on Internet technology : Zope.

Avant-propos

La conception d'un mémoire est un moment important dans la vie d'un étudiant. Il est l'aboutissement d'une formation et le reflet d'une connaissance et d'un savoir-faire. Acquis pendant trois années de maitrise informatique aux FUNDP, ce savoir-faire se doit d'être valorisé au mieux, d'où l'importance de bénéficier du soutien de personnes compétentes qui nous guident et nous conseillent.

Ainsi je voudrais profiter de l'occasion pour remercier Madame Françoise Sénéchal-Castan, Olivier Donzé, Hugues Devarax, Philippe Terrier, François Urbain, ainsi que les autres membres de l'équipe dont je n'ai pu me souvenir du nom et qui, je l'espère me pardonneront de cet oubli, pour leur accueil et leur soutien au sein de la société Evidian.

Je voudrais également remercier mon promoteur Monsieur Jean Ramaekers, professeur à l'institut d'informatique de Namur, pour les conseils précieux qu'il a pu m'apporter lors de la rédaction de ce travail.

Enfin, je terminerai par remercier les membres de ma famille et les proches qui m'ont donné la force nécessaire de surmonter tous les obstacles que j'ai rencontrés avant d'achever ma formation d'informaticien.

Table des matières

Table des figures

Glossaire

A

Agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu. [Wooldridge et al., 1998]

Agent COACH est un agent qui a pour but de fournir un moyen de distribuer les services d'administration OpenMaster à travers de grands réseaux hétérogènes.

Agent d'extension est un agent utilisé dans le but de retrouver des données dans une MIB sans pour autant devoir développer un agent complet.

Agent de contrôle est un agent qui gère plusieurs types de ressources nommées "alarm ressources". Lorsque cet agent détecte une anomalie, il a la possibilité d'envoyer un signal au serveur OpenMaster et d'effectuer des opérations correctives.

AIX est le système d'exploitation Unix de la société IBM.

Analyse syntaxique est une opération durant laquelle on vérifié la conformité d'une succession de lexèmes par rapport à une grammaire donnée. Elle suit l'analyse lexicale.

Annuaire électronique est un type de base de données spécialisé permettant de stocker des informations de manière hiérarchique en offrant des mécanismes simples pour rechercher l'information, la trier et l'organiser selon un nombre limité de critères.

API ou "Application Programming Interface" est une librairie contenant un ensemble de fonctions courantes de bas niveau bien documentées, permettant de programmer des applications de haut niveau. On obtient ainsi des bibliothèques de routines stockées par exemple dans des DLL.

C

C++ langage qui fut créer par le Dr. Bjarne Stroustrup de Bell Laboratories au début des années 1980. Le but était de créer un langage permettant la programmation orientée objets tout en restant hautement performant. Aujourd'hui, les langages C et C++ sont deux des langages les plus utilisés de la planète, et tirent leur force de leur flexibilité, leur performance et surtout, de leur immense popularité.

COACH nom diminutif de l'agent COACH.

CSS ou "Cascading Styles Sheets" est la norme du W3C définissant l'utilisation de feuilles de styles pour l'affichage des pages HTML.

D

"Directory Information tree" ou DIT est la structure arborescence hiérarchique des données dans LDAP.

"Directory service" un annuaire LDAP contient un ensemble d'objets. Le "directory service" est un service qui fournit les éléments essentiels pour créer, ajouter, supprimer ou modifier les attributs associés aux objets de cet annuaire.

le "Distinguish Name" chaque entrée est identifiée dans le directory Information Tree par son DN(distinguish Name). Celui-ci représente le nom de l'entrée sous forme d'un chemin d'accès depuis le sommet de l'arbre (c'est le chemin absolu depuis la racine).

DOM ou Document Object Model est le résultat d'une recommandation du consortium W3C. C'est un parseur qui construit un arbre structuré en mémoire à partir du document qu'il parse, et ce afin de traiter les informations.

DTD ou "Description Type Document" sont des SGL définies avec SGML. Associé à des documents XML, les DTD leur imposent des contraintes de structures.

DTML langage de script orienté serveur, basé sur les langages python et HTML. Il a été crée pour écrire de pages web dynamiques.

E

Espace de nommages est un mécanisme qui permet de stocker sur une même feuille XML des informations à destination de plusieurs applications sans obliger l'utilisateur à changer de vocabulaire pour chaque application. Les noms des espaces de nommage XML peuvent avoir la forme de noms qualifiés, qui contiennent un caractère deux-points séparant le nom en un préfixe d'espace de nommage et une partie locale.

Expat n'est pas un parseur en lui-même. C'est une librairie écrite en C par James Clark qui permet de créer ses propres parseurs XML.

Expressions régulières sont une famille de notations compactes et puissantes pour décrire certains ensembles de chaîne de caractères. Ces notations sont utilisées par plusieurs éditeurs de texte et utilitaires comme "Emacs" pour parcourir de façon automatique des textes à la recherche de morceaux de chaînes de caractères ayant certaines formes.

F

Framework est un bloc software sophistiqué qui gère les signaux envoyés par les agents.

G

GCC est la version libre du compilateur C/C++ "cc" de Solaris.

H

Handler ici, c'est le gestionnaire d'événements d'un parseur SAX. Il indique la procédure à suivre pour chaque événement survenant lors du parsing du document.

HTML ou "HyperText Markup Langage" est un langage de programmation balisé basé sur le langage SGML permettant d'écrire des documents destinés à être consultés sur internet.

HP-UX est le système d'exploitation Unix de la société Hewlett-Packard

J

Java langage qui a été développé par SUN Microsystems dans les années 1990. A l'origine on étudiait la possibilité de concevoir un projet d'environnement indépendant du hardware pouvant facilement permettre la programmation d'appareils aussi variés que les téléviseurs, les magnétoscopes,... Après l'échec de 1992, la technologie Java allait pouvoir renaître grâce au web où les principaux problèmes rencontrés à cet époque étaient liés à l'hétérogénéité des machines et des logiciels utilisés. En 1995, Java est soumit gratuitement à la communauté Internet sous forme d'une machine virtuelle composée d'un compilateur ainsi que de nombreuses spécifications. Aujourd'hui de plus en plus de sociétés (ou de particuliers) utilisent ce langage pour leurs développements.

Java station OpenMaster est un composant optionnel permettant d'administrer OpenMaster à partir d'une machine non dédiée. Cette technologie,basée sur un serveur web, offre comme gros avantage de permettre à plusieurs personnes d'accéder aux mêmes serveurs en même temps.

JNDI Java dispose d'une interface capable d'interagir avec des services comme LDAP, RMI ou encore Corba. Cet interface porte le nom de "Java Naming and Directory Interface" ou JNDI. Ces API fournissent des fonctionnalités permettant de gérer des services "naming" et "directory" en ne se servant que du langage de programmation Java.

L

LDAP ou "Lightweight Directory Access Protocol" vient d'un protocole basé sur un service d'annuaire X500. Ces serveurs sont conçus pour pour stocker une grande quantité d'information de faible volume dans un modèle hiérarchique.

LDIF ou "LDAP Data Interchange Format" permet de représenter les données LDAP dans un format texte standardisé. Ce fichier peut être utilisé pour ajouter ou modifier des données dans la base

Librairies dynamiques dans une librairie compilée en mode dynamique, on reporte le chargement en mémoire du code de la librairie lors de l'exécution du programme. En gros cela revient, à la compilation, à remplacer l'édition de liens statique par l'ajout de code et d'informations suffisants pour aller chercher les symboles non encore résolus.

Librairies statiques dans une librairie compilée en mode statique, le code des fonctions de la librairie utilisée dans un programme est directement intégré dans le code exécutable.

Licence GPL ou "Public General License" est une licence qui concerne les logiciels "libres". Elle donne à tout utilisateur le droit de partager ou de modifier un logiciel protégé par cette licence. En outre elle impose une cession intégrale de tous les droits qu'elle octroie 'a tout nouvel acquéreur d'un logiciel libre, et ce sans aucune restriction. Cette licence a été crée par la "Free Software Fondation" en 1989.

Linux est un système d'exploitation "libre" basé sur Unix. Il fut crée en 1991 par le Finlandais Linus Torvalds.

M

Machine virtuelle est une machine abstraite simulée au sein d'une autre machine bien réelle, qui utilise comme environnement d'exécution un langage portable de haut niveau. Exemple le plus connu : JVM ou "Java Virtual Machine".

"MakeC++SharedLib" ce script permet de créer des librairies sous AIX avec le compilateur gcc.

MIB ou "Management Information base" est la base de données des informations de gestion maintenue par l'agent, auprès de laquelle le manager va venir pour s'informer. Cette base de données d'informations de gestion est constituée d'objets qui représentent des variables. Deux MIB publics ont été normalisées : MIB I et MIB II (dite 1 et 2). Elles décrivent l'ensemble des variables TCP/IP.

N

"Naming service" de LDAP permet d'associer des noms aux objets afin d'y accéder via ce nom plutôt que par leur OID (object Identifier). Ce service est un peu analogue à celui du DNS pour les adresses IP.

O

OAC acronyme de l'agent de contrôle (Open Agent Control).

OAX acronyme de l'agent d'extension (Open Agent eXtension).

"Object Identifier" ou OID sont des adresses numériques qui permettent d'identifier de manière unique un objet partagé sur le réseau. Comme dans le cas des adresses IP, un organisme se charge d'attribuer ces adresses afin d'éviter tout conflit. En outre, leur format est assez analogue à celui utilisé par le protocole IP.

OpenMaster regroupe un ensemble d'applications permettant l'administration de ressources aussi diverses que des réseaux, des applications ou des utilisateurs système. Actuellement, cet ensemble fonctionne sous des environnements UNIX, Windows NT et Windows 2000.

P

Parser mot anglais pour parseur.

Parseur analyseur syntaxique.

Parsing mot anglais pour analyse syntaxique.

Pattern matching procédure qui consiste en une mise en correspondance, mot par mot, caractère par caractère, de la réponse de l'utilisateur et du 'pattern' spécifié par le concepteur.

Perl ou Practical Extraction and Report Language fut crée en 1986 par Larry Wall. Ce langage se caractérise surtout par le fait qu'il est interprété, portable, robuste, gratuit et simple. A l'origine Perl était destiné à la manipulation de fichier, de texte et de processus dans un environnement système UNIX.

R

"Related Distinguish Name" ou "RDN" est le nom d'un objet LDAP concaténé avec le nom de ses ancêtres (c'est le chemin absolu en considérant que le noeud courant est la racine de l'arbre).

Relax NG Relax NG est un encore un autre langage développé dans le but de concevoir des schémas XML. Il a été conçu par le comité technique RELAX NG d'OASIS de façon à être une alternative au fameux schéma XML du W3C.

S

SAX ou "Simple API for XML" est une API basée sur le modèle événementiel.

Schéma LDAP sont des fichiers qui contiennent les déclarations relatives aux objets (objectclass) et aux attributs utilisés dans ces objets. Ces deux types d'éléments sont caractérisés par une série de propriétés.

Schéma XML du W3C est un langage spécifié par le W3C permettant de développer des schémas XML.

Schematron est un langage de schéma développé par Rick Jelliffe et qui est basé sur le paradigme du "tree pattern".

"Segmentation fault error" erreur qui traduit le fait qu'un programme a tenté d'accéder à une adresse mémoire inaccessible ou inexistante.

Service de réplication de LDAP consiste à recopier tout ou en partie le contenu d'un arbre sur un ou plusieurs serveurs afin de notamment, rapprocher le service des utilisateurs, de faire du load balancing, d'importer des entrées générées sur un autre serveur, de pallier aux pannes et aux coupures du réseau.

SGML ou "Standard Generalized Markup Language" fut le premier langage de formats normalisés de données structurées reconnus par l'ISO en 1986. SGML peut être utilisé pour définir des structures logiques génériques(SLG).

SGL ou "structures logiques génériques" sont représentées par un ensemble de balises qui correspondent à un type particulier de documents, et qui permet d'en décrire la structure logique. Les SGL définies avec SGML sont appelées DTD.

SNMP ou "Simple Network Management Protocol" est un protocole habituellement employé sur l'Internet pour gérer les noeuds d'un réseau IP. Il décrit la manière dont sont véhiculées les données entre une station et un agent.

SOAP ou " Simple Object Access Protocol" est un nouveau protocole proposé par Microsoft à l' IETF dans le cadre de son nouveau modèle d'utilisation de l'informatique ".Net". Sa syntaxe d'utilisation est fondée sur XML et ses commandes sont envoyées sur Internet par l'intermédiaire du protocole HTTP. Il joue un rôle analogue à Corba, puisqu'il permet aux systèmes objets distribués de solliciter et d'obtenir des services rendus par d'autres objets. Il a cependant le grand avantage d'être nettement moins lourd à mettre en oeuvre.

Solaris est le système d'exploitation Unix de la société Sun Microsystem.

Station OpenMaster sont munies d'une série d'applications permettant d'obtenir des informations sur les objets gérés.

Stream-oriented parser parseur auquel on va associer des fonctions de "callback" (handling functions) qui seront alimentées lors du traitement du document. Lors qu'un événement sera reconnu par le parseur, il fera appel à la fonction de callback affectée à cet événement pour exécuter le traitement adéquat.

Swap la zone de "swap" est la zone d'échange entre la mémoire centrale et un espace du disque dur réservé à cet effet. Cet espace peut être un fichier ou une partition logique.

T

Toolbox est un mot anglais qui se traduit par "boite à outils". En informatique, une boite à outils est un ensemble de petits logiciels utilitaires.

"Trap" mot anglais qui signifie, dans le cadre particulier du protocole SNMP, "signal envoyé par un agent à destination du manager du système afin de l'avertir de la survenance d'un événement".

Trex ou "Tree Regular Expressions for XML" est un autre langage de conception de schémas XML.

V

Validation d'un document consiste à vérifier si un document est conforme à la structure d'une DTD ou d'un schéma XML. La validation se charge également de veiller à ce que le document soit bien formé (well-formed).

W

W3C ou "World Wide Web Consortium" est un organisme de promotion du web crée en 1994 sous l'impulsion de Tim Berners Lee. Son principal objectif est la mise au point de normes et de protocoles ouverts et libres, dans un souci d'interopérabilité maximale : RDF, XML. Il est géré conjointement par le MIT aux Etats-unis, par l'INRIA en France et par l'université Keio au Japon.

Windows NT est le système d'exploitation de la société Microsoft.

X

XML ou "eXtensible Markup Langage" a été développé par un groupe de travail XML (initialement connu comme étant le comité d'examen éditorial SGML) sous la direction du consortium W3C. Il décrit une classe d'objets de données appelés documents XML. Ce

langage de balisage extensible structure des données dans un document à la manière du HTML.

XPATH ou "XML Path Language" est un langage permettant d'adresser les différentes parties d'un document XML.

XSL ou "eXtensible Stylesheet Language" est un langage permettant de définir des feuilles de style, de manière analogue à "CSS" pour HTML. Il se compose d'ailleurs d'une part d'un langage de transformation de document XML(XSLT), et d'autre part d'un langage de description de sémantique de formatage.

XSLT ou "Extensible Style Language Transformation" est un langage sous-ensemble de XSL, qui permet d'effectuer des traitements et des transformations sur les données XML. Le document produit peut l'être dans différents langages (HTML, XHTML, WML, ...)

Z

Zope ou "Z Object Publishing Environment" est une plateforme de développement de projets web.

1 Introduction

Il n'est pas rare d'avoir un même logiciel présent sur chaque machine d'un réseau local. Parfois les conditions d'utilisation imposeront une configuration adaptée en fonction des besoins spécifiques d'une machine donnée, dans d'autres cas elles impliqueront une même configuration sur toutes les machines. Dans cette seconde situation, une distribution efficace des configurations devient alors un impératif pour les administrateurs de ce réseau. Mais une distribution efficace passe généralement par une centralisation qui donne l'assurance que chaque modification effectuée sera prise en compte partout où le fichier sera utilisé.
Mais qui dit modification dit également éditeur de texte. En effet, pour modifier un fichier, il faut pouvoir travailler sur un texte de contenu lisible et compréhensible par un utilisateur. A ce niveau tout le problème vient des fichiers encodés dans certains formats très particuliers, comme celui de word par exemple, où sans le logiciel adéquat, il est impossible de comprendre la signification de ce contenu. Aussi, afin de rendre ces fichiers accessibles via un simple traitement de texte et facilement transférables sur le réseau, on peut avoir recours à certaines technologies comme XML qui, non seulement dispose d'une structure légère, mais aussi se stocke aisément dans un annuaire électronique. Le choix de l'annuaire dans ce cas de figure se justifiant par une consommation nettement moindre en ressources matérielles et par une performance accrue en lecture par rapport aux systèmes de gestion de base de données relationnels.
Dans cet ouvrage nous allons donc analyser la possibilité d'intégrer des fichiers XML en tant que fichiers de configuration pour quelques agents d'un système informatique de surveillance réseau : OpenMaster. Bien sûr nous ne passerons pas en revue l'ensemble de tous les agents, mais nous nous concentrerons principalement sur les trois les plus utilisés : l'agent de contrôle, l'agent d'extension et l'agent COACH.

Dans la suite de cette introduction, nous ferons tout d'abord connaissance avec OpenMaster et son architecture. Nous décrirons aussi brièvement les caractéristiques et les fonctionnalités des agents que nous allons étudier. Le second chapitre nous plongera alors plus en détails dans les caractéristiques des fichiers de configuration qui devront être traduits en feuilles XML. Le troisième chapitre nous présentera ensuite les fichiers DTD et les schémas XML afin de mieux comprendre leur utilité et leur importance dans notre projet, en apportant ensuite quelques précisions concernant la validation de fichiers XML. Le chapitre suivant nous présentera les parseurs Expat, SAX et DOM en tant qu'outils de traitement et de validation, suivi d'une approche plus pratique concernant leur implémentation à l'aide de trois langages assez répandus dans la programmation : Perl, C++ et java. On y parlera aussi un peu du problème de portabilité de ces outils. Le dernier chapitre quant à lui abordera l'aspect de la centralisation à l'aide d'un annuaire, dont la principale fonction sera de distribuer les fichiers de configuration. On expliquera également comment on peut ranger des feuilles XML dans un système basée sur une organisation arborescente grâce aux similitudes qui existent entre ces deux technologies. On terminera alors avec une petite présentation d'un outil de développement de projets Internet, que nous utiliserons pour administrer les configurations ainsi que leurs différentes versions à partir d'un simple

navigateur comme Netscape ou Internet Explorer.

1.1 Qu'est-ce qu'OpenMaster ?

OpenMaster regroupe un ensemble d'applications permettant l'administration de ressources aussi diverses que des réseaux, des applications ou des utilisateurs système. Actuellement, cet ensemble fonctionne sous des environnements UNIX, Windows NT et Windows 2000. Une architecture classique d'OpenMaster se compose des quatre éléments décrits ci-dessous.

1.1.1 Les objets

Les objets surveillés par OpenMaster sont en général des serveurs, des applications, des équipements réseaux, ou bien les utilisateurs connectés au système. Pour récolter les informations, on doit installer des petits logiciels qui font du "monitoring" sur ces objets : ce sont les agents. Pour être opérationnel, ils doivent bien entendu être installés.

1.1.2 Le serveur OpenMaster

Outre un serveur web optionel, indispensable uniquement si on souhaite installer des stations java dans le système, le serveur comprend trois composantes principales que nous allons succinctement présenter ci-dessous.

Le framework L'architecture OpenMaster est basée sur le modèle du Framework (plateforme de gestion). Sans entrer dans les détails, le framework est un bloc software sophistiqué qui gère les signaux envoyés par les agents. Ces informations peuvent être consultées par l'intermédiaire des applications.

Les modules d'application serveur Ce sont les services de base disponibles lors de l'installation. On compte parmis ces services la gestion des avertissements, la gestion du "monitoring", la gestion des performances du système,...
Les applications envoient le résultats des requêtes au framework qui se charge de transmettre les informations à la station comme on le voit sur le schéma.

Les modules additionnels Ce sont des modules qui ne sont évidemment pas livrés en standard. Ce sont entre autre les "toolkits" ou encore des modules conçu sur mesure pour des clients particuliers.

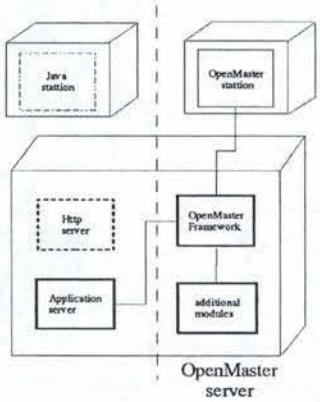

FIG. 1 – Schéma général et simplifié d'un serveur OpenMaster

1.1.3 Les stations OpenMaster

Ces stations sont munies d'une série d'applications permettant d'obtenir des informations sur les objets gérés. Ces logiciels sont un point principal de transit de l'information. Ils permettent entre autre de récolter des informations système et d'en stocker une image dans une base de données internes. Ils gèrent les alertes et collectent les informations requises par les différentes applications de gestion.

1.1.4 Les java stations d'OpenMaster

La station java est un composant optionnel permettant d'administrer OpenMaster à partir d'une machine non dédiée. Cet élément apporte des facilités en ce qui concerne l'administration mobile, car contrairement aux stations classiques, cette administration peut se faire à partir de n'importe quel station de travail. Cette technologie,basée sur un serveur web, offre comme gros avantage de permettre à plusieurs personnes d'accéder aux mêmes serveurs en même temps. D'un autre point de vue, le modèle "multi-tier" des stations java réduit les ressources nécessaires à leur fonctionnement. Sur certaines stations, ont peut activer l'option "station scalability end user" qui permet la connexion d'un plus grand nombre de stations java au système (puisque par défaut leur nombre est limité).

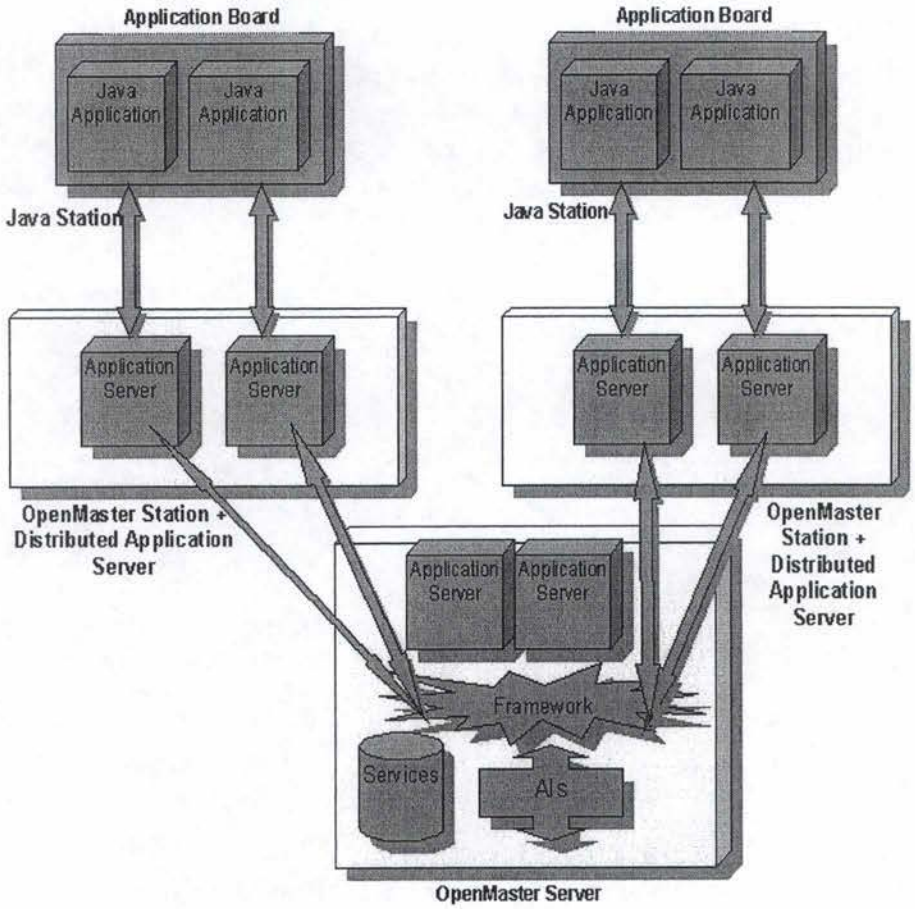

FIG. 2 – Description détaillée de l'architecture OpenMaster

1.2 Les agents

OpenMaster est une plate-forme de gestion développée à partir du modèle du protocole SNMP, élaborée en 1987. Celui-ci se présente sous la forme "agent/manager" reposant sur le protocole IP (Internet Protocole). On peut consulter trois normes qui décrivent les différentes composantes de ce protocole :

- La norme RFC1157 qui définit le protocole, les opérations de contrôle de base(Get, Get-next, Set,...) et la notification (trap)
- La norme RFC1155 qui précise la structure et l'identification de l'information de gestion contenue dans les MIB SNMP et fournit le modèle d'information générique pour la gestion des réseaux
- La norme RFC1212 qui se présente comme un complément au protocole précédent

Les agents OpenMaster sont donc des agents SNMP. De manière très schématique, un agent peut être décomposé en couches standard de communication, chacunes de ces couches offrant une syntaxe pour encoder/décoder ainsi que des méthodes pour manipuler les objets.

1.2.1 L'agent de contrôle

Description Cet agent gère plusieurs types de ressources nommées "alarm ressources". Lorsque cet agent détecte une anomalie, il a la possibilité d'envoyer un signal (trap) au serveur OpenMaster et d'effectuer des opérations correctives. Ces actions dépendent bien entendu de la nature de la source. Selon le résultat d'une interrogation on peut envisager plusieurs cas de figure comme d'envoyer une trap, lancer une procédure corrective ou encore les deux à la fois.

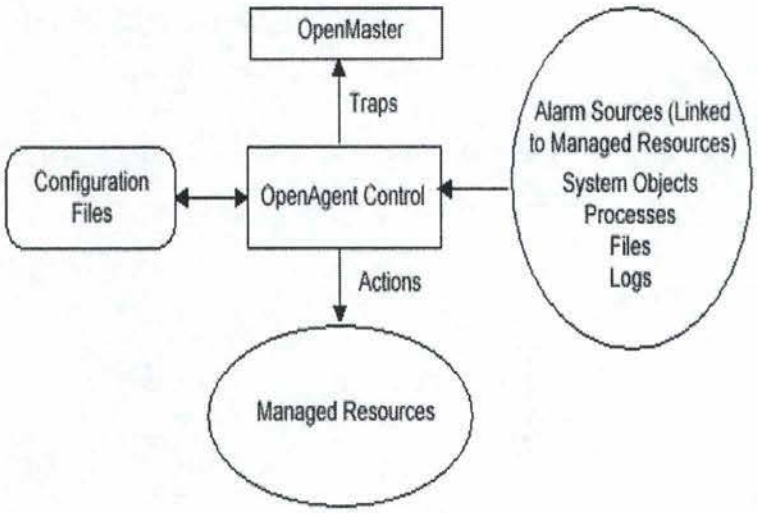

FIG. 3 – Schéma général de l'agent de contrôle

Utilisation Avant toute utilisation, il faut configurer cet agent. Ceci peut se faire soit par l'intermédiaire de la MIB, pour configurer manuellement une source d'alerte, soit en utilisant un fichier dit de "configuration". Une configuration se borne surtout à ajouter des nouvelles sources d'alertes, ou modifier celles qui existent déjà.
Chacunes des sources d'alarme peut être actives ou non et des périodes de polling (période entre deux interrogations d'état) peuvent être définies indépendamment. Regardons de plus près les différentes possibilités qui nous sont offertes.

1. Gestion de la swap et des systèmes de fichiers (FileSystems)

   L'agent de contrôle surveille le taux d'occupation de la partition swap (partition d'échange entre la mémoire et le disque dur sur les systèmes Unix) ainsi que des systèmes de fichiers, et ce à l'aide de deux bornes définies comme étant le "low et high watermark". Par défaut, si le taux monte à un niveau supérieur au "high watermark" une trap d'alerte est envoyée. De manière similaire si le taux d'occupation redescent en deça du "low watermark", une trap "clear" est également générée (elle indique un retour à une situation normale).
   Cette fonctionnalité n'est pas disponible sur tous les systèmes et notamment sous windows qui n'utilise pas de partition de swap.

2. Gestion des processus et fichiers

   Dans ce cadre, la surveillance se fait sur les occurences des processus et sur les opérations effectuées sur les fichiers. Selon la même philosophie, le nombre de processus (ou leur occurence) doit être compris entre des bornes définies dans la configuration. Au niveau fichier, il est particulièrement intéressant de contrôler la taille des fichiers de journalisation "log", qui ont une fâcheuse tendance à grossir très vite.

3. Gestion des fichiers de logs

   En plus, l'agent de contrôle dispose encore d'une fonctionnalité qui permet de contrôler la présence d'expressions régulières (et donc de messages d'erreur par la même occasion) en utilisant la technique du "pattern matching". Sans vraiment entrer dans les détails, disons que les expressions sont associées à un objet de la classe "filter element", le tout regroupé dans un objet "filtre". Le fichier log est passé par les différents éléments de filtre jusqu'à ce qu'une action soit déclenchée.

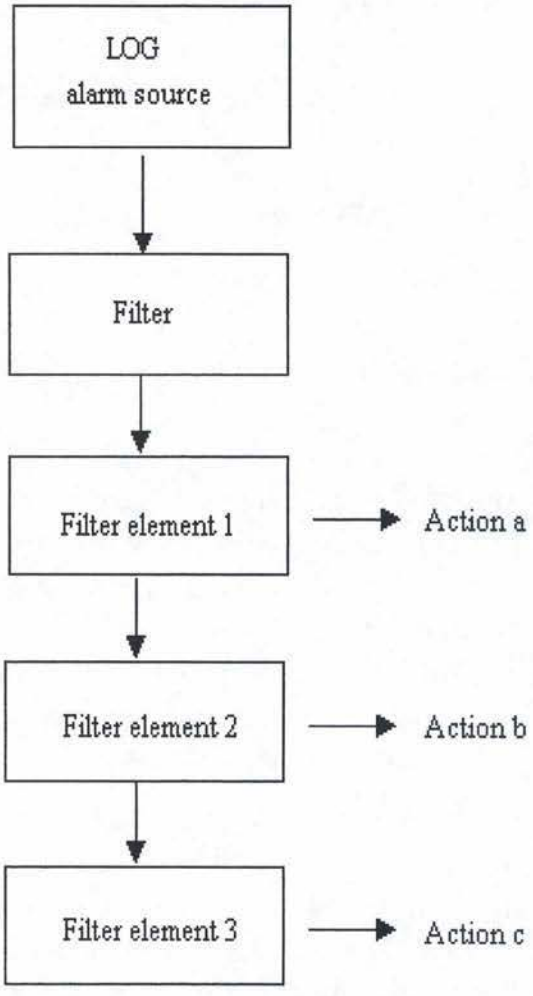

FIG. 4 – Schéma d'utilisation des filtres

4. Gestion des programmes

   Pour cette partie, un programme de gestion est activé à intervalles de temps réguliers. En fonction du résultat de ce programme, une trap est générée ou non.

L'agent de contrôle est bien sûr une entité de surveillance très complexe.

1.2.2 L'agent d'extension

Description De manière générale, l'agent d'extension est utilisé dans le but de retrouver des données dans une MIB sans pour autant devoir développer un agent complet. L'obtention des informations se fait à l'aide de commandes ou bien par lecture de fichiers.

Utilisation Le module d'extension permet de construire des agents en charge de répondre directement à des requêtes sur des MIB, sans devoir interroger la plateforme de gestion. Les utilisateurs pourront dès lors créer un agent SNMP à partir du module d'extension qui répond aux requêtes concernant les MIB sans pour autant à avoir à le développer entièrement.
Lorsqu'un agent d'extension reçoit une commande SNMP pour interroger une table dont il a la charge, il fournit la réponse selon le format définit à la configuration ; à savoir qu'il dispose des correspondances entre les attributs de la table SNMP interrogée et les colonnes du tableau du résultat.

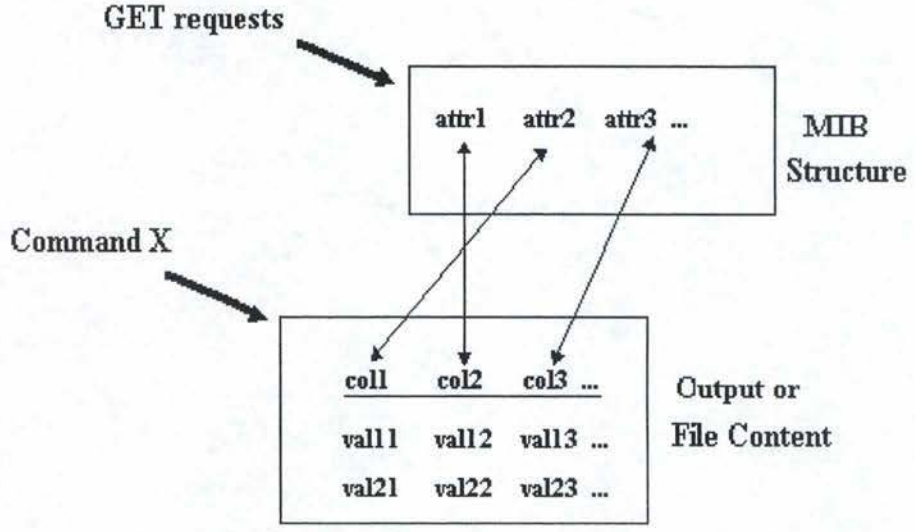

FIG. 5 – Schéma général de l'agent d'extension

1.2.3 L'agent COACH (Open Agent Concentrated Handling)

Description COACH est un agent qui a pour but de fournir un moyen de distribuer les services d'administration OpenMaster à travers de grands réseaux hétérogènes. Cette distribution devient nécessaire lorsque :

- Le réseau de l'entreprise est composé d'un grand nombre de systèmes autonomes
- Ce réseau est susceptible de se développer
- Les diagnostiques et la supervision prédictive impliquent une augmentation du traffic de données

Un service de distribution prédictif est basé sur les principes suivants :

- Transmission asynchrone et notifications d'événements non sollicités, la technique inverse du polling
- Traitement local des données
- Granularité faible des traitements parallèles
- Fonctionnement autonome de la qualité des liens entre les agents et le gestionnaire central

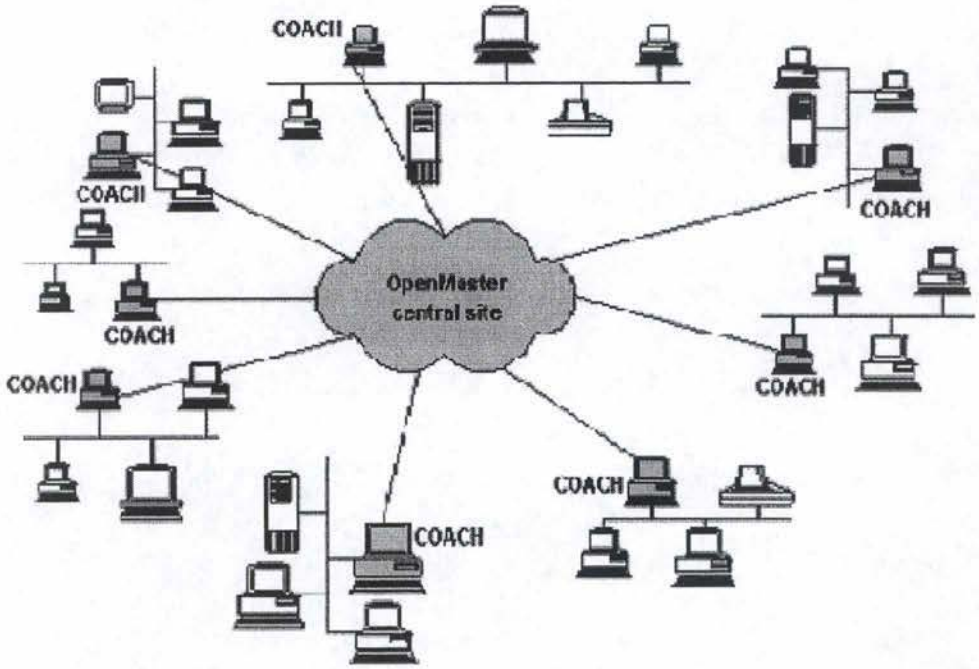

FIG. 6 – Schéma général de l'agent COACH

Utilisation Chaque composant d'OpenMaster présent sur une machine possède un agent, ce qui peut considérablement en augmenter le nombre si on travail au niveau du réseau local comme présenté sur la figure ci-dessus. De ce fait, afin de faire face à cette multitude, on peut envisager d'utiliser COACH suivant deux optiques qui, en fin de compte, n'a qu'un seul but : réduire le nombre d'informations transmises sur le réseau et alléger ainsi la charge du gestionnaire central.

Dans un premier temps, on peut voir COACH comme un gestionnaire central au niveau local, c'est-à-dire que certains événements seront traités localement sans avoir de répercution au niveau du gestionnaire central. Autrement dit, certaines décisions de traitement seront gérés par COACH sans que le gestionnaire central en soit informé.
D'autre part, on peut voir COACH comme un "relais" pour les informations issues d'agents

présents sur d'autres machines d'un même réseau local par exemple. Cette décentralisation de gestion a deux objectifs principaux :

- Il soulage les réseaux WAN reliant les sous-réseaux et le gestionnaire. Généralement ce sont des connexions très chargées ou bien très couteuses, car la tarification se fait au niveau des trames transférées ou bien au niveau du temps de connexion(lignes louées par exemple). COACH est capable de limiter le traffic sur ces lignes afin de ne transfèrer que les informations de gestion pertinentes
- Il soulage la plateforme de gestion centrale, qui sinon, devrait traiter un nombre très important de systèmes relatifs à tous les réseaux locaux connectés

Un autre avantage de COACH est qu'il offre une panoplie de possibilités de journalisation afin de prévenir toute perte d'informations. Ces archives peuvent être envoyées à la plateforme centrale sur demande ou à des moments particuliers comme durant la nuit. COACH est aussi capable d'envoyer des messages prioritaires directement à la plateforme centrale de manière sécurisée. Dans ce cas, l'algorithme de transmission n'arrête d'avertir le gestionnaire que lorsque celui-ci lui aura envoyé une confirmation de réception. Finalement, afin de remplir totalement sa mission, COACH dispose d'un mécanisme lui permettant de découvrir lui-même son environnement réseau. Ainsi chaque nouveau sous-résau sera toujours reconnu si les filtres lui permettent de l'être, et chaque déconnexion sera enregistrée.

Finalement, pour être complet dans notre exposé, parlons un peu de l'architecture de COACH, qui se compose en fait d'un ensemble de processus de communication. Le système dispose donc de six méthodes qui interagissent entre-elles par l'intermédiaire d'un Kernel, qui fournit également les moyens de communiquer avec le gestionnaire via le protocole SNMP.

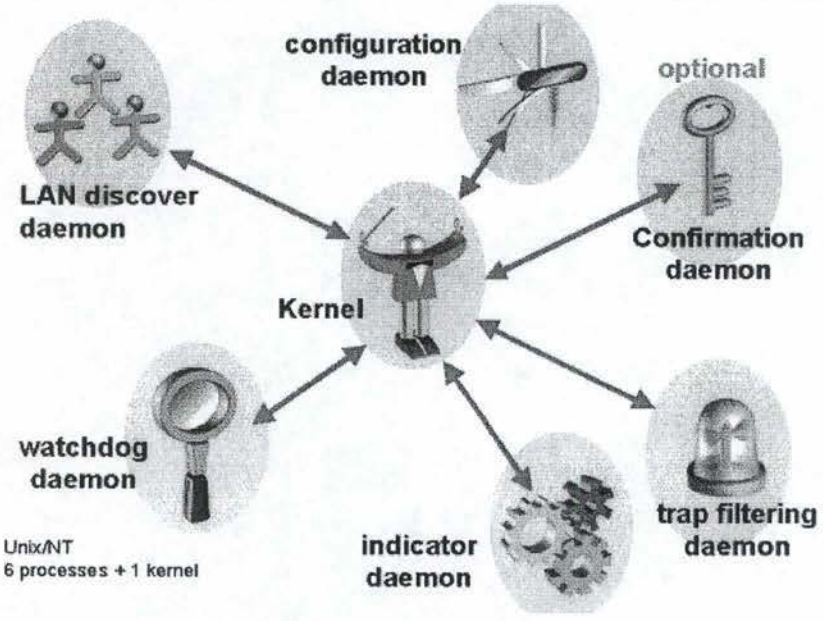

FIG. 7 – Schéma général du kernel de COACH

Passons maintenant en revue ces méthodes :

- Coach_discover : il reconnait les sous-réseaux et détermine à quel domaine les machines découvertes appartiennent ; ce qui signifie entre autre de déterminer quels agents SNMP tourne sur ces machines. C'est un bon complément au service centralisé "IP Discovery"
- Coach_indic : il évalue les équations sur les machines dont le système COACH à la charge. Ces équations sont en fait les indicateurs que l'on retrouve dans "confmod.ini"
- Coach_filtrage : il est conçu suivant la même optique que la méthode de calcul des indicateurs. Ce module filtre en fait les "traps" envoyées par les agents afin de ne transmettre que les plus pertinentes au gestionnaire central
- Coach_confmod : il est le lien entre la méthode qui découvre les machines et les deux méthodes précédentes. Ce module permet de configurer les informations à calculer et les "traps" à filter
- Coach_watchdog : il contrôle la découverte de machines. Si une machine disparait, il en avertit le gestionnaire et met à jour la liste des machines gèrées par COACH
- Coach_udpconf : ce module optionnel fournit une transmission sécurisée pour les "traps" SNMP depuis COACH jusqu'au gestionnaire central

1.3 Où en sommes-nous ?

A ce stade, nous n'avons encore rien de bien concret en ce qui concerne les objectifs que nous nous sommes fixés pour ce travail. A vrai dire, nous venons de présenter le cadre général dans lequel nous allons évoluer tout au long des chapitres suivants. La base de réflexion se fera donc au niveau des fichiers de configuration et non au niveau des agents eux-mêmes. Cependant, il nous apparaissait indispensable de faire un tour d'horizon du système concerné afin de mieux faire comprendre au lecteur les causes des enjeux auquels nous allons être confrontés par la suite, tant pour la centralisation que pour la gestion des configurations.

2 Descriptif de l'environnement de travail

2.1 Etude de l'existant

Le chapitre précédent nous a présenté l'architecture générale d'une plate-forme OpenMaster. Nous allons ici nous intéresser plus particulièrement aux agents et à la structure de leur fichier de configuration. De manière générale, chaque agent installé sur une machine dispose d'un fichier de configuration local.

2.2 Inconvénients de l'existant

Les agents sont installés sur les machines que l'on souhaite surveiller. Comme nous venons de le dire, chaque agent dispose sur cette machine d'un fichier de configuration valable uniquement sur la machine en question. Ce qui implique que chaque modification doit alors être effectuée sur toutes les plateformes que l'on désire mettre à jour, avec un risque d'incohérence et donc d'erreur très grand. En outre, les modifications sont actuellement faites à partir d'une application graphique, simple à utiliser mais dont le nombre de manipulations augmente considérablement le temps de mise à jour.
L'idée est donc de trouver une solution pour centraliser et simplifier la gestion des configurations de ces agents.

2.3 Les fichiers de configuration des agents

L'objectif ici est de faire connaissance avec la structure générale des fichiers de configurations de nos agents. Nous allons donc essayer ici de voir comment nous allons organiser les feuilles XML que l'on va associer à chacun d'eux, et ce afin de correspondre au mieux avec la structure du fichier existant. Plus les deux structures seront proches l'une de l'autre, plus le traitement en vue d'une conversion de la version XML vers la version dites "originale" sera aisée. En effet, comme l'un des buts a atteindre est de pouvoir configurer un agent à partir du XML, il faudra sans doute y intégrer un "convertisseur de format". En outre, il parait raisonnable de garder une certaine compatibilité avec les fichiers classiques de configuration. Le meilleur moyen pour y parvenir semble, à première vue, de tout traduire dans un même format. Quant à ce convertisseur dont nous venons de parler, nous verrons au troisième chapitre en quoi il consiste exactement.

Le code ci-dessous représente un exemple de configuration pour l'agent de contrôle et l'agent d'extension.

```
CLASS alixd {
        alixdMaxTrapsPerMinut =     300 ;
        alixdPath       =       /icmagent/etc/agix:/icmagent/etc/generix:/
            icmagent/etc:/icmagent/install/install_script:/etc:/bin:/usr/bin:/
            usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/local/bin:/usr/dt/bin:/
            home/alain/bin:.:/icmagent/install/instAgent:/icmagent/perl/bin:/
            icmagent/install_tmp/bin:/icmagent:/icmagent/bin:/usr/ccs/bin:/
            home/ismdev/bin ;
}
CLASS alixdProcessEntry {
    INSTANCE "Zombies.<defunct>" {
        alixdPrMaxNumber    =       5 ;
        alixdPrMinNumber    =       0 ;
    }
}
CLASS alixdSourcesEntry {
    INSTANCE "agovin_IsFtpOk" {
        alixdSourcesType    =       12 ;
        alixdSourcesFrequency   =   300 ;
        alixdSourcesFile    =       "grep '^ftp' /etc/inetd.conf" ;
    }
}
```

```
CLASS generix {
        generixTraceLevel   =       0 ;
        generixPath     =       /icmagent/etc/agix:/icmagent/etc/generix:/
            icmagent/etc:/icmagent/install/install_script:/etc:/bin:/usr/bin:/
            usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/local/bin:/usr/dt/bin:/
            home/alain/bin:.:/icmagent/install/instAgent:/icmagent/perl/bin:/
            icmagent/install_tmp/bin:/icmagent:/icmagent/bin:/usr/ccs/bin:/
            home/ismdev/bin:/icmagent/etc/generix/internet ;
}
```

Dans les exemples que nous venons de voir, nous pouvons observer une similitude structurelle parfaite entre ces deux fichiers de configuration. De plus, une première analyse rapide permet d'en comprendre la structure générale :

- Une classe "alixd" pour l'agent de contrôle ou "generix" pour l'agent d'extension regroupant les attributs généraux

- Un ensemble de classes diverses regroupant plusieurs instances

La structure du fichier de configuration de notre dernier agent concerné par notre étude est, quant à elle, totalement différente car elle est répartie sur deux fichiers que nous allons présenter.

1. confmod.ini

   Il reprend l'ensemble des filtres et des indicateurs qui seront utilisés par l'agent COACH. L'exemple ci-dessous nous montre comment sont déclarés les filtres et les indicateurs.

```
FIL 1 alix-fs-nfull 2 bull.118 6 4 1 60
FIL 2 alix-fs-error  2 1.3.6.1.4.1.107.114 6 5 0 6
FIL 3 alix-uxLoginSession-setFailed 2 Bull.118 6 33 3 30
IND 1 ifUtilizationBandWith 2 (8*$-(ifInOctets.1+ifOutOctets.1)/$t)/ifSpeed
    .1 600 10 1 1200 > LOG 1
IND 2 ifUtilizationBandWithAll 3 (!SUM(ifUtilizationBandWith))
    1210 10 1 3600 > LOG 1
IND 3 ifDiscards 2 ($-(ifInDiscards.1+ifOutDiscards.1)) 120 1 1 120 > LOG 1
IND 4 ifDiscardsAll 3 (!SUM(ifDiscards)) 320 3 1 320 > LOG 1
IND 5 coachIfOutQlen 2 (ifOutQLen.1) 330 5 1 330 > LOG 1
IND 6 coachIfOutQlenAll 3 (!SUM(coachIfOutQlen)) 670 50 1 670 > LOG 1
IND 7 coachtcpRetransSegs 2 (tcpRetransSegs.0) 340 5 1 340 > LOG 1
IND 8 ifErrors 2 ($-(ifInErrors.1+ifOutErrors.1)/$t) 290 5 1 290 > LOG 1
IND 9 ifErrorsSUM 3 (!SUM(ifErrors)) 620 2 1 620 > LOG 1
IND 10 ifErrorsMOY 3 (!MOY(ifErrors)*100) 630 5 1 630 > LOG 1
IND 11 ifInPackets 2 (ifInUcastPkts.1+ifInNUcastPkts.1)/$t 280 5 1 280 >
    LOG 1
IND 12 ifOutPackets 2 ((ifOutUcastPkts.1+ifOutNUcastPkts.1)/$t)
    280 5 1 280 > LOG 1
IND 13 ifErrorsRatio 2 (&ifErrors/(&ifInPackets+&ifOutPackets))
    570 5 1 570 > LOG 1
IND 14 ifErrorsRatioLinkMOY 3 (!MOY(ifErrorsRatio)) 1220 5 1 1220 > LOG
    1
IND 15 ifErrorsRatioLinkSUM 3 (!SUM(ifErrorsRatio)) 1220 20 1 1220 > LOG
    1
IND 16 ipInputErrors 2 ($-(ipInHdrErrors.0+ipInAddrErrors.0))
    650 5 1 650 > LOG 1
IND 17 ipInputErrorsPercent 2 (&ipInputErrors/($-(ipInDelivers.0)))
    *100 650 5 1 650 > LOG 1
IND 18 ipInputErrorsPercentOnLink 3 (!SUM(ipInputErrorsPercent))
    300 5 1 300 > LOG 1
```

Si nous regardons de plus près, nous verrons que chacunes de ces lignes respecte une syntaxe et un ordre bien défini, où chaque élément représente une information particulière. Si on s'en réfère à la documentation de l'agent COACH, on trouve les définitions suivantes respectivement pour les filtres et les indicateurs :

(a) Type (FIL signifie filtre)
(b) Id du filtre
(c) Nom du filtre
(d) Domaine où le filtre est appliqué
(e) Champ Entreprise : OID de l'objet "entreprise" concerné par le filtre
(f) Champ Generic
(g) Champ specific
(h) Occurence : nombre de survenances de l'événement concerné par le filtre avant l'envoit d'un avertissement(trap)
(i) Période : période de remise à zéro du compteur "Occurence" si aucun événement du type concerné n'a été reçu

(a) Type (IND signifie indicateur)
(b) Id de l'indicateur
(c) Nom de l'indicateur
(d) Domaine où l'indicateur est appliqué
(e) Equation de l'indicateur
(f) Tpolling ou période de polling de l'indicateur
(g) Threshold : seuil de décision pour l'envoit d'une "trap"
(h) Occurence : nombre de dépassements de seuil avant l'envoit d'un avertissement(trap)
(i) Période : période de remise à zéro du compteur "Occurence" si aucun dépassement de seuil n'a été reçu
(j) Opérateur de comparaison pour définir un dépassement de seuil
(k) Indicateur de journalisation : indique si l'indicateur est journalisé lors de la journalisation général du système
(l) Trap spécifique : détermine le type de trap à envoyer si un seuil est dépassé

2. aic.inst

   C'est le "véritable" fichier de configuration de l'agent. Il se compose de trois catégories d'éléments :

(a) La gestion des domaines ou "Domain Setup"
Ici on définit les différents paramètres propres à chaque domaine. Ainsi lorsqu'une machine est découverte sur le réseau, on vérifie quels sont les domaines pour lesquels cette machine remplit toutes les conditions "d'admission". Ce qui implique qu'une machine peut appartenir à plusieurs domaines simultanément

```
cfgDomainLabel.2 mib 2
cfgAtt1.2 sysUpTime.0
cfgDomainLabel.3 coach
cfgAtt1.3 logOnTraps.0
cfgDomainLabel.4 UNIX System
cfgAtt1.4 uxSysType.0
cfgAtt1Comparaison.4 1
cfgAtt1Value.4 "UNIX"
cfgDomainLabel.5 NT System
cfgAtt1.5 ntP01CaptureTime.0
cfgDomainLabel.11 Inconnu
cfgDomainLabel.16 einet
cfgAtt1.16 einetStatusId
cfgAtt1Comparaison.16 9
```

(b) La gestion de découverte de machines ou "Discovery Setup"
Ce sont les paramètres qui définissent le comportement de découverte de l'agent COACH. On lui indique ici les adresses IP à scanner pour trouver de nouvelles machines. On peut fournir pour cela trois types de "scope" : un masque de sous-réseau("cfgDiscoverSubnetMask"), une plage de valeurs("cfgDiscoverIpLower" et "cfgDiscoverIpUpper") ou une liste imposée("cfgDiscoverIpDomain")

```
cfgDiscoverDomainAtNextTime.0 0
cfgDiscoverIpLower.1 *.*.*.1
cfgDiscoverIpUpper.1 *.*.*.254
cfgDiscoverPeriod.0 50
cfgDiscoverRedoPeriod.0 3
cfgDiscoverSubnetMask.0 255.255.255.255
cfgDiscoverTimeOut.0 80
```

(c) La gestion des paramètres de journalisation ou "Log parameter Setup"
Il s'agit ici de définir les comportements de journalisation de COACH. Ces paramètres seront repris sous la rubrique "cfgSystem" de la feuille XML. Ils re-

prennent entre autre les champs "logOnIndicators", "logOnTraps", "detachIndicatorLogs",...

```
systemSaveConfiguration.0 0
systemSnmpCommunity.0 public
systemVersion.0 2.0
detachIndicatorLogs.0 0
detachTrapLogs.0 0
logOnIndicators.0 1
logOnTraps.0 1
```

Les éléments de configuration que l'on retrouve dans le fichier "aic.inst" se composent d'un ensemble de lignes issues des trois catégories que nous venons de découvrir. A ces dernières viennent encore s'ajouter les résultats des filtres, des indicateurs ainsi que quelques autres informations diverses gérées par le système. L'utilisateur n'ayant aucun contrôle sur ces renseignements, nous n'auront donc pas à nous en préoccuper ici.
Si on regarde maintenant comment sont écrites ces lignes, on observe une structure en deux parties :

- le nom du champs suivi de son index (le séparateur étant le caractère ".")
- la valeur du champs

Cela signifie que chacune de ces trois catégories sont indexées, et que l'on doit pouvoir retrouver et associer cet index aux éléments de configuration qui s'y rapportent. Une donnée importante dont il faudra tenir compte dans la feuille XML.

2.4 Description brêve de la philosophie du langage XML

Le langage XML (eXtensible Markup Langage) a été développé par un groupe de travail XML (initialement connu comme étant le comité d'examen éditorial SGML) sous la direction du consortium W3C. Il décrit une classe d'objets de données appelés documents XML. Ce langage de balisage extensible structure des données dans un document à la manière du HTML. On peut y adjoindre des DTDs, ceux-ci étant écrit en SGML (version simplifiée). On peut y écrire ses propres balises et XML utilise le jeu de caractère Unicode(ISO10646) pour le contenu de ses balises.

Définition du W3C "Les documents XML se composent d'unités de stockage appelés entités, qui contiennent des données analysables ou non. Les données analysables se composent de caractères, certains formant les données textuelles, et le reste formant le balisage. Le balisage décrit les structures logique et de stockage du document. XML fournit

un mécanisme pour imposer des contraintes à ces structures."

2.5 Pourquoi utiliser XML ?

XML a été conçu pour pouvoir utiliser des documents structurés de manière complexe sur Internet. Les balises n'ont pas vraiment de sens en elle-même, elles décrivent les contenus et la structure sans avoir beaucoup d'influence sur la présentation. XML permet de structurer, stocker et envoyer de l'information sur le web. Pour ajouter un aspect sémantique, il faut lui adjoindre des outils supplémentaires comme les schémas, les espaces de nommages, les feuilles de style (css,xsl,...). Les avantages liés à XML sont les suivants :

- Stockage sous forme de texte, ce qui implique une création et une édition à l'aide d'un simple éditeur.
- La structure donnée aux données permettent une manipulation aisée par les applications.
- La fragmentation permet une utilisation partielle des documents.
- La structure est hiérarchique, les accès y sont donc plus pratique.
- Ils peuvent être stockés dans une SGBD.
- L'échange de données entre applications incompatibles entre-elles est possible : simplification des échanges.

XML et HTML Contrairement à ce que l'on pourrait croire, XML et HTML sont des langages très différents. Il serait faux d'affirmer que XML serait la "prochaine version" de HTML et ce de par la fonction même des ces langages. HTML a été conçu pour afficher des données et se concentre donc sur l'aspect présentation tandis que XML est plus spécialisé dans la modélisation des données.

2.6 Les espaces de nommages

Comme cela a déjà été dit précédemment, on peut créer ses propres balises dans un document XML. L'ensemble de ces noms constituant ce que l'on appelle le "vocabulaire de balisage" pouvant être utilisé par plusieurs modules logiciels, et ce par soucis de modularité. Hors ils se peut que dans la même feuille XML, on veuille y mettre de l'information à destination de plusieurs applications, ce qui implique des risques de collisions, c'est-à-dire que deux balises puissent avoir le même nom mais balisant des données sémantiquement très différentes. Pour éviter à avoir à imposer un vocabulaire différent pour chacune des applications, on utilise des mécanismes comme l'espace de nommage. Les noms des espaces de nommage XML peuvent avoir la forme de noms qualifiés, qui contiennent un caractère deux-points séparant le nom en un préfixe d'espace de nommage et une partie locale.

Exemple :

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <!-- le prefixe ''edi'' est lie a http://ecommerce.org/schema
       pour l'element ''x'' et son contenu --></x>
```

On peut trouver un aperçu des recommandations du W3C concernant les espaces de nommages à l'adresse suivante :

http ://www.w3.org/TR/1999/REC-xml-names-19990114/Overview.html

2.7 Où en sommes-nous ?

Nous venons de prendre contact avec les différents fichiers de configuration des agents. Nous n'avons pas encore abordé la construction XML qui devrait en découler, mais nous avons pu observer les caractéristiques de la structure qui vont nous permettre de concevoir la DTD que l'ont va associer à chacun des agents, et par là même définir la structure de nos feuilles XML.

La seconde partie de ce chapitre nous a permis de découvrir la philosophie d'XML ainsi que les différents avantages qu'il procure comme format d'échange. Nous avons donc fait connaissance avec l'outil qui sera à la base de tout ce que nous développerons dans le cadre de ce travail.

3 Des DTD aux schémas XML

Dans ce chapitre, nous allons suivre une démarche qui va nous permettre de traduire les fichiers de configuration que nous avons détaillés dans le chapitre précédent. Nous commencerons tout d'abord par travailler avec des DTD pour ensuite vite s'apercevoir que toutes les contraintes requises par le projet ne pourront être vérifiées par ce moyen, sauf en utilisant un procédé, certes fonctionnel, mais qui s'assimile peut-être un peu trop à un bricolage. Nous nous tournerons alors vers un autre type de fichier, qui lui, sera en mesure de gérer ces extensions de manière simple : les schémas XML.

3.1 Les fichiers DTD

Pour commencer, il serait peut-être intéressant d'analyser l'origine de ces documents DTD. Tout est parti du langage SGML(Standard Generalized Markup Language) qui fut le premier langage de formats normalisés de données structurées reconnus par l'ISO en 1986. Ce dernier existait cependant depuis 1969 et fut développé par un certain Charles F. GoldFarb qui réalisa le produit commercial GML (Generalized Markup Language) d'IBM. SGML peut être utilisé pour définir des structures logiques génériques(SLG). Ces structures sont représentées par un ensemble de balises qui correspondent à un type particulier de documents, et qui permet d'en décrire la structure logique. Les SGL définies avec SGML sont appelées DTD(Description Type Document).
Le fichier DTD ainsi associé à des documents XML leur impose des contraintes de structures. Autrement dit, ils définissent les balises autorisées, l'ordre dans lequel elles apparaissent, celles qui sont obligatoires et celles facultatives,.... Mais ces fichiers en eux-même ne garantissent pas la validité d'un document, puisqu'on peut écrire un fichier XML sans pour autant leur associer une DTD. La validation de la conformité doit se faire par l'application qui utilisera le document XML, et cela à l'aide d'un "validateur" dont nous approfondirons le sujet plus loin.

3.1.1 Pourquoi utiliser des fichiers DTD ?

En fait, nous venons de répondre partiellement à cette question dans le paragraphe précédent. En général, un fichier DTD spécifie les éléments autorisés dans un document, en d'autres termes on définit les attributs attendus pour chacun d'eux. Il est clair que n'importe quel fichier de configuration créé doit pouvoir être utilisé par n'importe quel agent auquel ce fichier se rapporte. Afin de garantir que les options présentes seront prises en compte, il est important de pouvoir vérifier la validité des paramètres. C'est là le rôle que nous attendons des DTD associées aux documents représentant la configuration d'un agent.

3.1.2 Description des DTD des agents de contrôle et d'extension

Nous entrons maintenant dans le vif du sujet. Nous allons nous intéresser ici aux fichiers de configuration de ces deux agents. Grâce à leur ressemblance structurelle parfaite, nous pourrons utiliser la même méthodologie indépendamment pour l'un comme pour l'autre.

Afin de mieux comprendre la façon de procéder, nous allons commencer par regarder de plus près la découpe arborescente envisagée pour représenter respectivement les DTD de l'agent de contrôle et d'extension.

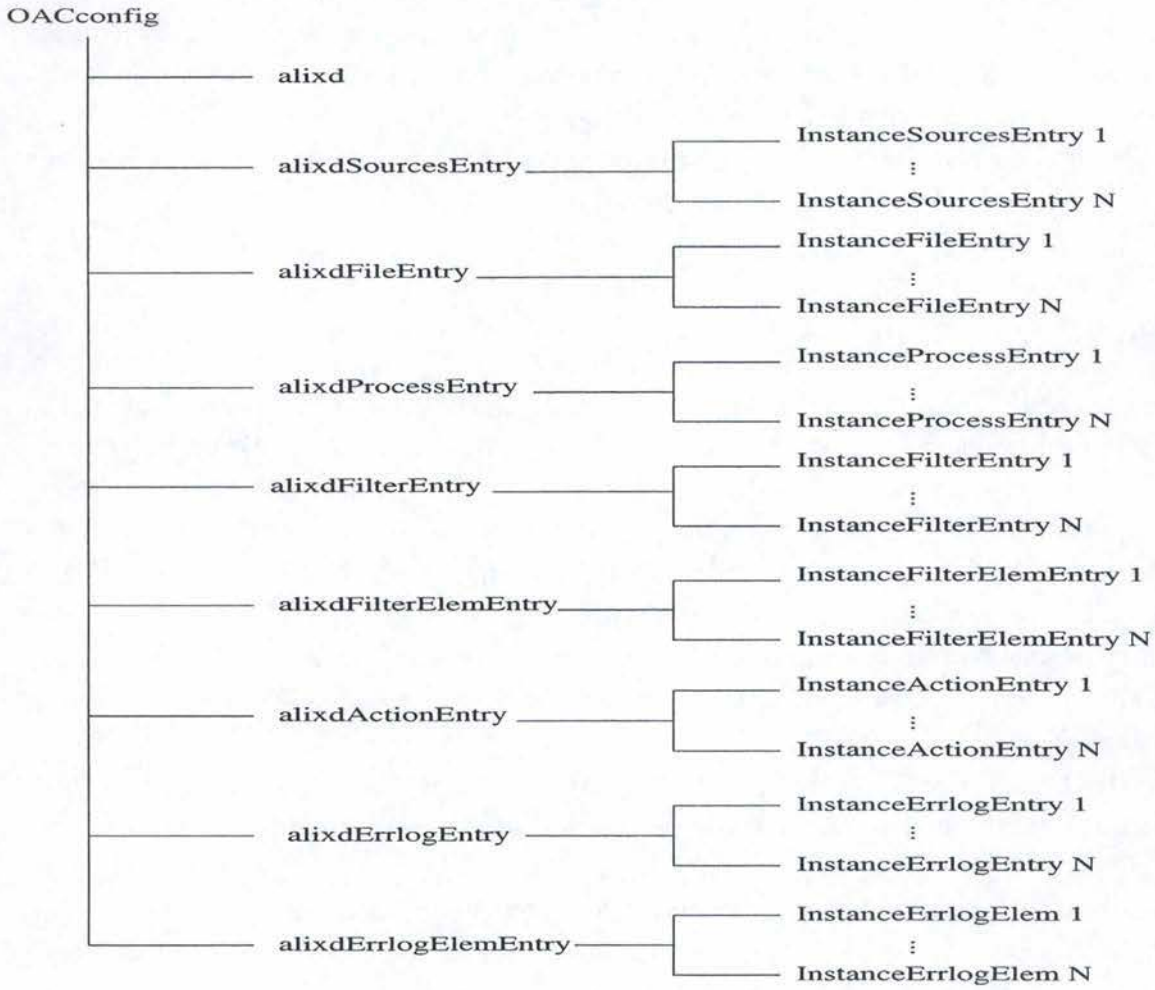

FIG. 8 – Schéma de l'arborescence de la configuration de l'agent de contrôle

Rappelons-nous d'abord ce que nous avions observé précédemment concernant le fichier de configuration de cet agent, à savoir qu'il se composait d'une classe générale alixd et d'autres classes diverses regroupant un ensemble d'instances. Pour mieux faire comprendre le rôle des instances, on pourrait comparer la classe à un tableau et les instances aux entrées dans ce tableau. Maintenant, regardons de plus près la découpe proposée pour la structure de la DTD. Nous avons un élément racine "OACconfig" qui indique à quel type de configuration nous avons affaire. Ensuite, au second niveau nous retrouvons l'ensemble des tableaux appartenant à l'agent, et enfin au dernier niveau, nous retrouvons les instances. Dans notre optique, les différents attributs de configuration seront respectivements pour alixd et les instances des attributs d'éléments XML. Nous allons maintenant voir comment tout cela va se traduire en format DTD.

Pour l'élément racine nous avons la déclaration suivante

```
<!ELEMENT OACconfig (alixd,alixdSourcesEntry?,alixdFileEntry?,
                     alixdProcessEntry?,alixdFilterEntry?,
                     alixdFilterElemEntry?,alixdActionEntry?,
                     alixdErrlogEntry?,alixdErrlogElemEntry?)>
```

Dans cette déclaration nous retrouvons plusieurs choses. D'abord, tous les autres éléments en font partie, donc nous sommes bien en présence de l'élément racine. Ensuite, il implique qu'"alixd" est obligatoire et que les autres éléments sont optionnels via le point d'interrogation.

Pour l'élément général "alixd" nous avons la déclaration suivante

```
<!ELEMENT alixd EMPTY>
    <!ATTLIST alixd
        alixdTraceLevel (none|requests|functions|details) #IMPLIED
        alixdSaveConfigFlag (true|false) #IMPLIED
        alixdSaveConfigWhenExitFlag (true|false) #IMPLIED
        alixdOldPasswd CDATA #IMPLIED
        alixdNewPasswd CDATA #IMPLIED
        alixdPollingStatus (enabled|disabled) #IMPLIED
        alixdMaxTrapsPerMinut CDATA #IMPLIED
        alixdPath CDATA #IMPLIED
        alixActionLogType (overwrite|append|separate) #IMPLIED
    >
```

Analysons un peu cette structure un peu plus complexe que la précédente. L'élément "alixd" est de type "EMPTY", cela signifie qu'aucun caractère ne sera autorisé entre la balise de début et celle de fin. Ensuite l'élément nommé "ATTLIST" définit la liste des attributs relatifs à "alixd".
La définition d'un attribut comporte trois parties : son nom, son type et une "déclaration de valeur implicite". Comme la première partie ne posera sans aucun doute aucun problème de compréhension, nous nous concentrerons sur les deux autres. Pour le type, on remarque ici deux cas de figure : soit on utilise un chaine de caractère entre parenthèses avec un "pipe" (le caractère "|") comme séparateur, soit on utilise le mot clé "CDATA". La première déclaration représente un type énuméré qui indique explicitement la liste exacte des valeurs autorisées, et le type CDATA, quant à lui, représente un attribut qui va comporter une chaine de caractères quelconques. Pour la dernière partie, on retrouve "#IMPLIED", qui signifie que l'attribut est facultatif. En XML cela donne par exemple ceci :

```
<alixd alixdTraceLevel="none"
       alixdSaveConfigFlag="false"
       alixdPollingStatus="enabled"
       alixdPath="/icmagent/etc/agix:/icmagent/etc/generix:/icmagent/etc:"
       alixdMaxTrapsPerMinut="200"
       alixActionLogType="overwrite"></alixd>
```

Le caractère facultatif de certains attributs se justifie par le fait que des valeurs par défaut sont prévues au cas où on aurait omis de les spécifier. Nous verrons par le suite que d'autres, à la fonction bien précise, seront requis et d'autres fixés.

Les déclarations de tableaux

```
<!ELEMENT alixdSourcesEntry (InstanceSourcesEntry)+>
<!ELEMENT alixdFileEntry (InstanceFileEntry)+>
<!ELEMENT alixdProcessEntry (InstanceProcessEntry)+>
<!ELEMENT alixdFilterEntry (InstanceFilterEntry)+>
<!ELEMENT alixdFilterElemEntry (InstanceFilterElemEntry)+>
<!ELEMENT alixdActionEntry (InstanceActionEntry)+>
<!ELEMENT alixdErrlogEntry (InstanceErrlogEntry)+>
<!ELEMENT alixdErrlogElemEntry (InstanceErrlogElemEntry)+>
```

Les déclarations ne sont pas difficiles à comprendre. Ici, chaque élément se compose d'un moins un élément d'instance tout simplement.

Les déclarations d'instances

```
<!ELEMENT InstanceFilterElemEntry EMPTY>
    <!ATTLIST InstanceFilterElemEntry
        alixdFilterElemFilterName CDATA #REQUIRED
        alixdFilterElemIndex CDATA #REQUIRED
        alixdFilterElemStatus (normal|overThreshold) #IMPLIED
        alixdFilterElemType CDATA #FIXED "regularExpression"
        alixdFilterElemExpression CDATA #IMPLIED
        alixdFilterElemCount CDATA #IMPLIED
        alixdFilterElemThreshold CDATA #IMPLIED
        alixdFilterElemDuration CDATA #IMPLIED
        alixdFilterElemLastMatchTime CDATA #IMPLIED
        alixdFilterElemLastMessage CDATA #IMPLIED
        alixdFilterElemTriggering (overMatch|overThreshold) #IMPLIED
        alixdFilterElemAction CDATA #IMPLIED
        alixdFilterElemSeverity (indeterminate|critical|major
                                 |minor|warning|clear) #IMPLIED
    >
```

Les commentaires, ici, sont les mêmes que ceux d' "alixd". Cependant, nous observons deux petits détails qu'il convient d'expliquer. Tout d'abord, on trouve deux champs "#REQUIRED". Cette déclaration signifie qu'ils sont requis et donc obligatoires. Ils représentent en fait les deux champs utilisés pour construire l'index de l'instance. Pour mieux se faire une idée, regardons l'extrait de code (d'un autre tableau) ci-dessous, issus d'un fichier de configuration de l'agent de contrôle :

```
CLASS alixdProcessEntry {
        INSTANCE "Zombies.<defunct>" {
                alixdPrMaxNumber        =       5 ;
                alixdPrMinNumber        =       0 ;
        }
```

La chaine "Zombies" représente la valeur de l'attribut alixdPrGroupName, et "<defunct>" celle de "alixdPrProcessName". Ces deux champs sont également de type "requis". La concaténation représente la "clé d'index" de l'instance, c'est-à-dire son identifiant dans le tableau "alixdProcessEntry". De par cette fonction d'identifiant, leur définition est alors essentielle.
Ensuite, nous avons une "déclaration de valeur implicite" "#FIXED" pour l'attribut "alixdFilterElemType". Cela signifie simplement que la seule valeur autorisée pour cet attribut est "regularExpression". De manière pratique, cette déclaration permet non seulement d'empêcher l'assignation d'une autre valeur, mais elle permet aussi de ne pas la déclarer explicitement. En effet, si l'attribut n'est pas présent dans le document, la majorité des outils conçus pour traiter les feuilles XML, que nous aborderons plus loin, renverront tout de même la valeur fixée pour le champs en question.
Nous venons à présent de montrer comment nous avons traduit le schéma arborescent de l'agent de contrôle en DTD. La version complète pourra être consultée dans la section "A" de l'annexe. De manière similaire, on pourrait refaire le même raisonnement pour l'agent d'extension, mais cela ne nous apporterait rien puisque les structures sont identiques. Néanmoins, le schéma ci-dessous expose l'arborescence proposée pour cet agent. La version complète de la DTD pourra elle aussi être consultée en annexe "A".

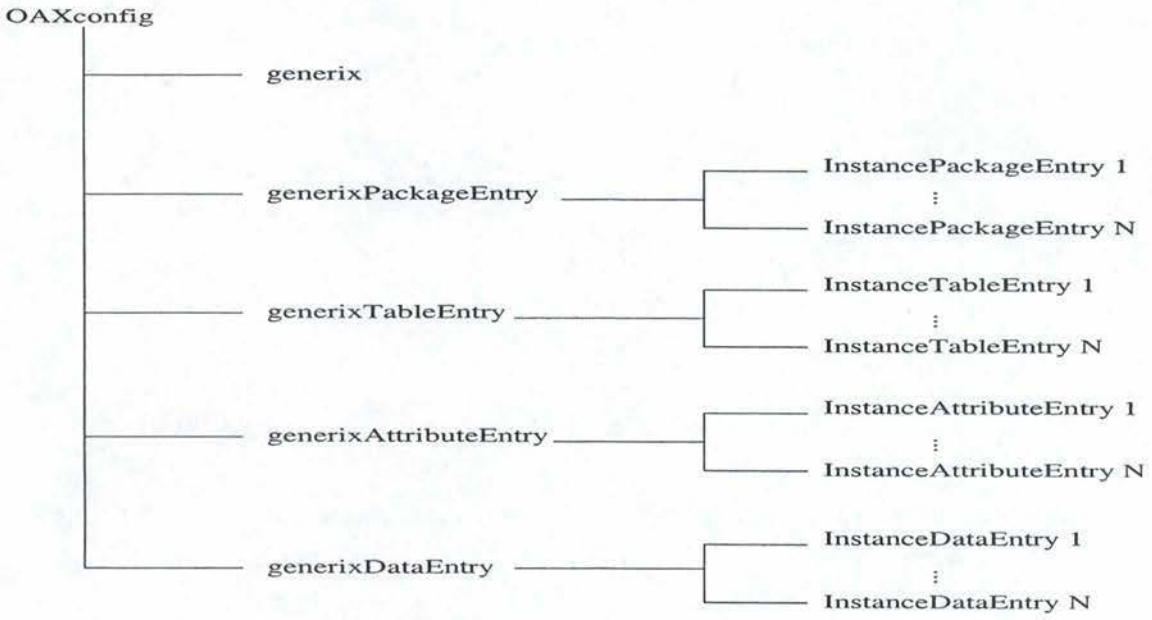

FIG. 9 – Schéma de l'arborescence de la configuration de l'agent d'extension

3.1.3 Description des DTD de l'agent COACH

Nous allons maintenant nous intéresser au dernier agent étudié dans le cadre de ce projet. Sa structure est un peu plus complexe, dû au fait notamment, de sa configuration étalée sur deux fichiers. L'arborescence envisagée ci-dessous décrit comment nous avons organisé le fichier de configuration :

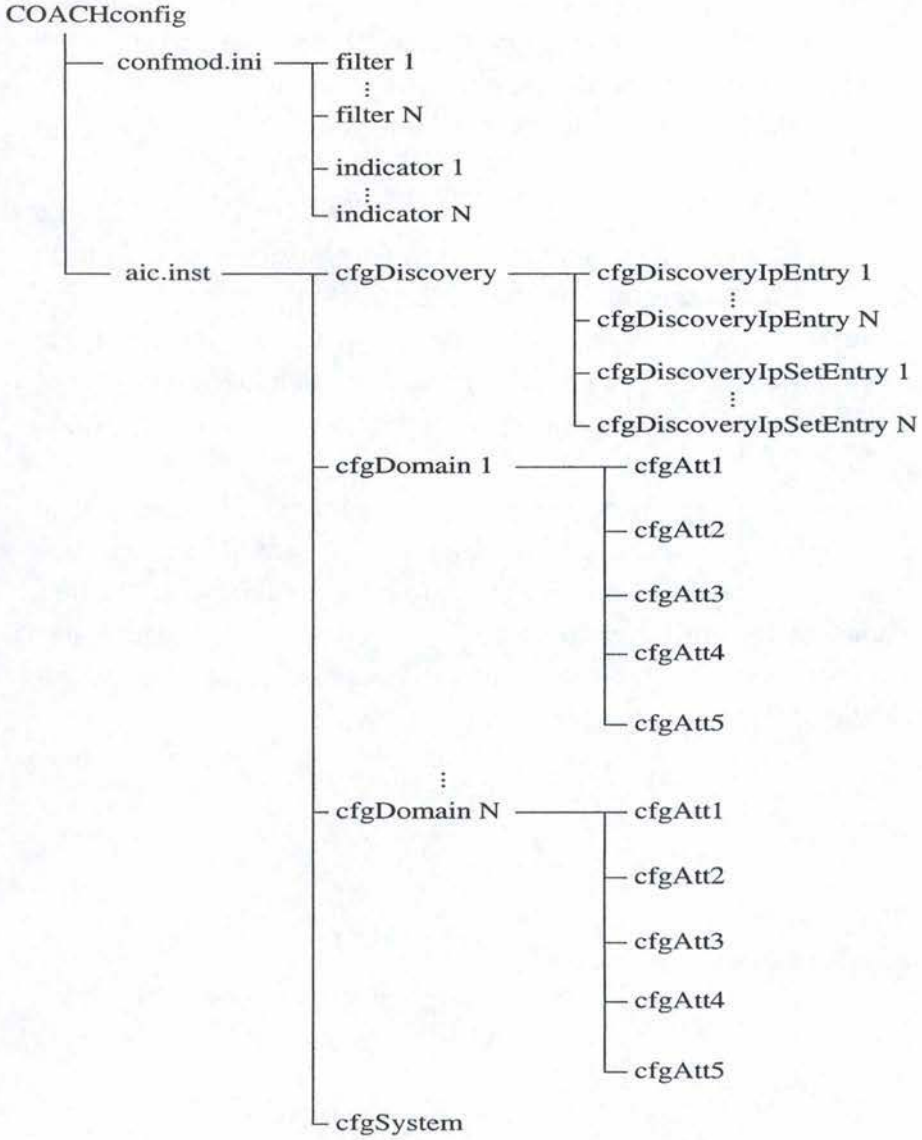

FIG. 10 – Schéma de l'arborescence de la configuration de l'agent COACH

Regardons la maintenant de plus près. D'abord les informations concernant les deux fichiers sont présentes sur la même feuille. Ensuite, nous retrouvons les filtres et les indicateurs ainsi que les trois catégories d'éléments de configuration : "Discovery", "Domain" et ce que nous avons appelé ici les "cfgSystem". La traduction de ces différentes parties sera très analogue à ce que nous avons vu dans la section précédente.

La déclaration des filtres et des indicateurs

```
<!ELEMENT filter EMPTY>
<!ATTLIST filter
    cfgFilterId CDATA #REQUIRED
    cfgFilterLabel CDATA #REQUIRED
    cfgFilterDomain CDATA #REQUIRED
    cfgFilterEnterprise CDATA #REQUIRED
    cfgFilterGeneric CDATA #REQUIRED
    cfgFilterSpecific CDATA #REQUIRED
    cfgFilterCptMax CDATA #REQUIRED
    cfgFilterPeriodValid CDATA #REQUIRED
>
```

La structure étant la même pour les filtres comme pour les indicateurs, nous nous contenterons de cet exemple. A remarquer surtout l'aspect obligatoire de tous les attributs, puisque ceux-ci occupent, comme nous l'avons déjà dit, une place bien définie dans chaque déclaration.

La déclaration de l'élément "Discover"

```
<!ELEMENT cfgDiscovery (cfgDiscoverIpEntry*,cfgDiscoverIpSetEntry*)>
<!ATTLIST cfgDiscovery
    cfgDiscoverId CDATA #FIXED "0"
    cfgDiscoverPeriod CDATA #REQUIRED
    cfgDiscoverSubnetMask CDATA #REQUIRED
    cfgDiscoverDomainAtNextTime (yes|no) #REQUIRED
    cfgDiscoverTimeOut CDATA #REQUIRED
    cfgDiscoverRedoPeriod CDATA #REQUIRED
>

<!ELEMENT cfgDiscoverIpEntry EMPTY>
<!ATTLIST cfgDiscoverIpEntry
    cfgDiscoverIpIndex CDATA #REQUIRED
    cfgDiscoverIp CDATA #REQUIRED
    cfgDiscoverIpDomain CDATA #IMPLIED
>

<!ELEMENT cfgDiscoverIpSetEntry EMPTY>
<!ATTLIST cfgDiscoverIpSetEntry
    cfgDiscoverIpSetIndex CDATA #REQUIRED
    cfgDiscoverIpLower CDATA #REQUIRED
    cfgDiscoverIpUpper CDATA #REQUIRED
>
```

Remarquons simplement ici l'aspect facultatif des sous-éléments "cfgDiscoverIpEntry" et "cfgDiscoverIpSetEntry", ainsi que la valeur imposée "0" pour l'attribut "cfgDiscoverId". Le caractère "*" signifie simplement la cardinalité (0-N).

La déclaration des éléments "Domain"

```
<!ELEMENT cfgDomain (cfgAtt1,cfgAtt2?,cfgAtt3?,cfgAtt4?,cfgAtt5?)>
<!ATTLIST cfgDomain
    cfgDomainId CDATA #REQUIRED
    cfgDomainLabel CDATA #REQUIRED
>

<!ELEMENT cfgAtt1 EMPTY>
<!ATTLIST cfgAtt1
    cfgAtt1 CDATA #REQUIRED
    cfgAtt1Value CDATA #REQUIRED
    cfgAtt1Comparaison (exist|equal|greater|different|lessequal|greaterequal
                        |less|include|notinclude|node) #REQUIRED
>
<!ELEMENT cfgAtt2 EMPTY>
<!ATTLIST cfgAtt2
    cfgAtt2 CDATA #REQUIRED
    cfgAtt2Value CDATA #REQUIRED
    cfgAtt2Comparaison (exist|equal|greater|different|lessequal|greaterequal
                        |less|include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt3 EMPTY>
<!ATTLIST cfgAtt3
    cfgAtt3 CDATA #REQUIRED
    cfgAtt3Value CDATA #REQUIRED
    cfgAtt3Comparaison (exist|equal|greater|different|lessequal|greaterequal
                        |less|include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt4 EMPTY>
<!ATTLIST cfgAtt4
    cfgAtt4 CDATA #REQUIRED
    cfgAtt4Value CDATA #REQUIRED
    cfgAtt4Comparaison (exist|equal|greater|different|lessequal|greaterequal
                        |less|include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt5 EMPTY>
```

```
<!ATTLIST cfgAtt5
    cfgAtt5 CDATA #REQUIRED
    cfgAtt5Value CDATA #REQUIRED
    cfgAtt5Comparaison (exist|equal|greater|different|lessequal|greaterequal
                        |less|include|notinclude|node) #REQUIRED
>
```

A chaque déclaration d'un élément "domain" doit être associé obligatoirement un sous-élément "cfgAtt1". Les autres sont facultatifs, mais doivent se suivre par leur numérotation. Par exemple si on souhaite déclarer un sous-élément "cfgAtt4", il faut impérativement avoir déclaré les sous-éléments "cfgAtt2" et "cfgAtt3". Cette contrainte n'a malheureusement pas pu être intégrée dans la DTD, et de ce fait, a dû être vérifiée d'une autre manière. Nous aborderons plus en détails les limitations que nous avons rencontrées avec les DTD dans la section suivante. Le caractère " ?" signifie simplement la cardinalité (0-1).

La déclaration de l'élément "cfgSystem"

```
<!ELEMENT cfgSystem EMPTY>
<!ATTLIST cfgSystem
    cfgSystemId CDATA #FIXED "0"
    systemSaveConfiguration (on|off) #IMPLIED
    detachTrapLogs (on|off) #IMPLIED
    detachIndicatorLogs (on|off) #IMPLIED
    logOnTraps (on|off) #IMPLIED
    logOnIndicators (on|off) #IMPLIED
    indicatorMaxTimePolling CDATA #IMPLIED
    indicatorPollingUnitPeriod CDATA #IMPLIED
    systemSnmpCommunity CDATA #IMPLIED
>
```

Les attributs repris ici sont quelques informations système configurables rarement utilisées par l'administrateur.

A nouveau, comme pour les deux autres agents, la version intégrale de la DTD de l'agent COACH se trouve aussi en annexe "A".

3.1.4 Limitations des DTD

Nous venons de construire les DTD des différents agents concernés par ce projet. On sait que les fichiers DTD définissent de manière structurée le format d'un document XML bien déterminé. Malheureusement, on ne peut exprimer toutes les contraintes que l'on voudrait, car pour une DTD les données sont des chaines de caractères uniquement. Ainsi, il est par exemple impossible de définir des types et des bornes numériques sur les valeurs d'entités ou d'attributs. Nous avons donc rencontrer certaines difficultés pour exprimer certains aspects que l'on aurait voulu pouvoir traiter de cette manière. Cependant, nous allons voir

qu'il est toujours possible de s'en sortir, soit avec un peu d'astuce, soit en se tournant vers une autre technologie.

Mais avant de changer complètement de cap et d'abandonner les DTD, nous allons d'abord présenter dans les deux sections suivantes deux solutions visant à contourner les problèmes que nous n'avons pas pu traiter directement, tout en conservant la DTD comme support de travail. La première se servira d'éléments "ENTITY" pour introduire les informations dont nous avons besoin, et la seconde stockera ces mêmes informations sur une feuille XML externe.

3.1.5 Limitations et éléments "ENTITY"

Avant d'entamer la description de la solution proprement dite, nous allons voir exactement quelles sont les contraintes que nous n'avons pas pu traiter. Ce qu'il faut dire aussi, c'est que le but de toute cette gymnastique pour traiter ces problèmes via la DTD trouve son origine dans l'obligation de cacher certains aspects de conception à l'administrateur. Par exemple, il n'est pas nécessaire que ce dernier s'inquiète de savoir quels sont les champs qui interviennent dans les différents index d'instances. D'un autre point de vue, il semble raisonnable de pouvoir modifier des bornes numériques de manière aisée, d'ajouter ou modifier des informations système comme les valeurs de remplacement sans devoir toucher au code des outils de traitement. Mais regardons maintenant nos quatre fameuses contraintes reprises ci-dessous :

- La vérification de bornes numériques pour des champs compris comme tels
- Le remplacement de valeurs alphanumériques par leur valeur numérique correspondante
- La construction de l'index des instances à partir d'attributs appartenant à cette instance
- La construction des déclarations d'indicateurs et de filtres suivant l'ordre imposé des attributs

Enumérés comme tels, ces problèmes ne paraissent pas forcément clairs aux premiers abords. Nous allons donc les expliciter un par un en détails.

La vérification de bornes numériques s'imposent pour certains attributs devant être considérés comme numériques. En effet, certains d'entre eux représentent des pourcentages, ce qui les amènent à être compris nécessairement entre 0 et 100 (voire 110 dans certains cas particuliers). D'autres sont des bornes modulables par l'administrateur mais ne pouvant, pour des raisons évidentes de logique, dépasser certaines valeurs. Par exemple, on évitera les valeurs négatives pour indiquer un nombre de vérifications par minute.

Le remplacement de valeurs alphanumériques concerne quant à lui les attributs de type "énuérés". Ce type est, rappelons le, caractérisé par une énumération de valeurs dans une liste entre parenthèses, où le caractère "pipe" est utilisé comme séparateur. Ce qu'il faut savoir, c'est que ces chaines de caractères ne sont pas écrites telles quelles dans le fichier de configuration. En fait, elles sont remplacées par des valeurs numériques afin de

faciliter le traitement par l'agent. D'un autre point de vue, ces mêmes valeurs numériques ne seraient pas vraiment intuitives pour le travail de configuration d'un administrateur. En suivant notre raisonnement, il est facile de comprendre que nous garderons les chaines de caractères dans la feuille XML, et qu'il sera dès lors indispensable de les convertir en chiffres lors de la conversion du document XML en fichier de configuration classique.

La construction de l'index des instances concerne plus particulièrement les agents de contrôle et d'extension. Rappelons-nous l'exemple avec l'instance "zombies.<defunct>". Les valeurs "zombies" et "<defunct>" viennent en fait de deux champs constituant l'index du tableau "InstanceProcessEntry". Ce qu'il faut savoir, c'est que lorsque l'on traite les attributs d'un élément, on n'est pas toujours certain de les retrouver dans l'ordre dans lequel ils sont déclarés dans le document. Tout dépend de l'algorithme de l'outil que l'on utilise (l'outil étant ici un parseur XML). De ce fait, on ne peut se fier à l'ordre de déclaration. Il faut donc trouver un autre moyen de démarquer ces champs particuliers (qui peuvent être unique ou au nombre de deux) afin de les retrouver très facilement. Or il existe bien un type ID qui permet de déterminer que chaque valeur de cet attribut sera unique dans tout le document, mais il ne peut être attribué qu'à un seul attribut par élément, d'où la nécessité de trouver une alternative.

La construction des déclarations d'indicateurs et de filtres doit se faire en disposant les informations dans un ordre très précis. Il faut donc trouver un moyen de donner un numéro d'ordre à chaque attribut.

Après ces petites précisions, nous sommes prêts à entamer la mise en oeuvre de la première solution. L'aspect général des déclarations des entités sont les suivantes :

```
<!ENTITY alixdMaxTrapPerMinut ''(0,100)''>
<!ENTITY alixdTraceLevelnone ''0''>
<!ENTITY alixdSourcesNameIdx ''1''>
<!ENTITY cfgIndicatorLabelPos ''2''>
```

Nous avons repris chaque solution dans le même ordre que les différents problèmes énumérés plus haut. Nous avons donc quatre possibilités de type d'entités :

1. le nom de "l'attribut"
2. le nom de "l'attribut" auquel on adjoint une des valeurs possibles
3. le nom de "l'attribut" auquel on adjoint les caractères "Idx"
4. le nom de "l'attribut" auquel on adjoint les caractères "Pos"

La première définition indique le nom de l'attribut ainsi que les bornes inférieure et supérieure qui lui sont associées. Ces bornes sont représentées sous la forme d'un couple de valeur.
Dans le second cas, le nombre "0" correspond à la valeur de remplacement de la valeur "none" pour l'élément "alixdTraceLevel". Nous venons ainsi de faire le lien entre le nombre "0" et la chaine de caractères "none" pour l'élément "alixdTraceLevel". Cependant, ce lien

est lourd de conséquence car il implique autant de déclaration d'entités qu'il n'y a de valeurs différentes possibles pour chaque élément de type énuméré, ce qui provoque une multiplication considérable de leur nombre. Cette difficulté vient du fait que la valeur "none" n'a pas forcément "0" comme valeur de remplacement pour tous les attributs de tous les éléments.
Quant aux deux dernières propositions, elles peuvent être englobés dans le même "sac". En effet, la seconde valeur indique soit la position du champs dans l'index d'un tableau, soit la position du champs dans la déclaration d'un indicateur (ou d'un filtre selon le cas). La fonction est la même dans les deux cas de figure.

Après avoir exposé les différentes propositions afin de résoudre nos problèmes, nous allons voir comment nous allons pouvoir aller rechercher cette information de manière pratique. Avant tout, il faut savoir que lorsque une application souhaite utiliser une DTD pour traiter un document XML, elle doit tout d'abord la lire et la stocker en mémoire. Lors de cette opération, certains outils offrent une possibilité d'action pour chaque déclaration d'entité. Il suffit alors de créer une phase d'initialisation durant laquelle une table de hashing ou un tableau "indicé" sera garni d'une nouvelle entrée à chaque lecture d'une nouvelle déclaration d'entité. En outre, on peut travailler avec un ou plusieurs tableau en fonction de la classification en quatre parties que nous venons de faire, et ce afin d'améliorer les performances de recherche. Les clés de recherche seront donc la première partie de la déclaration et la valeur indexée sera bien entendu la seconde. L'attribut sans contrainte, lui, sera identifié par son absence en tant que clé.
Une fois notre ou nos tableaux construits, nous sommes prêts pour le traitement. La vérification des bornes et la recherche des valeurs de substitution ne représente pas un traitement complexe, car il suffit dans le premier cas de tester une valeur numérique par rapport à deux autres, et dans le second cas de substituer une valeur à une autre. Là, où ça se complique, c'est lorsque nous devons reconstituer des valeurs dans un ordre très précis. En effet, on se rapelle sans doute bien que chaque élément "CLASS" d'une configuration d'un agent de contrôle ou d'un agent d'extension est une table où les instances sont les enregistrements. Chacun de ces enregistrements sont évidemment identifiés par une clé constituée d'un ou deux champs de l'enregistrement. La structure se présente sous la forme : champs 1.champs 2. ... .champs n
De même, les déclarations dans le fichier de configuration "confmod.ini" des indicateurs et des filtres de l'agent COACH sont composés de divers champs disposés dans un ordre très précis et non interchangeable.

La méthode proposée pour arriver au résultat escompté, consiste à parcourir en boucle tous les attributs d'un élément et d'aller rechercher les valeurs d'index leur correspondant. Dans certains cas, tous les champs seront concernés, et dans d'autres seulement certains. A chaque tour, et si ils sont utilisés pour la construction, les éléments seront classés par ordre croissant de clé d'index dans un autre tableau indicé temporaire. Ensuite, il suffira de restituer le résultat en parcourant séquentiellement ce fameux tableau temporaire dont nous venons de parler.

Nous venons ici de montrer une première façon de contourner nos petits problèmes. Cela dit, une telle implémentation est difficilement exploitable notamment par le nombre de déclarations qu'il suppose. Imaginez un instant d'avoir un élément d'index contraint par des bornes numériques. Dans ce cas, il vous faudra deux déclarations et deux tests pour traiter le même élément ...

3.1.6 Limitations et document externe

Nous allons maintenant essayer de voir si l'utilisation d'un document XML externe ne permettrait pas de solutionner nos problèmes d'une manière plus élégante. En fait, l'idée qui se cache ici est de reporter les informations que l'on introduisait à l'aide d'entités dans la DTD dans un autre fichier mieux structuré et donc plus simple d'emploi. L'organisation proposée pour ce fichier pourra être consultée dans la section "C" de l'annexe. Regardons maintenant comment nous avons mis les informations pour résoudre les problèmes que nous avons tenté de solutionner avec des entités. De manière générale, nous avons quatre niveaux d'arborescence. Outre l'élément racine "AgentTypes" nous avons en premier niveau le nom de l'agent : OAC pour l'agent de contrôle, OAX pour l'agent d'extension et COACH pour l'agent COACH. Ensuite, au second niveau nous retrouvons le nom des différentes instances comme "alixd" ou "InstanceSourcesEntry". Une petite remarque cependant, nous n'avons pas repris les tableaux car ils ne contenaient aucune informations nécessitant l'utilisation de cette feuille XML. Pour les deux derniers niveaux, des explications complémentaires s'avèrent nécessaires. Voyons de plus près le cas "InstanceSourcesEntry".

```
<InstanceSourcesEntry>

    <InstanceSourcesEntryId>
        <alixdSourcesName id="1"/>
    </InstanceSourcesEntryId>
    <alixdSourcesStatus>
        <enabled>1</enabled>
        <disabled>2</disabled>
        <inError>3</inError>
        <pending>4</pending>
    </alixdSourcesStatus>
    <alixdSourcesErrorInformation />
    <alixdSourcesFrequency />
    <alixdSourcesHighWaterMark inf="0" sup="100"/>
    <alixdSourcesLowWaterMark inf="0" sup="100"/>
    <alixdSourcesFile />
    <alixdSourcesType>
        <swap>1</swap>
        <filesystems>2</filesystems>
        <logFiles>3</logFiles>
```

```
            <fileGroup>4</fileGroup>
            <processGroup>5</processGroup>
            <syslog>6</syslog>
            <errlog>7</errlog>
            <printers>8</printers>
            <program>12</program>
        </alixdSourcesType>
        <alixdSourcesSpecificName />
        <alixdSourcesFilterName />
        <alixdSourcesDefaultAction />
        <alixdSourcesPasswd />

    </InstanceSourcesEntry>
```

Quand on y regarde de plus près, on observe trois possibilités.

1. le nom d'un élément suivi du suffix "Id" impliquant un quatrième niveau constitué des éléments de l'index ainsi que leur position (valeur de l'attribut "id")
2. le nom d'un élément sans attribut avec un quatrième niveau donnant la correspondance entre les différentes valeurs possibles et leur valeur de remplacement
3. le nom de l'attribut muni de deux attributs représentant les bornes numériques supérieures et inférieures
4. le nom d'un élément sans attribut ni quatrième niveau. Ce sont des champs alphanumériques sans contrainte. D'un point de vue pratique, ces derniers auraient pu disparaître complètement du fichier, mais dans un soucis de complétude, nous avons décidé de les conserver au cas où de nouvelles contraintes feraient leur apparition

Nous avons maintenant deux exceptions à traiter. La première concerne un champ particulier ayant un type énuméré et appartenant à un index. Le cinquième niveau ici donne la correspondance numérique comme on peut le voir ci-dessous.

```
    <InstanceErrlogEntryId>

        <alErrlogClass id="1">
            <hard>1</hard>
            <soft>2</soft>
            <other>3</other>
            <pend>4</pend>
        </alErrlogClass>
        <alErrlogIdentifier id="2"/>

    </InstanceErrlogEntryId>
```

Il existe une seconde exception à cette organisation : les informations relatives aux indicateurs et aux filtres. Pour eux, nous avions aussi besoin d'indiquer leur place respective

dans la déclaration des indicateurs et des filtres.

```
<indicator>

    <cfgIndicatorId pos="1"/>
    <cfgIndicatorLabel pos="2"/>
    <cfgIndicatorDomain pos="3"/>
    <cfgIndicatorEquation pos="4"/>
    <cfgIndicatorPeriodPolling pos="5"/>
    <cfgIndicatorThreshold pos="6"/>
    <cfgIndicatorCptMax pos="7" sup="20"/>
    <cfgIndicatorPeriodValid pos="8"/>
    <cfgIndicatorComparaison pos="9">
        <less>&lt;</less>
        <greater>&gt;</greater>
        <equal>=</equal>
        <different>!=</different>
    </cfgIndicatorComparaison>
    <cfgIndicatorLogOn pos="10">
        <on>LOG</on>
        <off>NLOG</off>
    </cfgIndicatorLogOn>
    <cfgIndicatorTrap pos="11">
        <aicTrapCritical>1</aicTrapCritical>
        <aicTrapMajor>2</aicTrapMajor>
        <aicTrapMinor>3</aicTrapMinor>
        <aicTrapWarning>4</aicTrapWarning>
        <aicTrapClear>5</aicTrapClear>
        <aicAutoTrapCritical>11</aicAutoTrapCritical>
        <aicAutoTrapMajor>12</aicAutoTrapMajor>
        <aicAutoTrapMinor>13</aicAutoTrapMinor>
        <aicAutoTrapWarning>14</aicAutoTrapWarning>
        <aicAutoTrapClear>15</aicAutoTrapClear>
        <aicMibTrapCritical>21</aicMibTrapCritical>
        <aicMibTrapMajor>22</aicMibTrapMajor>
        <aicMibTrapMinor>23</aicMibTrapMinor>
        <aicMibTrapWarning>24</aicMibTrapWarning>
        <aicMibTrapClear>25</aicMibTrapClear>
    </cfgIndicatorTrap>

</indicator>
```

Cette position est donnée par l'attribut "pos". Remarquons qu'il est toujours possible de définir les autres contraintes de bornage et de substitution.

Nous venons maintenant de faire le tour de la seconde solution qui ne nous a pas convaincu non plus. Nous allons dès lors essayer de nous tourner vers une autre technologie et abandonner les DTD, qui ne nous ont pas permis de travailler comme on l'aurait voulu en nous obligeant à avoir recours à quelques bricolages assez scabreux.

3.2 Une alternative intéressante : les schémas XML du W3C

La solution de la feuille XML, citée dans le paragraphe précédent, ressemble étrangement au principe des schémas XML. En outre cette solution comportait l'inconvénient de faire parcourir un arbre à chaque attribut rencontré afin d'en vérifier les bornes, ou bien d'en transformer la valeur mnémotechnique en sa valeur réelle utilisée par l'agent. La dégradation des performances sur les gros fichiers était facilement observable rendant le temps de réponse vraiment inacceptable. De ce fait, la nécessité de se tourner vers un autre système de travail devenait indispensable. De plus, l'utilisation de schémas XML vont permettre de simplifier les traitements des bornes, qui seront prises en compte dans le schéma lui-même, déchargeant ainsi le développeur de cette tâche.
Parmis tous les types disponibles dans le monde des schémas, ceux du consortium W3C semblent être les plus répandus. Non seulement ils permettent de définir toutes les restrictions disponibles avec les DTD, mais ils offrent en plus la possibilité de définir des types nous permettant d'inclure des contraintes bornées sur les valeurs numériques. Comme dans le cas des DTD, les quatres problèmes fondamentaux doivent à nouveau trouver une solution pratique et performante. Afin de démontrer les avantages des schémas sur les DTD dans le cadre de ce projet, nous avons adapté la structure de la feuille XML. En effet les différents attributs des instances sont devenus pour l'occasion des sous-éléments. Mais plutôt que de donner un descriptif complet de ce langage ou de montrer comment traduire des DTD en schémas XML, nous allons présenter les trois constructions différentes nous permettant de résoudre les quatre contraintes qui nous ont forcé à abandonner les DTD.
Pour commencer, nous allons vous présenter un extrait du schéma mettant en évidence la structure nous permettant de retrouver les positions de chaque champs dans une clé d'instance. Pour cela, nous avons adjoint à chacun des éléments concernés un attribut "index" dont la valeur fixée indique sa position dans cet index. Donc si on regarde le petit exemple ci-dessous, on en déduit que l'index de l'instance "InstanceFileEntry" se compose d'"alixdFileGroupName" en première position et d'"alixdFileIndex" en seconde position.

```
<xsd:element name="InstanceFileEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdFileGroupName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
```

```
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>

        <xsd:element name="alixdFileIndex">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                        <xsd:attribute name="index" fixed="2"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
```

Ainsi l'index de chaque champs est intégré de manière discrète et transparente pour l'utilisateur. Pour le restituer lors du traitement, il suffira de refaire la même chose que ce qui a été proposé dans la première solution, à savoir utiliser un tableau indicé temporaire. Ensuite, nous avons construit de manière similaire la partie de schéma relative aux indicateurs et aux filtres. L'extrait ci-dessous nous donne un petit aperçu de la partie déclarative d'un filtre.

```
<xsd:element name="filter">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgFilterId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="cfgFilterLabel">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>
```

Pour traiter les bornes numériques, nous allons déclarer des types qui feront intervenir des limites inférieure et supérieure. Les types sont classés dans deux grandes familles : les types simple et les types complexe. Les types simple, qui englobent les types classiques comme Integer ou String, peuvent directement être utilisé, tandis que les types complexes doivent d'abord être défini explicitement. En schéma XML, on le fait ainsi :

```
<xsd:complexType name="alixdErrlogElemEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceErrlogElemEntry"
                     minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
```

Après les avoir définit, on peut alors les assigner aux éléments de cette manière :

```
<xsd:element name="alixdErrlogElemEntry"
             type="alixdErrlogElemEntryType"/>
```

Seulement voilà, lorsque l'on ne doit utiliser un type qu'une seule fois, il serait dommage d'être obligé de lui donner un nom. Pour éviter cela, on utilise des types dits "anonymes". Grâce à ce mécanisme, le problème des bornes devient trivial. Il suffit alors de définir le type comme présenté ci-dessous :

```
<xsd:element name="alixdMaxTrapsPerMinut" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="200"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
```

Remarquons ici que les attributs "minInclusive" et "maxInclusive" représentant respectivement les bornes inférieure et supérieure.
La dernière construction va nous montrer comment on peut intégrer des informations destinées uniquement aux applications qui exploitent les feuilles XML relatives à un schéma particulier. Cette technique va nous permettre de faire la correspondance entre les valeurs reprises dans un type énuméré et les valeurs nuériques auxquelles ils correspondent. Pour cela, on utilise les balises "appInfo" qui ont été prévues pour permettre d'ajouter des informations de traitement supplémentaires. La description ci-desous correspond à l'équivalent DTD "`alixdSourcesStatus (enabled|disabled|inError|pending) #IMPLIED`"

```
<xsd:element name="alixdSourcesStatus" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
```

```
            <xsd:enumeration value="enabled">
                <xsd:appInfo>alixdSourcesStatus:1</xsd:appInfo>
            </xsd:enumeration>
            <xsd:enumeration value="disabled">
                <xsd:appInfo>alixdSourcesStatus:2</xsd:appInfo>
            </xsd:enumeration>
            <xsd:enumeration value="inError">
                <xsd:appInfo>alixdSourcesStatus:3</xsd:appInfo>
            </xsd:enumeration>
            <xsd:enumeration value="pending">
                <xsd:appInfo>alixdSourcesStatus:4</xsd:appInfo>
            </xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
```

Regardons maintenant ce qui se trouve effectivement à l'intérieur même de la balise : le nom de lélément auquel on se rapporte suivit d'un nombre ; le caractère " :" faisant ici office de séparateur. La valeur numérique représente bien entendu la valeur utilisée pour la conversion. Pour ce qui est du nom de lélément, il est vrai que dans un premier temps, il semble redondant, mais imaginons que "enabled" valent "1" pour "alixdSourcesStatus" et "2" pour un autre champ, il nous faut bien alors pouvoir distinguer ces deux cas. Mais pour mieux comprendre le raisonnement, regardons comment nous allons nous y prendre pour utiliser cette information.

Avant toute chose, le programme commence par faire un premier passage et repère toutes les balises de type "appInfo" à l'aide d'un outil de traitement approprié que nous détaillerons plus loin. Ensuite, il ajoute le contenu de chacunes d'entre elles dans, par exemple, une table de hashage en se servant de la valeur de la balise parente "énumération" comme clé, et de la valeur numérique présente dans appInfo comme valeur indexée. Mais comme on va devoir distinguer des valeurs différentes pour un même nom, ce n'est pas suffisant. Idéalement, il faudrait faire intervenir le nom de l'élément auquelle se rapporte cette valeur. Or il se fait que la découverte des éléments appInfo se fait en parcourant la feuille schéma (qui n'est rien d'autre qu'une feuille XML) avec un parseur DOM. Ce qui signifie, comme nous le verrons dans le chapitre suivant, que le résultat du parsing se trouve être un arbre. Grâce à cette organisation arborescente, on va pouvoir très facilement retrouver le nom de l'élément en question puisque il s'agit du noeud se situant à quatre niveaux au-dessus. Seulement, comme rien n'est jamais simple, si on s'intéresse au cas du champs "alErrlogClass" dans le schéma relatif à l'agent de contrôle, on se rend bien compte qu'avec une déclaration externe, il est bien difficile de trouver le nom du champs en question en se servant du lien hiérarchique. La raison qui nous a amené à définir ce type de manière externe vient du fait que cet élément était limité au point de vue des valeurs par une liste (type énuméré donc), mais qu'en plus il faisait partie d'un identifiant d'instance. Si on regarde la façon de traduire ces conditions, on s'aperçoit vite qu'elles sont incompatibles

entre-elles. En effet, pour identifier un champs comme étant un index il faut lui adjoindre un attribut "index", et pour définir la liste des éléments, il faut qu'il soit de type simple absolument. Or dans un schéma XML du W3C, les attributs ne peuvent être ajouté qu'a des éléments de type complexe. Nous sommes donc en face d'une contradiction que nous ne pouvons résoudre qu'en déclarant de manière externe un type simple auquel se trouve lié un attribut "index". Ensuite nous redéfinissons le champs "alErrlogClass" comme étant un type complexe représentant une extension du type simple cité plus haut. Afin d'éviter de jouer avec des tests supplémentaires et uniformiser les traitements, nous avons donc choisi d'indiquer le nom de l'élément dans la balise "appInfo".

L'extrait de code ci-dessous illustre ce que nous venons d'expliquer.

```
<xsd:simpleType name="alErrlogClassType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="hard">
            <xsd:appInfo>alErrlogClass:1</xsd:appInfo>
        </xsd:enumeration>
        <xsd:enumeration value="soft">
            <xsd:appInfo>alErrlogClass:2</xsd:appInfo>
        </xsd:enumeration>
        <xsd:enumeration value="other">
            <xsd:appInfo>alErrlogClass:3</xsd:appInfo>
        </xsd:enumeration>
        <xsd:enumeration value="pend">
            <xsd:appInfo>alErrlogClass:4</xsd:appInfo>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:element name="alErrlogClass">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="alErrlogClassType">
                <xsd:attribute name="index" fixed="1"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
```

Nous avons donc résolu toutes nos contraintes de manière assez simple. Ce type de schéma semble d'ailleurs bien adapté pour définir une structure logique et imposer des limites aux valeurs. En outre, les balises "appInfo" nous ont apporté une aide précieuse pour faire le lien entre des chaines de caractères et leur équivalent numérique.

Les schémas XML complets des différents agents OpenMaster pourront être consultés dans la section "D" de l'annexe.

3.3 Petite présentation de quelques autres langages pour les schémas XML

Nous donnons ici au lecteur des informations complémentaires sur d'autres langages de schémas XML. Comme le but n'est de toute façon pas d'être exhaustif, nous nous contenterons donc d'une description minimale suffisante afin de montrer d'autres outils de travail qu'il conviendrait peut-être d'analyser plus en profondeur ultérieurement.

3.3.1 Schematron

Avant d'entrer réllement dans le vif du sujet, nous allons d'abord définir deux normes sur lesquelles s'appuie le premier type de schéma "alternatif" présenté ici pour valider des documents XML.

La norme XSL

XSL ou "eXtensible Stylesheet Language" est un langage permettant de définir des feuilles de style, de manière analogue à "CSS" pour HTML. Il se compose d'ailleurs d'une part d'un langage de transformation de document XML(XSLT), et d'autre part d'un langage de description de sémantique de formatage. XSL est donc un outil utilisé pour la présentation de document XML indépendant de tout système ou logiciel. La particularité de XSL est de concevoir une spécification de présentation qui ne se limite pas au cadre d'Internet.

Une feuille XSL n'est rien d'autre qu'une feuille XML contenant des informations de transformation et de formatage d'objets. Le but étant de transformer un document XML en entrée en un autre document XML, dont les éléments de structure sont liés à des éléments typographiques représentant le formatage comme les sauts de page, les paragraphes,...
Le document résultat doit alors est pris en charge par un logiciel qui le transformera en un autre format dans un but d'impression par exemple(PDF,DVI,PS,...). Mais cette finalité d'impression n'est pas la seule car on pourrait aussi envisager un format de sortie en HTML pour un affichage sur le web.
Le compilateur XSLT est donc un transformateur de document qui travaille à partir d'une arborescence source construite sur base du document XML afin de lui faire subir des transformations, et ce sur base de règles appelées "templates rules". Ces règles sont d'ailleurs contenues dans la feuille de style XML. Chacune de ces règles se rapportent à un élément particulier du document afin de lui faire subir un traitement défini(transformer des balises XML en balises HTML par exemple). Il existe encore un autre langage XSL, le XSL/FO qui permet de définir la mise en page d'un résultat fournit par XSLT.
Petit exemple glané sur le site "www.commentcamarche.net" :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
    xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
<xsl:template match="/">
    <HTML>
        <HEAD>
            <TITLE>Titre de la page</TITLE>
        </HEAD>
        <BODY BGCOLOR="#FFFFFF">
            <xsl:apply-templates/>
        </BODY>
    </HTML>
</xsl:template >

<xsl:template match="personne" >
    <ul>
        <li>
            <xsl:value-of select="nom"/>
                    -
            <xsl:value-of select="prenom"/>
        </li>
   </ul>
</xsl:template>
</xsl:stylesheet>
```

Cette feuille de style va transformer la feuille XML suivante :

```
<personne>
    <nom>Pillou</nom>
    <prenom>Jean-Franois</prenom>
</personne>

<personne>
    <nom>VanHaute</nom>
    <prenom>Nico</prenom>
</personne>

<personne>
    <nom>Andrieu</nom>
    <prenom>Seb</prenom>
</personne>
```

Dans le format HTML suivant :

```
<ul>
    <li>Pillou - Jean-Franois</li>
```

```
        <li>VanHaute - Nico</li>
        <li>Andrieu - Seb</li>
    </ul>
```

La norme XPath

XPath ou "XML Path Language" est un langage permettant d'adresser les différentes parties d'un document XML. Il est d'ailleurs aussi utilisé par XSLT. XPath supporte des facilités pour manipuler des chaines de caractères, des nombres ou des booléens. la syntaxe utilisée par XPath est volontairement compacte et de nature non-XML afin que les expressions XPath puissent facilement être incluses dans des URI ou être utilisée comme valeur d'attribut. XPath s'applique sur la structure abstraite et donc logique d'un document XML et permet de naviguer à travers leur arborescence. Outre son utilisation pour l'adressage, XPath est aussi conçu pour tester si un noeud vérifie un pattern ou non, et ce grâce à XSLT. Pour XPath, un document XML est organisé en arbre où les noeuds sont de différents types : éléments, attributs, textes,ldots
La construction syntaxique de base se trouve être l'expression, qui sera évaluée pour trouver des objets de différents types définis dans la spécification. Cette évaluation devra se faire en fonction d'un contexte, qui sera lui déterminé par XPointer ou XSLT. Un contexte est toujours structuré comme suit :

- Un noeud de contexte
- Une paire de nombres strictement positifs (la position et la taille du contexte)
- Un ensemble de "variable bindings", c'est-à-dire un mapping entre les noms de variable et leur valeur
- Une librairie de fonctions, qui consiste à faire un mapping entre les noms de fonction et les fonctions en elles-mêmes
- Un ensemble de déclarations d'espaces de nom en rapport avec l'expression, en sachant qu'une déclaration de ce type consiste en un mapping entre les préfixes et les URI d'espace de nom

Parmi les expressions, la plus importante est la "location path", qui sélectionne un ensemble de noeuds relatifs à un noeud de contexte. Donc, le résultat de l'évaluation d'une expression "location path" donne un ensemble de noeuds dont les éléments constitutifs sont les noeuds sélectionnés par la "location path".

Voici quelques exemples d'expressions "Location Path" :

- child : :para sélectionne les éléments fils "para" du noeud de contexte
- child : :* sélectionne tous les éléments fils du noeud de contexte
- child : :text() sélectionne les éléments fils texte du noeud de contexte
- child : :node() sélectionne tous les éléments fils du noeud de contexte, peu importe le type de noeud

- attribute : :name sélectionne l'attribut nom du noeud de contexte
- attribute : :* sélectionne tous les attributs du noeud de contexte
- descendant : :para sélectionne les éléments "para" descendant du noeud de contexte
- ancestor : :div sélectionne les éléments "div" ascendant au noeud de contexte

Les spécifications XPath donne encore des précisions sur la syntaxe des autres expressions, ainsi que les fonctions d'évaluation qui doivent être présentes dans toutes implémentation de XPath. Nous ne les aborderons pas ici. Par contre nous alons encore donner quelques détails concernant le modèle de données.
Comme nous l'avons djà dit au début de cette section, pour Xpath un document XML est une arborescence. Il existe cependant sept types de noeud distincts :

- root nodes
- element nodes
- text nodes
- attribute nodes
- namespace nodes
- processing instruction nodes
- comment nodes

Les spécifications XPath donne une description des types présentés ci-dessus mais aussi les règles à respecter pour construire une arborescence correcte. Par exemple, le noeud racine doit être unique, les fils d'un noeud doivent être définis à un niveau inférieur, chaque noeud fils n'a qu'un et un seul parent,...

Le langage "schematron"

Nous allons introduire ici un genre différent de schéma que celui que nous avons abordé dans la section précédente. Schematron est un langage de schéma développé par Rick Jelliffe et qui est basé sur le paradigme du "tree pattern" ; tandis que les W3C schémas et les DTD étaient plutôt orientés "grammaire régulière". Schematron a spécialement été conçu pour la validation.
Un document schematron se présente sous la forme d'un ensemble de règles(assertions) basée sur le modèle XPATH permettant de valider un document XML. Mais voyons cela plus en détails...

Les éléments de base sur les "assert" et les "reports". Ils définissent ce que l'on appelle des contraintes et représentent des tests booléens entre les patterns du documents XML et les assertions du schematron.

Exemple :

```
<assert test="count(walls) = 4">
    This house does not have four walls
</assert>
```

Cette contrainte traduit le fait que dans un certain contexte(nous verrons de suite de quoi il s'agit) le nombre d'éléments "walls" doit être égal à quatre dans le document. Si cette condition n'est pas remplie, alors le parseur renvoit un message d'erreur à l'utilisateur.
Bien que les "report" et les "assert" semblent être diamétralement opposé, leur utilisation première est un peu différente. On utilisera des "assert" pour tester si un document est conforme à un schéma particulier(et générant une ou des actions en cas de non respect) tandis que les "report" sont utilisés pour mettre en valeur les caractéristiques de données d'un niveau inférieur.

Exemple :

```
<report test="not(roof)">This house does not have a roof</report>
```

Les actions à entreprendre si une assertion échoue ou si un "report" réussi ne sont pas spécifié par schematron ; elles sont à charge du développeurs. Le comportement par défaut se contente d'envoyer un avertissement à l'utilisateur. Afin d'encore améliorer le feedback, on peut associer à une assertion ou à un report un attribut "diagnostic" permettant d'inclure des informations supplémentaires personnalisées sur l'origine de l'erreur. Pour ce faire la déclaration se fait comme suit :

```
<assert test="count(walls) = 4" diagnostics="1">
                            This house does not have four walls</assert>
    ...
<diagnostics>
    <diagnostic id="1">
        Its an odd house which has more or less than four walls!
        Consult your architect...
    </diagnostic>
        ...
</diagnostics>
```

Remarquons que ces "diagnostic" sont groupés à l'extérieur de toute définition de contraintes. Comme nous l'avons vu plus haut, les assertions et les report sont exécutés dans un contexte particulier. Dans un schematron, on regroupe ces éléments de base relatifs au même contexte dans ce que l'on appelle une règle(rule). La définition schematron d'une règle se présente sous la forme suivante :

```
<rule context="house">
    <assert test="count(wall) = 4">A house should have four walls</assert>
    <report test="not(roof)">This house does not have a roof</report>
</rule>
```

Ensuite nous regroupons ces règles dans des "pattern" qui constitue l'élément clé d'un schematron. Et ensuite, nous pouvons regrouper plusieurs pattern dans un même schéma.

Exemple d'un schematron complet :

```
<sch:schema xmlns:sch="http://www.ascc.net/xml/schematron"
    icon="http://www.ascc.net/xml/resource/schematron/bilby.jpg" defaultPhase="
        built">
    <sch:p>This is an example schema for the <emph>Building Projects XML</emph
        > language.</sch:p>

    <sch:phase id="underConstruction">
        <sch:active pattern="construction"></sch:active>
        <sch:active pattern="admin"></sch:active>
    </sch:phase>

    <sch:phase id="built">
        <sch:active pattern="completed">completed</sch:active>
        <sch:active pattern="admin">admin</sch:active>
    </sch:phase>

    <sch:pattern name="Construction Checks" id="construction">
        <sch:p>Constraints which are applied during construction</sch:p>

        <sch:rule context="house">
            <sch:assert test="count(wall) = 4"> A house should have 4 walls</sch:
                assert>
            <sch:report test="not(roof)"> The house is incomplete, it still needs a
                roof</sch:report>
            <sch:assert test="builder"> An incomplete house must have a builder
                assigned to it</sch:assert>
            <sch:assert test="not(owner)"> An incomplete house cannot have an
                owner</sch:assert>
        </sch:rule>
    </sch:pattern>

    <sch:pattern name="Final Checks" id="completed">
    <sch:p>Constraints which are applied after construction</sch:p>

        <sch:rule context="house">
            <sch:assert test="count(wall) = 4"> A house should have 4 walls</sch:
                assert>
            <sch:report test="roof"> The house is incomplete, it still needs a roof</
                sch:report>
```

```
        <sch:assert test="owner"> An incomplete house must have an owner</sch
            :assert>
        <sch:assert test="not(builder)"> An incomplete house doesn't need a
            builder</sch:assert>
    </sch:rule>
</sch:pattern>

<sch:pattern name="Adminstration Checks" id="admin">
<sch:p>Adminstrative constraints which are <sch:emph> always</sch:emph>
    applied</sch:p>

    <sch:rule context="house">
        <sch:assert test="address">A house must have an address</sch:assert>
    </sch:rule>

    <sch:rule context="address">
        <sch:assert test="count(*) = count(street) + count(town) + count(
            postcode)"> An address may only include street, town and postcode
            elements. </sch:assert>

        <sch:assert test="street"> An address must include the street details</sch
            :assert>
        <sch:assert test="town"> An address must identify the town</sch:assert>
        <sch:assert test="postcode"> An address must have a postcode</sch:
            assert>
    </sch:rule>

    <sch:rule abstract="true" id="nameChecks">
        <sch:assert test="firstname"> A <name/> element must have a first name
            </sch:assert>
        <sch:assert test="lastname"> A <name/> element must have a last name
            </sch:assert>
    </sch:rule>

    <sch:rule context="builder">
        <sch:extends rule="nameChecks"></sch:extends>
        <sch:assert test=" certification "> A <name/> must be certified</sch:
            assert>
    </sch:rule>

    <sch:rule context="owner">
        <sch:extends rule="nameChecks"></sch:extends>
        <sch:assert test="telephone"> An <name/> must have a telephone</sch:
```

```
            assert>
        </sch:rule>

        <sch:rule context=" certification">
            <sch:assert test="@number"> Certification numbers must be recorded in
                the number attribute </sch:assert>
        </sch:rule>
    </sch:pattern>
</sch:schema>
```

Remarquons un dernier détail intéressant : la déclaration des phases. Grâce à elles, on peut activer sélectivement les pattern, et donc les règles à appliquer en fonction des besoins.

Comment fonctionne schematron ? Quand un schematron est défini, une feuille de style XSLT schematron va permettre de transformer ce schéma en feuille de style validante. Cette feuille de style sera alors appliquée sur les instances du document pour le valider à l'aide d'un compilateur XSLT. Il existe plusieurs feuilles de style pour schematron, chacune d'entre-elles étant dédiée à une fonctionnalité particulière.

3.3.2 TREX(Tree Regular Expressions for XML)

Le concept de base du langage TREX est le pattern. Une collection non ordonnée d'attributs, une sequence ordonnée déléments et les caractères doivent correspondre conjointement à un pattern selon un environnment donné. On appelle environnment un mapping de noms avec des patterns sous la possibilité d'avoir une référence nulle à un environnement parent. Le résultat d'un matching est booléen. Selon cette vision, un document XML est valide selon un TREX pattern, si une collection vide d'attributs et une séquence contenant juste l'élément "document" du document traité correspond à un environnement vide.

Selon le modèle de données, un élément se présente sous la forme d'un triplet `<name,attributes,children>` où :

- "name" est un nom étendu(expanded-name)
- "attributes" est une collection non ordonnée de zéro ou plus d'attributs
- "children" est une séquence ordonnée de zéro ou plus d'éléments ou de caractères

Un attribut est une paire `<name, value>` où :

- "name" est un nom étendu(expanded-name)
- "value" est une séquence de zéro ou plus de caractères

Un nom étendu(expanded-name) est une paire `<namespace URI, local name>` où :

- "namespace URI" est une chaine de caractères représentant une référence URI

– "local name" est une chaine de caractères représentant le "NCName", production de la recommandation XML

Voici un exemple de schéma TREX :

```
<grammar xmlns="http://www.thaiopensource.com/trex">
<define name="table" combine="replace">
  <element name="table">
    <ref name="table.attlist"/>
    <optional>
      <ref name="caption"/>
    </optional>
    <choice>
      <zeroOrMore>
        <ref name="col"/>
      </zeroOrMore>
      <zeroOrMore>
        <ref name="colgroup"/>
      </zeroOrMore>
    </choice>
    <choice>
      <group>
        <optional>
           <ref name="thead"/>
        </optional>
        <optional>
          <ref name="tfoot"/>
        </optional>
        <oneOrMore>
          <ref name="tbody"/>
        </oneOrMore>
      </group>
      <oneOrMore>
        <ref name="tr"/>
      </oneOrMore>
    </choice>
  </element>
</define>

<define name="table.attlist" combine="group">
  <optional>
    <attribute name="width">
      <ref name="Length.datatype"/>
    </attribute>
```

```
      </optional>
      <optional>
        <attribute name="border">
          <ref name="Pixels.datatype"/>
        </attribute>
      </optional>
      <ref name="frame.attrib"/>
      <ref name="rules.attrib"/>
      <optional>
        <attribute name="cellspacing">
          <ref name="Length.datatype"/>
        </attribute>
      </optional>
      <optional>
        <attribute name="cellpadding">
          <ref name="Length.datatype"/>
        </attribute>
      </optional>
    </define>

    <define name="col">
      <element name="col">
        <ref name="col.attlist"/>
      </element>
    </define>

    ....

</grammar>
```

3.3.3 Relax NG

Relax NG est un encore un autre langage développé dans le but de concevoir des schémas XML. Il a été conçu par le comité technique RELAX NG d'OASIS de façon à être une alternative au fameux schéma XML du W3C. Ses auteurs lui ont voulu une vocation de langage simple et facile à apprendre. RELAX NG a t dvelopp sous la direction de James Clark (créateur de TREX) et MURATA Makoto (créateur de RELAX), c'est donc un compromis entre ces deux langages.
Les propriétés de RELAX NG sont, aux dires des auteurs, les suivantes :

- La simplicit et la facilit d'apprentissage et d'implmentation
- Son absence d'impact sur les documents XML
- Un traitement homognes des lments et des attributs
- Un support sans restriction des contenus mixtes et non-ordonns

- Sa neutralit vis vis du systme de types de donnes qui lui permet d'tre utilis avec n'importe quel systme (tel que les types de donnes du schéma XML du W3C)

Grâce à cela, ils espèrent attirer et convaincre un bon nombre de développeurs vers cette alternative aux schémas XML du W3C. D'ailleurs James Clark a dcrit ce langage comme "une évolution conservatrice des idées des DTD XML et SGML ayant fait leurs preuves".

Voici un exemple de schéma Relax NG :

```
<grammar datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"
  ns="http://relaxng.org/ns/structure/1.0"
  xmlns="http://relaxng.org/ns/structure/1.0">   <start>
     <ref name="pattern"/>
  </start>   <define name="pattern">
    <choice>
      <element name="element">
        <choice>
          <attribute name="name">
            <data type="QName"/>
          </attribute>
          <ref name="open-name-class"/>
        </choice>
        <ref name="common-atts"/>
        <ref name="open-patterns"/>
      </element>
      <element name="attribute">
        <ref name="common-atts"/>
        <choice>
          <attribute name="name">
            <data type="QName"/>
          </attribute>
          <ref name="open-name-class"/>
        </choice>
        <interleave>
          <ref name="other"/>
          <optional>
            <ref name="pattern"/>
          </optional>
        </interleave>
      </element>
      <element name="group">
        <ref name="common-atts"/>
        <ref name="open-patterns"/>
      </element>
```

```
    <element name="interleave">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="choice">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="optional">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="zeroOrMore">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="oneOrMore">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="list">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="mixed">
      <ref name="common-atts"/>
      <ref name="open-patterns"/>
    </element>
    <element name="ref">
      <attribute name="name">
        <data type="NCName"/>
      </attribute>
      <ref name="common-atts"/>
    </element>
    <element name="parentRef">
      <attribute name="name">
        <data type="NCName"/>
      </attribute>
      <ref name="common-atts"/>
    </element>
        ...

</grammar>
```

3.4 La validation

Comme nous l'avons dit en début de chapitre, l'objectif des DTD et des schémas XML est de valider des documents XML. Nous allons apporter ici quelques petites précisions sur cette notion de validation.
Comme nous le verrons au chapitre suivant, pour traiter un document XML, il faut utiliser un outil qui s'appelle "parseur". Sans réellement entrer dans les détails maintenant, il faut savoir qu'il en existe deux grandes familles : les validants et ceux qui ne le sont pas. Comme vous l'avez déjà certainement compris, nous allons aborder ici l'aspect validation en nous basant sur un petit parseur validant : RXP.
Tout d'abord, il faut faire la différence entre valider un document XML et vérifier si un document XML est bien formé(well-formed document).
Un document est bien formé si il respecte les conditions suivantes :

- Le document commence avec une déclaration XML du genre
  "`<?xml version="1.0" standalone="yes"?>`"
- Chaque balise de départ est vide (`<tag/>`) ou possède sa balise de fin correspondante
- L'élément racine doit être unique. La racine est l'élément qui contient tous les autres. Seul les commentaires, les espaces blancs et les instructions de traitements sont autorisés après la balise de fermeture de la racine
- Tous les éléments s'imbriquent l'un dans l'autre correctement
- Toutes les valeurs des attributs sont encapsulées dans dans quotes(simples ou doubles)

La validation d'un document consiste à vérifier si un document est conforme à la structure d'une DTD ou d'un schéma XML. Cela signifie entre autre que l'on va vérifier si le vocabulaire des balises est respecté, si les valeurs des éléments et des attributs respectent les contraintes qu'on leur a imposés(énumérations, bornes numériques,...) et si le document est bien formé évidemment.

Quelques messages relatifs à des erreurs courantes :

- Si une valeur n'est pas reprise dans une énumération :
  ```
  Warning: In the attribute alixdTraceLevel of element alixd, none1 is not
           one of the allowed values in unnamed entity at line 4 char 31 of
           file:/home/raptor/Memoire/rxp/dtd/OACconfig.xml
  ```
- Utilisation d'une balise incorrecte :
  ```
  Warning: Content model for OACconfig does not allow element alixd1 here
           in unnamed entity at line 4 char 8 of file:
           /home/raptor/Memoire/rxp/dtd/OACconfig.xml
  ```
- Faute de frappe dans une balise fermante (remarquons l'accumulation des erreurs) :
  ```
  Warning: Start tag for undeclared element alixdSourcesEntrys in unnamed
           entity at line 5 char 20 of file:
           /home/raptor/Memoire/rxp/dtd/OACconfig.xml
  ```

```
Warning: Content model for OACconfig does not allow element
         alixdSourcesEntrys here in unnamed entity
         at line 5 char 20 of file:
         /home/raptor/Memoire/rxp/dtd/OACconfig.xml

Error: Mismatched end tag: expected </alixdSourcesEntrys>, got
       </alixdSourcesEntry> in unnamed entity at line 9 char 20
       of file:/home/raptor/Memoire/rxp/dtd/OACconfig.xml
```

Dans le cas où le document serait valide, aucun message n'apparait bien sûr à l'écran. Mais regardons le dernier de nos trois exemples, il nous prouve bien qu'une validation vérifie aussi qu'un document est bien formé en plus de vérifier les contraintes du schéma.

3.5 Où en sommes-nous ?

Nous venons d'atteindre avec ce troisième chapitre le premier de nos objectifs, à savoir la construction d'une DTD pour chacun des agents. Même si par la suite, nous sommes passés aux schémas XML, cela ne change rien au fait que nous sommes maintenant capables d'écrire des configurations au format XML.
Rappelons-nous aussi que le traitement envisagé pour ces feuilles XML était une conversion au format que nous avons appelé "classique", c'est-à-dire au format présenté au chapitre deux. Lors du dernier paragraphe nous avons introduit le terme "parseur" sans savoir encore vraiment de quoi il retournait. Dans la première partie du chapitre suivant, nous allons faire connaissance avec ces parseurs qui seront les outils utilisés dans le but d'obtenir la transformation souhaitée. Une seconde partie complémentaire abordera aussi un peu l'aspect programmation, en démontrant comment il fonctionne concrètement.

4 Les parseurs XML

Dans le première partie de ce chapitre, nous aborderons les outils courants de traitement de fichiers XML : les parseurs Expat, DOM et SAX. Ce sont d'ailleurs ces derniers qui utiliseront les DTD et les schémas XML pour valider les documents qu'ils sont censés traiter.
Une seconde partie plus pratique viendra alors complémentariser les différents points de notre présentation générale.

4.1 Présentation générale

Nous allons décrire dans cette section les différentes spécifications techniques des parseurs courants que l'on peut rencontrer. Nous en avons en fait trois : Expat, Sax et DOM.

4.1.1 Parseur Expat

Expat n'est pas un parseur en lui-même. C'est une librairie écrite en C par James Clark, qui fut le directeur technique du groupe de travail XML au sein du W3, lui même étant à la base des spécifications XML. Expat est actuellement utilisé dans plusieurs projets dont le parseur XML du projet Mozilla ainsi que dans le module Perl XML : :Parser. Expat ne peut par contre être utilisé pour valider des documents.

Quelques précisions Expat est un "stream-oriented parser", c'est-à-dire que l'on va associer des fonctions de "callback"(handling functions) au parseur qui seront alimentées lors du traitement du document. Lors qu'un événement sera reconnu par le parseur, il fera appel à la fonction de callback affectée à cet événement pour exécuter le traitement adéquat. De plus, lors du parsing, le document est découpé en plusieurs morceaux, ce qui apporte deux avantages majeurs :

1. Le parsing peut commencer avant même que l'ensemble du document ne soit lu : gain de temps
2. Ce système évite que les grands documents ne saturent la mémoire et n'en affectent les performances

Expat possède un certain nombre de fonctions callback et d'options disponibles, mais plus vous décidez d'en rajouter dans votre Handler, plus Expat risque de voir ses performances chuter. Aussi il est préférable d'en limiter le nombre. Sachez pour cela que quatre fonctions vont vous permettre de traiter plus de 80% des cas :

- XML_ParserCreate : créer un nouveau parseur
- XML_SetElementHandler : fonction callback pour les balises de début et fin d'élément
- XML_SetCharacterDataHandler : fonction callback pour le traitement du text
- XML_Parse : fournit un buffer contenant une partie du document au parseur

Notons au passage que Expat gère les espaces de nommages et qu'il reconnait quatre types

d'encodage : UTF-8, UTF-16, ISO-8859-1, US-ASCII.

Expat a la possibilité de parser les DTD. Pour ce faire, il suffit de déclarer la macro "XML_DTD" lors de la compilation de Expat. Ensuite, pour utiliser cette compétence, il suffit d'appeler "XML_SetParamEntityParsing" avant de créer une instance du parseur dans notre programme. Les arguments possibles pour cette fonction sont :

- XML_PARAM_ENTITY_PARSING_NEVER : ne pas traiter les entités utilisées comme paramètre ou les sous-ensembles externes
- XML_PARAM_ENTITY_PARSING_UNLESS_STANDALONE : traiter les entités utilisées comme paramètre ou les sous-ensembles externes sauf si la close "standalone" est positionnée sur "yes" dans la déclaration XML
- XML_PARAM_ENTITY_PARSING_ALWAYS : toujours traiter les entités utilisées comme paramètre ou les sous-ensembles externes

Comment font les "stream-oriented parser" pour parcourir des documents hiérarchisés ? La question qui se pose ici c'est : "Comment associer un text avec l'élément auquel il se rapporte ?". L'idée est très simple : on utilise un système de pile. De manière pratique, lorsque l'on rencontre une balise de début d'élément, on empile toutes les informations qui lui sont relatives comme ses attributs et sa valeur ; et lorsque l'on rencontre la balise de fin, on dépile les informations de cet élément.

Voilà, nous venons de faire un tour rapide des éléments essentiels qui composent ce parseur. Donner plus de détails nous entrainerait dans un dédale de spécifications techniques qui sortirait du cadre de ce travail.

4.1.2 Parseur SAX

SAX ou Simple API for XML est une API basée sur le modèle événementiel, en d'autres termes, chaque fois qu'un événement est reconnu par SAX au cours du traitement du document, il fait appel à la fonction concernée dans le Handler approprié.

Pour créer une instance de parseur SAX, on utilise le "SAX Parser Factory", et les différents éléments de SAX sont les suivants :

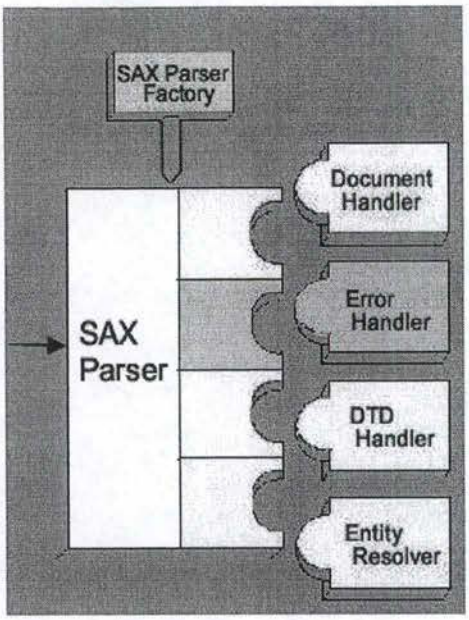

FIG. 11 – Schéma des composants du parseur SAX

Comme on peut le voir sur ce schéma, SAX se compose de quatre types de Handler différents :

- EntityResolver : cet handler est invoqué lorsque le parseur se trouve face à des données identifées par une URL
- Document Handler : cet handler est invoqué lorsque dans le document, il se trouve en face de quatre événements : startDocument, endDocument, startElement, et endElement
- DTDHandler : cet handler est invoqué pour traiter les définitions dans les DTD
- Error Handler : cet Handler permet de gérer les erreurs rencontrées lors du parsing, quelles soient fatales ou de simples avertissements

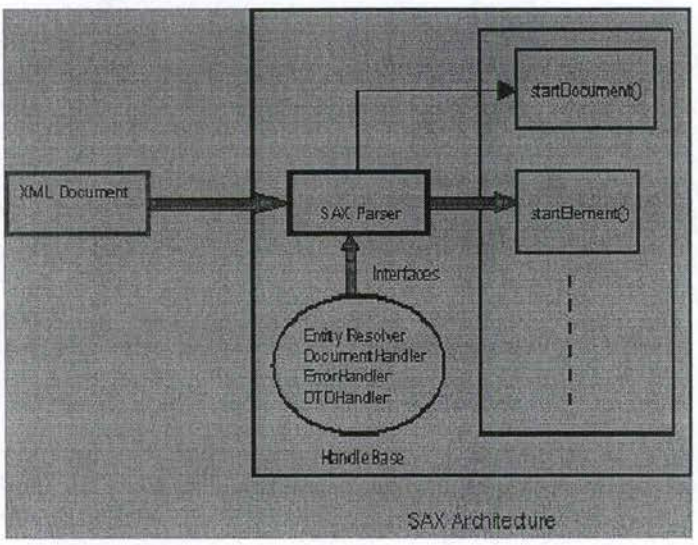

FIG. 12 – Schéma de l'architecture de SAX

SAX 1.0 et 2.0 Le but ici n'est pas de donner une liste exhaustives des détails de spécifications de chacunes des version. Sachez seulement que la relation liant les deux versions se situe au niveau d'une évolution de SAX. En effet, la version 2.0 permet de corriger certains bugs mais aussi d'ajouter de nouvelles API. La version 2.0 permet entre autre de gérer les espaces de noms et de traiter les contenus de DTD. La permière version fut officialisée en mai 1998, tandis que la seconde en mai 2000.

4.1.3 Parseur DOM

DOM ou Document Object Model est le résultat d'une recommandation du consortium W3C. DOM n'est pas événementiel comme SAX. En fait lors du parsing, DOM construit en mémoire une structure arborescente représentant la structure du document traité. Le résultat pouvant ensuite être exploité selon les besoins du développeur.
Pour créer une instance de parseur DOM, on utilise le "Document Builder Factory", et les différents éléments de DOM sont les suivants :

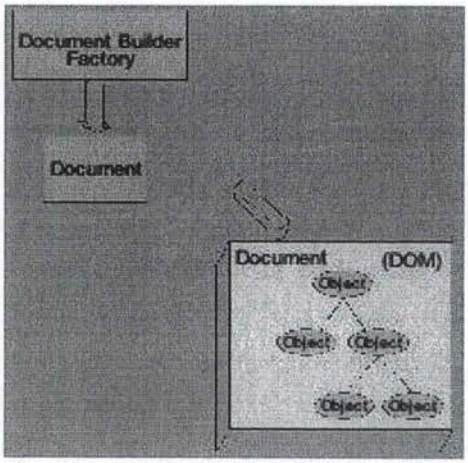

FIG. 13 – Schéma des composants du parseur DOM

A noter que l'élément "Document" représente le document résultat de l'opération de parsing. L'architecture, quant à elle, se présente sous cette forme :

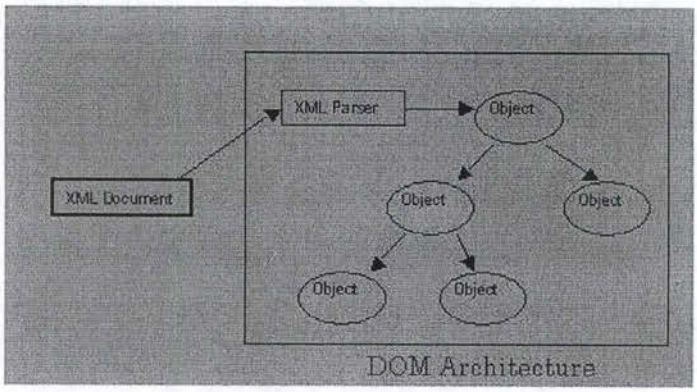

FIG. 14 – Schéma de l'architecture de DOM

4.1.4 Les différents niveaux de spécification du modèle DOM

Nous n'allons pas exposer ici les spécifications en détails, cela reviendrait à faire du recopiage. Néanmoins, ceux qui souhaiteraient plus de précisions sont invités de consulter le site du W3C à l'adresse suivante : "http ://www.3c.org/TR". Nous allons donner ici un aperçu des différents documents que l'on peut trouver concernant la spécification DOM.

DOM level 1 core Dans cette partie, on définit les différents objets ainsi que les différentes interfaces disponibles pour manipuler les éléments XML. Il s'agit de spécifier le coeur du système DOM. On y retrouve d'ailleurs les éléments suivants :

1. Le modèle de structure DOM : définitions des types de noeuds (Element,Attribute, Text,Comment,etc), les méthodes pour la manipuler (getChildNodes,getParent,...) ainsi que les différentes interfaces comme "NodeList" ou bien "NamedNodeMap"
2. La gestion de la mémoire
3. Les conventions pour les espaces de noms
4. Les types d'exceptions et leur gestion
5. La sensibilité à la casse

DOM level 2 Core A ce niveau, il n'y a pas grand chose à dire si ce n'est que le noyau s'est vu fourni de nouvelles API. En fait certaines interfaces ont été complètée avec de nouveaux attributs ou/et de nouvelles méthodes.

DOM level 2 Events L'objectif principal de cette spécification sera de fournir un système de gestion générique d'événements permettant de concevoir des gestionnaires(event handlers), ceux-ci décrivant les flux d'événements à travers une structure arborescente, et fournissant un contexte informationnel de base pour chaque événement. En outre, la spécification fournira aussi des modules et des événements standards pour les interfaces de contrôle utilisateur, ainsi que des notifications de modifications de document, incluant de ce fait des informations contextuelles définies pour chacun de ces modules.

DOM level 2 Style Il sera question ici de spécifier des interfaces permettant à DOM de représenter tous les types de feuilles de style et ainsi de les manipuler.

DOM level 2 Traversal and Range Specification Cette spécification définit les interfaces et méthodes permettant de manipuler les objets "TreeWalker" et "NodeIterator" utilisés pour parcourir des sous arbres de documents construits à partir de vues(Views). Ces fonctionalités sont optionnelles.

DOM level 2 View Cette spécification détermine les méthodes et interfaces permettant de manipuler les vues. Ces dernières étant obtenues par application d'une feuille de style sur le document.
Les vues peuvent être statiques(elle représente alors l'état du document à sa création) ou

dynamiques(les vues sont mises à jour au moment ou des changements se produisent). Cet aspect n'est pas pris en considération ici.

DOM level 3 Core De manière analogue au level 2, DOM level 3 Core apporte des corrections au niveau des spécifications antérieures.

DOM level 3 Load and Save Cette spécification définit un ensemble d'interface pour sauver et charger les objets définit dans le DOM level 3 Core. Ce qui permettra aux développeurs de sauver ou charger des contenus XML à partir de n'importe quelles applications.

DOM level 3 Validation Cette spécification définit des interfaces et des méthodes pouvant être util à la validation. En d'autres termes, ce module fournit des APIs capables d'obtenir des informations concernant les documents XML et les grammaires associées(comme les DTD et les schémas XML).

4.1.5 Petite comparaison entre les parseurs SAX et DOM

Nous avons maintenant fait connaissance avec SAX et DOM. Comme nous venons de le voir, chaque parseur possède ses propres spécificités dont il faudra tenir compte lorsque nous seront amenés à choisir une méthode plutôt que l'autre. A cet effet, nous nous proposons ici de faire un petit récapitulatif à titre de suggestion, afin d'aider le lecteur à se forger une opinion sur les critères qui peuvent déterminer le choix d'un parseur par rapport à l'autre.

Taille du document

Pour le parseur SAX, la taille des documents importe peu. En effet, un parseur SAX permet de traiter un document même si ce dernier n'est pas encore tout à fait chargé en mémoire. D'un autre côté, il est également possible de limiter l'accessibilité des données à un sous-ensemble en évitant, par exemple, de traiter certains événements dans le "Handler" du parseur.
Par contre, le parseur DOM est très sensible à ce critère. Pour donner un ordre d'idée, stocker 100kb de document en mémoire avec DOM consomme plus ou moins 1MB de cette mémoire. Cela influera donc beaucoup sur les performances.
Mais, comme dans le cas du parseur SAX, il est également possible de restreindre l'accessibilité des noeuds de l'arbre en utilisant les vues (DOM level 2 View). Ces dernières construisent en mémoire des sous-arbres qui auront pour but de limiter le parcours de l'arborescence aux données pertinentes dans le cadre d'une tâche particulière.

Performances

SAX est simple et rapide car il permet de restituer toutes les données recherchées en une seule passe. Malheureusement, il n'autorise pas les accès "random"(aléatoires) au document, puisque ce dernier n'est pas obligé d'être totalement en mémoire pour être lu. Ce

qui a pour conséquence qu'on ne peut tirer parti des attributs de type ID et IDREF ou accomplir des recherches complexes. DOM, lui, accède aux données à partir d'une organisation en arbre. D'un premier point de vue, cela complexifie un peu la lecture, mais d'un autre côté, il peut profiter très facilement des informations dont il dispose sur la structure du document, comme obtenir le parent ou l'ensemble des fils d'un élément donné.

Validation

Bien qu'au début, le traitement des contenus de DTD avec SAX n'était pas possible, cette limitation a été levée avec la version 2.0. De ce fait, les DTD, de même que la technologie naissance des schémas XML, sont maintenant parfaitement supportées par SAX et DOM. Ajoutons aussi simplement que l'opération de validation avec SAX se fait parallèlement au traitement, tandis qu'avec DOM, elle se fera pendant le chargement de l'arborescence en mémoire. Cette remarque a pour but de faire observer que, dans le second cas, cela permettra d'éviter le lancement du traitement si le document n'est pas valide selon les critères de conformité imposés par la DTD ou le schéma XML associé.

Méthodes d'accès

SAX ne peut accèder aux documents qu'en lecture seule, contrairement à DOM, qui lui permet de modifier et d'effacer des noeuds dans la structure en mémoire.

Flexibilité

SAX est avantageux pour construire sa propre structure de donnée. Imaginons, que vous deviez mettre en oeuvre une application qui aura pour but de construire une structure de données "haut niveau" en termes de livres,auteurs,... plutôt qu'en termes d'éléments et d'attributs par exemple. En outre prenons l'hypothèse que les données ne proviennent pas exclusivement d'un fichier XML donné. Il n'est pas avantageux de contruire un document DOM "bas niveau" en mémoire et ensuite de devoir le reconstruire à chaque changement de structutre induites par les données externes au document. Par contre, il est beaucoup plus simple de traiter les événements quand ils surviennent et d'appliquer des changements incrémentaux au modèle

Applications Web

Les navigateurs web ne supporte pas SAX. A l'heure actuelle, et bien que de nombreux parseurs supporte cette interface, celle-ci n'a pas encore été incluse dans aucun navigateur. Cette limitation peut être contournée si on utilise un parseur SAX-compatible dans une applet Java. Malheureusement, cette solution sera rébarbative pour les gens disposant d'une connexion à faible débit. DOM, quant à lui, commence a être supporté par certains navigateurs comme Netscape ou Mozilla.

4.2 Langages de programmation et parseurs

Après avoir présenté les différents parseurs dans la section précédente, il faudrait maintenant voir comment on peut en tirer parti de manière pratique. Dans cette seconde partie, nous allons faire le tour des différents librairies disponibles dans trois langage classique que sont le C++, le perl et le java.

4.3 Le langage C++

Le groupe Apache à l'origine du serveur web du même nom a mis au point une librairie C++ complète permettant de concevoir des applications capable de traiter les feuilles XML : le projet Xerces-C++ ; la dernière version actuelle étant la 2.0.0 Cette librairie permet également de valider les feuilles non seulement à l'aide de DTD mais aussi avec des schémas XML du W3C, qui maintenant sont totalement pris en charge. Ces librairies peuvent être téléchargées sur le site du "Apache XML project" à l'adresse http ://xml.apache.org.
Pour être précis, Xerces-C++ se conforme aux normes suivantes :

- XML 1.0 (Second Edition), W3C Recommendation of October 6, 2000
- DOM Level 1 Specification, W3C Recommendation of October 1, 1998
- DOM Level 2 Core Specification, W3C Recommendation of November 13, 2000
- DOM Level 2 Traversal and Range Specification,W3C Recommendation of November 13, 2000
- SAX 1.0 and SAX 2.0
- Namespaces in XML, W3C Recommendation of January 14, 1999
- XML Schema Part 1 : Structure, W3C Recommendation 2 May 2001
- XML Schema Part 2 : Datatypes, W3C Recommendation 2 May 2001

Malheureusement, le "DOM Level 3.0 Core Specification" n'est que partiellement implémenté.

4.3.1 Exemple de parseur SAX

L'exemple ci-dessous ne fait rien de particulier. Il s'agit de présenter la structure minimale nécessaire pour traiter un document XML. Afin de l'utiliser, il suffira de complèter les évéments du "Handler" par les traitements adéquats. Dans notre exemple le parseur et le Handler seront disposés dans deux fichiers différents SAXParser.cpp et MyHandler.cpp et la validation se fait à l'aide d'un schéma XML du W3C.

Code du fichier SAXParser.cpp

```
#include <xercesc/util/PlatformUtils.hpp>
#include <xercesc/util/TransService.hpp>
#include <xercesc/sax2/SAX2XMLReader.hpp>
#include <xercesc/sax2/XMLReaderFactory.hpp>
#include "SAXParser.hpp"

static XMLFormatter::UnRepFlags unRepFlags = XMLFormatter::UnRep_CharRef;
static char* xmlFile = 0;

int main(int argC, char* argV[])
    {
    SAX2XMLReader* parser;
    int errorCount = 0;
    xmlFile = argV[1];
    try
        {
        XMLPlatformUtils::Initialize();
        parser = XMLReaderFactory::createXMLReader();
        parser->setFeature(XMLUni::fgSAX2CoreValidation, true);
        parser->setFeature(XMLUni::fgXercesDynamic, true);
        parser->setFeature(XMLUni::fgSAX2CoreNameSpaces,true);
        parser->setFeature(XMLUni::fgXercesSchema,true);
        parser->setFeature(XMLUni::fgXercesSchemaFullChecking,true);
        parser->setFeature(XMLUni::fgSAX2CoreNameSpacePrefixes,false);
        MySaxHandler handler("LATIN1", unRepFlags, false);
        parser->setContentHandler(&handler);
        parser->setErrorHandler(&handler);
        parser->parse(xmlFile);
        errorCount = parser->getErrorCount();
        delete parser;
        XMLPlatformUtils::Terminate();
        }

    catch (const XMLException& toCatch)
        {
        cerr << "\nAn error occurred\n  Error: "
             << StrX(toCatch.getMessage())
             << "\n" << endl;
        delete parser;
        XMLPlatformUtils::Terminate();
        return 4;
```

```
        }

    if (errorCount > 0)
        return 4;
    else
        return 0;
    }
```

Code du fichier MySaxHandler.cpp

```
#include <xercesc/util/XMLUniDefs.hpp>
#include <xercesc/sax2/Attributes.hpp>
#include "SAXParser.hpp"

static const XMLCh gEndElement[] = { chOpenAngle, chForwardSlash, chNull };
static const XMLCh gEndPI[] = { chQuestion, chCloseAngle, chNull };
static const XMLCh gStartPI[] = { chOpenAngle, chQuestion, chNull };
static const XMLCh gXMLDecl1[] =
    {
        chOpenAngle, chQuestion, chLatin_x, chLatin_m, chLatin_l
    ,   chSpace, chLatin_v, chLatin_e, chLatin_r, chLatin_s, chLatin_i
    ,   chLatin_o, chLatin_n, chEqual, chDoubleQuote, chDigit_1, chPeriod
    ,   chDigit_0, chDoubleQuote, chSpace, chLatin_e, chLatin_n, chLatin_c
    ,   chLatin_o, chLatin_d, chLatin_i, chLatin_n, chLatin_g, chEqual
    ,   chDoubleQuote, chNull
    };

static const XMLCh gXMLDecl2[] ={chDoubleQuote,chQuestion,chCloseAngle,chLF,
    chNull};

MySaxHandler::MySaxHandler(const char* const encodingName,const
    XMLFormatter::UnRepFlags unRepFlags,const bool expandNamespaces):
    fFormatter(encodingName,this,XMLFormatter::NoEscapes,unRepFlags),fExpandNS
        ( expandNamespaces )
        {fFormatter << gXMLDecl1 << fFormatter.getEncodingName() <<
            gXMLDecl2;}

MySaxHandler::~MySaxHandler()
{}

void MySaxHandler::writeChars(const XMLByte* const toWrite)
{}
```

```
void MySaxHandler::writeChars(const XMLByte* const toWrite,const unsigned int
    count,XMLFormatter* const formatter)
{}

void MySaxHandler::error(const SAXParseException& e)
{
    cerr << "\nError at file " << StrX(e.getSystemId())
         << ", line " << e.getLineNumber()
         << ", char " << e.getColumnNumber()
         << "\n  Message: " << StrX(e.getMessage())
         << endl;
}

void MySaxHandler::fatalError(const SAXParseException& e)
{
    cerr << "\nFatal Error at file " << StrX(e.getSystemId())
         << ", line " << e.getLineNumber()
         << ", char " << e.getColumnNumber()
         << "\n  Message: " << StrX(e.getMessage())
         << endl;
}

void MySaxHandler::warning(const SAXParseException& e)
{
    cerr << "\nWarning at file " << StrX(e.getSystemId())
         << ", line " << e.getLineNumber()
         << ", char " << e.getColumnNumber()
         << "\n  Message: " << StrX(e.getMessage())
         << endl;
}

void MySaxHandler::unparsedEntityDecl(const XMLCh* const name,const XMLCh*
    const publicId,const XMLCh* const systemId,const XMLCh* const
    notationName)
{}

void MySaxHandler::notationDecl(const XMLCh* const name, const XMLCh* const
    publicId,const XMLCh* const systemId)
{}

void MySaxHandler::characters(const XMLCh* const chars,const unsigned int
    length)
{}
```

```
void MySaxHandler::endDocument()
{}

void MySaxHandler::endElement(const XMLCh* const uri,const XMLCh* const
    localname,const XMLCh* const qname)
{}

void MySaxHandler::ignorableWhitespace(const XMLCh* const chars,const
    unsigned int length)
{}

void MySaxHandler::processingInstruction(const XMLCh* const target,const XMLCh
    * const data)
{}

void MySaxHandler::startDocument()
{}

void MySaxHandler::startElement(const XMLCh* const uri,const XMLCh* const
    localname,const XMLCh* const qname,const Attributes& attributes)
{}
```

4.3.2 Exemple de parseur DOM

De même que l'exemple précédent, le code ci-dessous reprend les lignes essentielles qui valident le document XML traité à l'aide d'un schéma XML, et construit en mémoire l'arbre DOM de ce même document.

```
#include <xercesc/util/PlatformUtils.hpp>
#include <xercesc/parsers/AbstractDOMParser.hpp>
#include <xercesc/dom/DOMImplementation.hpp>
#include <xercesc/dom/DOMImplementationLS.hpp>
#include <xercesc/dom/DOMImplementationRegistry.hpp>
#include <xercesc/dom/DOMBuilder.hpp>
#include <xercesc/dom/DOMException.hpp>
#include <xercesc/dom/DOMDocument.hpp>
#include <xercesc/dom/DOMNodeList.hpp>
#include <xercesc/dom/DOMError.hpp>
#include <xercesc/dom/DOMLocator.hpp>
#include "DOMParser.hpp"
#include <string.h>
#include <stdlib.h>
#include <fstream.h>
```

```
int main(int argC, char* argV[])
    {
    const char* xmlFile = 0;
    bool errorOccurred = false;
    DOMBuilder *parser;
    MyDOMErrorHandler errorHandler;
    DOMDocument *doc = 0;
    try
        {
        XMLPlatformUtils::Initialize();
        static const XMLCh gLS[] = { chLatin_L, chLatin_S, chNull };
        DOMImplementation *impl = DOMImplementationRegistry::
            getDOMImplementation(gLS);
        parser = ((DOMImplementationLS*)impl)->createDOMBuilder(
            DOMImplementationLS::MODE_SYNCHRONOUS, 0);
        xmlFile=argV[1];
        parser->setFeature(XMLUni::fgDOMValidateIfSchema,true);
        parser->setFeature(XMLUni::fgDOMDatatypeNormalization,true);
        parser->setFeature(XMLUni::fgDOMNamespaces,true);
        parser->setFeature(XMLUni::fgXercesSchema,true);
        parser->setFeature(XMLUni::fgXercesSchemaFullChecking,true);
        parser->setErrorHandler(&errorHandler);
        errorHandler.resetErrors();
        parser->resetDocumentPool();
        doc = parser->parseURI(xmlFile);
        }
        catch (const XMLException& toCatch)
            {
            cerr << "\nError during parsing: '" << xmlFile << "'\n"
                 << "Exception message is:  \n"
                 << StrX(toCatch.getMessage()) << "\n" << endl;
            errorOccurred = true;
            }
        catch (const DOMException& toCatch)
            {
            cerr << "\nDOM Error during parsing: '" << xmlFile << "'\n"
                 << "DOMException code is:  \n"
                 << toCatch.code << "\n" << endl;
            errorOccurred = true;
            }
        catch (...)
            {
```

```
            cerr << "\nUnexpected exception during parsing: '" << xmlFile << "'\n"
                ;
            errorOccurred = true;
            }

        if (!errorHandler.getSawErrors())
            {
            cout << XMLString::transcode(doc->getElementsByTagName(XMLString
                ::transcode("alixd"))->item(0)->getChildNodes()->item(1)->
                getNodeName())
                 << "  =  "
                 <<XMLString::transcode(doc->getElementsByTagName(XMLString::
                     transcode("alixd"))->item(0)->getChildNodes()->item(1)->
                     getFirstChild()->getNodeValue())
                 <<endl;
            }

    parser->release();
    XMLPlatformUtils::Terminate();
    if (errorOccurred)
        return 4;
    else
        return 0;
    }

MyDOMErrorHandler::MyDOMErrorHandler() :

    fSawErrors(false)
        {}

    MyDOMErrorHandler::~MyDOMErrorHandler()
        {}

    bool MyDOMErrorHandler::handleError(const DOMError& domError)
        {
        fSawErrors = true;
        if (domError.getSeverity() == DOMError::DOM_SEVERITY_WARNING)
            cerr << "\nWarning at file ";
        else if (domError.getSeverity() == DOMError::DOM_SEVERITY_ERROR)
            cerr << "\nError at file ";
        else
            cerr << "\nFatal Error at file ";
```

```
        cerr << StrX(domError.getLocation()->getURI())
             << ",␣line␣" << domError.getLocation()->getLineNumber()
             << ",␣char␣" << domError.getLocation()->getColumnNumber()
             << "\n␣␣Message:␣" << StrX(domError.getMessage()) << endl;

        return false;
        }

    void MyDOMErrorHandler::resetErrors()
        {fSawErrors = false;}
```

4.4 Le langage scriptural Perl

4.4.1 Qu'est-ce que Perl ?

Le langage Perl ou Practical Extraction and Report Language fut crée en 1986 par Larry Wall. Ce langage se caractérise surtout par le fait qu'il est interprété, ce qui signifie qu'il sera plus lent qu'un programme compilé et nécéssitera un interpréteur Perl pour l'exécution. La popularité de Perl est issu de plusieurs facteurs comme sa portabilité, sa robustesse, sa gratuité et sa simplicité.
A l'origine Perl était destiné à la manipulation de fichier, de texte et de processus dans un environnement système UNIX.

Aujourd'hui, Perl s'est étendu de manière spectaculaire et les modules disponibles couvrent un ensemble de domaines très diversifiés allant de l'analyse de fichier HTML jusqu'à l'interfaçage avec des systèmes de gestion de base données ou des annuaires comme LDAP. Les modules Perl sont majoritairement disponibles sur le site du CPAN(http ://www.cpan.org).

Par contre, Perl n'est pas conçu pour écrire des interfaces graphiques interactives(mais il existe un module tkperl qui le permet), ni même de traiter des calculs scientifiques, notamment à cause de ses faibles performances à l'exécution(Perl est interprété rappelons le).

4.4.2 Perl et XML

Comme C++ et java, Perl permet de construire des applications utilisant des documents XML comme données. De par le fait que le nombre de modules disponibles pour XML est assez grand, nous réduirons notre analyse sur les quatre parseurs suivants :

- $\rightsquigarrow$ XML : :Parseur et XML : :Parser : :Expat $\longrightarrow$ Parseur XML basé sur expat
- $\rightsquigarrow$ XML : :Checker : :Parser $\longrightarrow$ Parseur "valideur" (DTD)
- $\rightsquigarrow$ XML : :Parser : :PerlSAX $\longrightarrow$ Parseur SAX
- $\rightsquigarrow$ XML : :DOM $\longrightarrow$ Parseur DOM

XML : :Parseur et XML : :Parser : :Expat La différence entre ces deux parseurs est infimes car le XML : :Parser est une interface au dessus du module XML : :Parseur : :Expat, qui est un module d'accès bas niveau pour le parseur de James Clark. En fait chaque appel à une méthode de parsing du XML : :Parser génère une nouvelle instance du parseur bas niveau. Pour utiliser ce module, les librairies du programme expat doivent être présent sur le système. La meilleure façon de le faire est d'installer le logiciel lui-même. Ce parseur permet de parser les documents XML mais pas de les valider. Pour ce donner une meilleure idée de l'ulisation que l'on peut en faire, voici quelques événements manipulables par le Handler de ce parseur :

- Init : début de document
- Final : fin de document
- Start : tag de début d'élément avec les attributs qui lui sont assignés
- End : tag de fin d'élément
- Notation : déclaration d'une notation
- Entity : déclaration d'une entité
- Element : déclaration d'un élément
- Attlist : déclaration d'une liste d'attributs
- Doctype : lecture de la déclaration du "DOCTYPE"

XML : :Checker : :Parser Il n'y a pas grand chose à en dire, si ce n'est qu'il s'agit d'une extension du parseur présenté précédemment qui permet de valider des documents à l'aide de DTD uniquement.

XML : :Parser : :PerlSAX PerlSAX est un parseur qui ne travaille actuellement qu'avec la version 1.0 du modèle SAX. Pour un support de la version 2.0, il faudra se tourner vers un autre module : Orchard : :SAXDriver : :Expat.
Le module offre ici la possibilité de mettre en oeuvre deux types différents de Handler, un pour les documents qui comporte les événements classiques du SAX 1.0, et l'autre (DTDHandler) pour traiter les différents types de déclarations(entité, élément, liste d'attributs,...)

XML : :DOM Le dernier de la liste va nous permettre de travailler avec le parseur "DOM Level 1 compliant document structure". Que dire si ce n'est qu'il comporte toutes les méthodes recommandées par les spécifications.
Un exemple exploitant ces librairies pourra être trouvé en annexe "B". Mais il existe aussi une autre librairie Perl pour valider des documents XML à partir de schémas XML du W3C : XML : :Xerces. Ces modules ne sont rien d'autre qu'une interface Perl exploitant les librairies Xerces du groupe Apache. Voici ci-dessous deux petits exemples pour SAX et DOM respectivement :

Code du fichier SAXParser.pl

```
use XML::Xerces;

package MyDocumentHandler;
use strict;
use vars qw(@ISA);
@ISA = qw(XML::Xerces::PerlDocumentHandler);

sub start_element {}

sub end_element {}

sub characters {}

sub ignorable_whitespace {}

package main;

my $file = $ARGV[0];
-f $file or die "File ' $file ' does not exist!\n";

my $parser = XML::Xerces::SAXParser->new();
$parser->setValidationScheme (1);
$parser->setDoNamespaces (1);
$parser->setDoSchema (1);
my $ERROR_HANDLER = XML::Xerces::PerlErrorHandler->new();
$parser->setErrorHandler($ERROR_HANDLER);
my $DOCUMENT_HANDLER = MyDocumentHandler->new();
$parser->setDocumentHandler($DOCUMENT_HANDLER);
eval {
  $parser->parse (XML::Xerces::LocalFileInputSource->new($file));
};
if ($@) {
  if (ref $@) {
    die $@->getMessage();
  } else {
    die $@;
  }
}

print "Validation OK :-)\n";
```

Code du fichier DOMParser.pl

```
use XML::Xerces;
use XML::Xerces::DOMParse;

my $file = $ARGV[0];
-f $file  or die "File ' $file ' does not exist!\n";

my $parser = XML::Xerces::DOMParser->new();
$parser->setValidationScheme (1);
$parser->setDoNamespaces (1);
$parser->setCreateEntityReferenceNodes(1);
$parser->setDoSchema (1);

my $ERROR_HANDLER = XML::Xerces::PerlErrorHandler->new();
$parser->setErrorHandler($ERROR_HANDLER);
eval {
  $parser->parse (XML::Xerces::LocalFileInputSource->new($file));
};
if ($@) {
  if (ref $@) {
    die $@->getMessage();
  } else {
    die $@;
  }
}
my $doc = $parser->getDocument ();

print "Validation OK :-)\n";
```

4.5 Le langage java

Le langage Java de chez Sun offre maintenant une suite d'API pour traiter les documents XML : Java™ XML Pack Summer 02 Bundle Release Notes.
Cette suite se décompose en quatre modules distincts : JAXM, JAXP, JAXR, JAX-RPC.

Le module JAXM(Java API XML Messaging) JAXM permet décrire des applications Java capables d'envoyer et de recevoir des documents XML sous forme de messages, en utilisant pour ce faire le protocole SOAP(Simple Object Access Protocol) 1.1. De cette manière, le développeur peut se concentrer sur l'essentiel plutôt que de perdre son énergie à programmer des routines de communication bas niveau. Ce module a aussi été prévu pour supporter la notion de profiles de messages, l'idée étant d'établir un support de base pour une famille de protocoles de messages de plus haut niveau. Un exemple de profil serait

une implémentation du "ebXML Transportation, Routing, and Packaging Message" ; mais nous ne détaillerons pas ces concepts ici.

Le module JAXP(Java API XML Parsers) JAXP est une collection de classes conçue pour permettre aux développeurs de créer des procédures de traitement de documents XML à l'aide de parseurs DOM(level 1 and 2 compliant), SAX(version 1.0 et 2.0) et XSLT. Ce module supporte également les schémas XML et le compilateur XML(XSLTC).

Le module JAXR(Java API XML Registry) JAXR fournit une API standard et uniforme pour accéder aux différents types de services "Registry" XML. Un service "Registry" a comme fonction de construire, déployer et découvrir des services sur le web.

Le module JAX-RPC JAX-RPC est une API Java conçue pour construire des applications web ainsi que des services web incorporant des fonctionnalités RPC basée sur XML suivant les spécifications SOAP 1.1. Avec ce module, il est possible de développer rapidemment des services web basés sur une large gamme de standards et protocoles. Le mécanisme RPC(Remote Procedure Call) quant à lui permet à un client d'envoyer des appels de procédure sur un serveur distant. Autrement dit, on peut imaginer un modèle dans lequel un serveur proposerait en service une collection de procédures utilisables à partir d'une autre machine géographiquement éloignée.
Une RPC basée sur XML peut se présenter sous la forme d'un appel de procédure utilisant un protocole comme SOAP 1.1 ; ce dernier définissant les conventions de représentation des appels et des réponses.

Une application complète utilisant le module JAXP pourra être trouvée en annexe "E.2". Il s'agit d'une application qui convertit les feuilles XML de configuration d'agent dans un autre format, celui définit par OpenMaster pour ses agents. En outre, les feuilles sont validées à l'aide d'un schéma XML du W3C. Cette validation a pour but de garantir la cohérence des configurations stockées.

4.6 Analyse de portabilité

Nous allons aborder ici la problématique de la portabilité des librairies présentées dans la section précédente. Nous allons nous intéresser plus spécialement aux cinq plateformes rencontrées le plus fréquemment sur les stations de travail : Windows, Linux, Unix AIX (IBM), Unix solaris et HP Unix.

4.6.1 Windows NT

Cette plateforme est sans nulle doute la plus courante, surtout dans le monde du PC ou elle possède un taux d'utilisation supérieur à 90%. De ce fait, il est évident que les développeurs pensent à fournir une version de chacune de leurs librairies pour ces utilisateurs.

Les librairies Xerces C++ le groupe "Apache.org" met à disposition non seulement les sources mais également une version compilée(des binaires) pour Windows. La seule restriction provient du fait que ces binaires ont été compilés sur Microsoft Visual C++ et que les auteurs ne garantissent pas une compatibilité complète avec d'autres programmes de compilation.

Perl Perl est apparu comme étant le langage le plus portable des trois. En effet, il fut l'un des seuls testés dans le cadre de ce mémoire n'ayant posé aucune difficulté sur aucune des plateformes. Pour Windows, il convient de se procurer l'interprèteur Perl développé par la firme Active State dénommé "Active State Perl". En outre, les modules Perl fournit entre autre par le site du CPAN peuvent y être installé sans la moindre difficulté.

Java Sun nous fournit une version de sa machine virtuelle pour Windows, ce qui implique une parfaite compatibilité avec tous les autres packages Java que Sun pourrait développer.

4.6.2 Linux

Le petit système développé par Linus Benedict Torvalds au début des années nonantes n'a cessé de se développer depuis sa création. Se voulant le plus ouvert possible, aussi bien sur le plan philosophique(GPL) que sur la multitude d'outils de programmation disponibles, Linux s'est vu muni d'un compilateur C++, Perl et Java. Ainsi, aucune librairie présentée ci-dessus n'ont donné de fils à retordre pour leur installation. Il est donc plus que possible de travailler avec le langage de son choix sous cette plateforme.

4.6.3 Unix AIX

La plateforme Unix de chez IBM est un système commercial et propriétaire possédant ses propres packages. Néanmoins, la version 4.3 dispose maintenant d'une toolbox lui permettant d'avoir une plus grande interopérabilité avec Linux, ce qui ouvre pas mal de perspectives pour utiliser les nombreux outils de développement linuxiens disponibles.

Les librairies Xerces C++ Malheureusement, l'installation de librairies comme celles de Xerces peuvent poser des difficultés inattendues. En effet, la volonté des auteurs des librairies Xerces étaient d'avoir une compatibilité maximum(on avait des binaires disponibles pour windows, solaris, AIX, Linux et HP Unix). Donc, ils avaient prévu un script "runConfigure" mis au point pour générer, à partir des sources, les librairies en fonction de la plateforme et des compilateurs utilisés. Mais si, pour une certaine raison, vous vouliez travailler avec gcc sous AIX, ce qui est maintenant tout à fait possible grâce à la linux toolbox, alors le script devait être adapté de manière à faire appel au script makeC++SharedLib, conçu pour générer des librairies dynamiques avec gcc sous AIX. Autrement dit, le script de configuration de base avait été pensé en partant du principe que des compilateurs précis étaient utilisés sur chacune des plateformes, tout autre combinaison devant être gérée au cas par cas par le développeur. De plus, et étant donné que les binaires pour AIX avaient été compilés par défaut avec xlC, il nous fallait absolument les recompiler avec gcc, pour

utiliser ce compilateur. Cette exigence vient du fait qu'en pratique, il s'avére que si le compilateur utilisé pour créer les librairies et celui utilisé pour construire le programme sont différents, alors la résolution des symboles dynamiques au moment de la compilation du programme ne se fait pas de manière correcte, car les environnements de compilation sont incompatibles, impliquant très souvent une erreur de "segmentation fault".

Pour mieux faire comprendre pourquoi il est plus intéressant de travailler avec des librairies dynamiques plutôt qu'avec des statiques, nous allons décrire ici ces deux modes de fonctionnement et expliquer pourquoi des librairies statiques ne sont pas recommandées dans ce cas.

Le mode dynamique est une technique qui consiste à reporter le chargement en mémoire du code des librairies lors de l'exécution du programme. En gros cela revient, à la compilation, à remplacer l'édition de liens statique par l'ajout de code et d'informations suffisants pour aller chercher les symboles non encore résolus. En mode statique par contre, le code des fonctions des librairies utilisées dans un programme est directement intégré dans le code exécutable. Dans notre cas, étant donné la taille des librairies et les nombreuses fonctions requises pour traiter un document XML, nous aurions obtenu après compilation des exécutables de taille gigantesque, ce qui aurait pu saturer la mémoire.[1]

Le tableau ci-dessous reprend les compilateurs recommandés pour les librairies Xerces en fonction de la plateforme :

Système d'exploitation	Compilateur C++,C
Windows	Visual C++
Linux	g++, gcc(egcs)
AIX 4.2.1 et supérieur	xlC_r, xlc_r
Solaris	CC, cc
HP-UX	aCC, cc

Perl et Java Pour ce qui est des deux autres langages qui nous intéresse ici, nous savons déjà que Perl ne pose aucune difficulté. Par contre, bien que Sun ne fournisse aucune machine virtuelle java pour AIX, cela ne l'empêche pas d'en posséder une bien à lui. On ne peut cependant rien dire concernant la compatibilité du XML pack de Sun puisqu'aucun test n'a été réalisé concretement. Par contre, la philosophie java nous permettrait d'affirmer qu'aucun problème ne devrait être rencontré.

4.6.4 Unix solaris

Solaris est la plateforme Unix de chez Sun Microsystem.

[1] http ://www.enseignement.polytechnique.fr/profs/informatique/Luc.Maranget/compil/poly/poly003.html

Les librairies Xerces C++ Elles ont été prévues pour fonctionner sous Solaris avec son compilateur C "cc". Néanmoins, il s'est avéré après quelques petits tests, qu'utiliser gcc sous Solaris pour compiler ces librairies n'était véritablement pas un obstacle, comme ce fut le cas pour AIX, où rappelons le, il fût nécessaire de modifier le script pour mener l'opération d'installation à son terme.

Perl et Java La portabilité des modules perl ainsi que des librairies java(rappelons que la version Java utilisée ici provient de chez Sun) n'ont posé aucune difficulté.

4.6.5 HP Unix

Voici la dernière et la plus compliquée des plateformes testées. En fait, il est très difficile de donner beaucoup de détails concernant ce Unix, car l'étude de portabilité a été réalisée sur une machine Evidian dont les accès et les possibilités de tests étaient fortement réduites, et ce afin de ne pas endommager la configuration de la machine (il s'agissait d'une machine de production). De manière pratique, Perl a été testé avec succès, tandis que Xerces n'as pu être totalement éprouvé. Quant à Java, il ne faisait pas partie des objectifs du stage et n'as donc, de ce fait, pas pu être essayé.

Les librairies Xerces C++ L'utilisation de Xerces avec le compilateur gcc s'est avéré être un super casse-tête. Il est grandement recommandé ici de garder le compilateur préconisé par l'équipe Apache sur cette architecture, à savoir "aCC". En outre, le système de gestion des librairies dynamiques semblent être différent que sur les autres Unix car il y a une discordance des extensions : "sl" pour HP-UX et "so" pour les autres. Pour contourner ce problème, il suffisait parfois de créer un lien symbolique dont le nom correspondait à la cible mais avec une extension dynamique "so" pour s'en sortir. Précisons tout de même que cela ne marchait pas dans tous les cas.

Perl Comme nous l'avons dit plus haut, il n'y a pas eu de problème pour utiliser Perl.

Java Il semble y avoir un support pour une version propriétaire HP. Ainsi, si la philosophie java est respectée, il ne devrait y avoir aucun problème. On pourra par ailleurs trouver tous les renseignements sur :

```
http://www.hp.com/products1/unix/java
```

4.7 Où en sommes-nous ?

Ce chapitre nous permet d'apprécier non seulement les différents avantages et inconvénients des parseurs courants disponibles, mais aussi les conditions dans lesquelles nous pouvons travailler. Rappelons-nous qu'OpenMaster est un système de surveillance de réseaux hétérogènes, où la diversité des systèmes d'exploitation s'ajoute aux multiples architectures matérielles. Il est donc important de savoir comment et avec quels outils nous serons en

mesure de porter nos parseurs sur un maximum de machines, pour ne pas dire toutes. La portabilité est donc primordiale ici car elle implique une universalité de notre travail en matière de configuration à partir de la technologie XML.

5 Centralisation des configurations dans un annuaire électronique

LDAP ou Lightweight Directory access Protocol est un service d'annuaire électronique dont la fonction première est de retourner des attributs d'objets à l'aide de fonctions de recherche multi-critères. Il peut être considéré comme une base de données spécialisée dont l'objectif premier est de centraliser les informations et de les rendre disponibles. LDAP est très performant en lecture mais l'est beaucoup moins en écriture. Il vient d'une adaptation du protocole DAP (protocole d'OSI pour les accès aux annuaire X500) à TCP/IP. En 1995, il devenait un annuaire natif grâce au travail d'une équipe de l'Université du Michigan.

5.1 Pourquoi choisir LDAP ?

Dans le cadre qui nous intéresse, à savoir comment stocker des configurations d'agent de format XML dans un système de base de données, il est apparu que le service d'annuaire électronique se trouvait être une bonne solution. En effet les avantages de cet architecture sont les suivants :

- La structure hiérarchique commune entre LDAP et XML
- Les performances en lecture d'un serveur LDAP
- Les possibilités étendues dans les critères de recherches
- LDAP permet d'organiser les résultats d'une recherche
- Le système de réplication qui permet d'augmenter la disponibilité des informations

Analysons quelque peu les points cités ci-dessus :

La structure hiérarchique Les documents XML sont, comme nous l'avons vu au chapitre 3, structurés hiérarchiquement. Cette similitude avec la structure LDAP va nous facilité grandement les choses lorsque le contenu d'une configuration devra être exportée vers un annuaire. En effet, si nous devions le faire avec un système de base de données dites relationnelle, qui elle se structure de manière tabulaire, l'exportation en serait plus difficile puisque il faudrait au moins une table par niveau de hiérarchie ; le lien se faisant entre-elles grâce à un système de clés et de clés étrangères relativement complexe.

Les performances en lecture et la réplication Les serveurs LDAP ont indéniablement été optimisé pour la lecture, les mises à jour étant quant à elles un peu plus complexe à réaliser. Cet avantage, lié à la réplication, permet d'entrevoir une solution où tous les agents présents sur chacunes des machines d'un réseau local donné pourront charger leur configuration non pas avec un fichier local, comme c'est encore le cas aujourd'hui, mais bien de le faire à partir d'un annuaire en fonction du critère du type de configuration(champ configName que nous expliquerons plus loin) ainsi que sa version. De cette manière la réplication permet à tous les agents d'obtenir un temps de réponse acceptable à leurs requête dans des

conditions extrême d'utilisation (tous les agents demandent à lire la même configuration en même temps par exemple).

Les critères de recherches et l'organisation des résultats A la requête "ldapsearch -b "version=1.0,configName=essai2,o=RedEagle Corp.,c=be" "objectclass=*"" on obtient le résultat suivant :

```
version: 2

#
# filter: objectclass=*
# requesting: ALL
#

# 1.0, essai2, RedEagle Corp., be
dn: version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
version: 1.0
objectClass: aic

# 3, 1.0, essai2, RedEagle Corp., be
dn: cfgFilterId=3,version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
cfgFilterSpecific: 33
cfgFilterPeriodValid: 30
cfgFilterLabel: alix-uxLoginSession-setFailed
cfgFilterId: 3
cfgFilterGeneric: 6
cfgFilterEnterprise: bull.118
cfgFilterDomain: 2
cfgFilterCptMax: 3
objectClass: cfgFilterEntry

...

# 3, 1.0, essai2, RedEagle Corp., be
dn: cfgDomainId=3,version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
cfgDomainId: 3
cfgDomainLabel: coach
objectClass: cfgDomainEntry

# 1.3.6.1.4.1.107.146.1.3.4.3.0, 3, 1.0, essai2, RedEagle Corp., be
```

```
dn: cfgAtt1=1.3.6.1.4.1.107.146.1.3.4.3.0,cfgDomainId=3,version=1.0,
configName=essai2,o=RedEagle Corp.,c=be
cfgAtt1Value:: dmFsdWUg
cfgAtt1Comparaison:: ZXhpc3Qg
cfgAtt1: 1.3.6.1.4.1.107.146.1.3.4.3.0
objectClass: cfgAtt1Entry

# 0.0, 3, 1.0, essai2, RedEagle Corp., be
dn: cfgAtt2=0.0,cfgDomainId=3,version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
cfgAtt2Value:: dmFsdWUg
cfgAtt2Comparaison:: ZXhpc3Qg
cfgAtt2: 0.0
objectClass: cfgAtt2Entry

...

# 2, 1.0, essai2, RedEagle Corp., be
dn: cfgDomainId=2,version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
cfgDomainId: 2
cfgDomainLabel: mib 2
objectClass: cfgDomainEntry

# 1.3.6.1.2.1.1.3.0, 2, 1.0, essai2, RedEagle Corp., be
dn: cfgAtt1=1.3.6.1.2.1.1.3.0,cfgDomainId=2,version=1.0,
configName=essai2,o=RedEagle Corp.,c=be
cfgAtt1Value:: dmFsdWUg
cfgAtt1Comparaison:: ZXhpc3Qg
cfgAtt1: 1.3.6.1.2.1.1.3.0
objectClass: cfgAtt1Entry

# 0.0, 2, 1.0, essai2, RedEagle Corp., be
dn: cfgAtt2=0.0,cfgDomainId=2,version=1.0,configName=essai2,
o=RedEagle Corp.,c=be
cfgAtt2Value:: dmFsdWUg
cfgAtt2Comparaison:: ZXhpc3Qg
cfgAtt2: 0.0
objectClass: cfgAtt2Entry

...
```

En observant la structuration de ce résultat, on peut voir que l'affichage de l'arbre se fait en profondeur d'abord. Ceci nous apporte l'assurance qu'à partir d'un noeud donné de

l'arborescence, on pourra recupérer tous les noeuds fils dans un ordre bien établi. Nous aurons donc une certaine facilité pour traiter le résultat reçu et le reconvertir en feuille XML par la suite. Cette remarque est importante puisque pour obtenir une centralisation complète, il faut être capable de transférer ces configurations du format XML vers le format LDAP et vice-versa.

5.2 Description d'une architecture LDAP

L'architecture de communication LDAP est basée sur le modèle client-serveur. Il permet aux utilisateurs de se connecter, de se déconnecter, de rechercher, de comparer, de créer, modifier ou effacer des données. Il supporte aussi des mécanismes de chiffrement comme SSL ou TLS ainsi que des mécanismes d'authentification comme SASL. Les règles d'accès (ACL) permettent quant à elles de protéger les transactions et l'accès aux données.

5.3 Le modèle de données LDAP

Comme LDAP vient d'un protocole basé sur un service d'annuaire X500, les serveurs sont conçus pour pour stocker une grande quantité d'information de faible volume dans un modèle hiérarchique.

5.3.1 Le directory Information Tree

Le modèle LDAP de stockage de données est donc une arborescence hiérarchique hérité du protocole X500. Chaque noeud de l'arborescence correspond à une entrée de l'annuaire. Ces entrées sont des objects concrèts ou abstraits du monde réel come des personnes ou des paramètres de configuration.

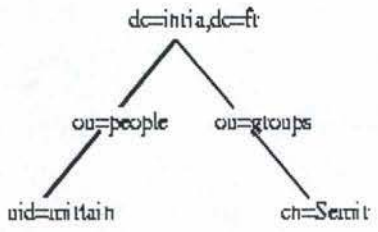

FIG. 15 – Exemple de structure LDAP

5.3.2 Les schémas

Les schémas sont des fichiers qui contiennent les déclarations relatives aux objets (object-class) et aux attributs utilisés dans ces objets. Ces deux types d'éléments sont caractérisés par une série de propriétés.
Les déclarations d'attributs sont composées des éléments suivants :

- Un nom qui l'identifie

- Un OID qui l'identifie aussi
- Un paramètre qui permet de déterminé si un élément un multi-valué ou non
- Une syntaxe pour les règles de comparaison
- Un indicateur d'usage
- Un format ou une limite de taille de valeur

Exemple de déclaration :

```
attributetype ( 2.16.840.1.113730.3.1.1 NAME 'carLicense'
    DESC 'RFC2798: vehicle license or registration plate'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

- La première ligne donne l'OID 2.16.840.1.113730.3.1.1 suivi du nom 'carLicense'
- La seconde ligne donne une description textuelle de l'attribut
- La troisième ligne donne la règle EQUALITY,la syntaxe pour les comparaisons exactes (case insensitive, space insensitive)
- La quatrième ligne donne la règle SUBSTR,la syntaxe pour les comparaisons approximatives (case insensitive, space insensitive)
- La dernière ligne donne la syntaxe, c'est-à-dire le typage de l'attribut (chaine de caractères en UTF-8)

Les déclarations de classes d'objet sont composées des éléments suivants :

- Un nom qui l'identifie
- Un OID qui l'identifie aussi
- Les attributs obligatoires
- Les attributs facultatifs
- Le type de la classe (structurelle—auxiliaire—abstraite)

Exemple de déclaration :

```
objectclass ( 2.5.6.12 NAME 'applicationEntity' SUP top STRUCTURAL
    MUST ( presentationAddress $ cn )
    MAY ( supportedApplicationContext $ seeAlso $ ou $ o $ l
                                                $ description ) )
```

- La première ligne donne l'OID 2.5.6.12, le nom de la classe 'applicationEntity',la classe parente 'top' et le type de la classe
- La seconde ligne donne une description textuelle de l'attribut
- La troisième ligne donne les attributs obligatoires
- La dernière ligne donne les attributs facultatifs

Le qualificatif "SUP" indique la classe parente, ce qui signifie que les attributs de cette classe peuvent également être utilisées dans la classe fille, un peu à la manière du mécanisme d'héritage de la programmation Orientée objet.
Les classes structurelles sont les seules classes instanciables. Elles décrivent les objets que l'on souhaite utiliser dans l'annuaire. Les classes auxiliaires permettent de rajouter des informations complémentaires à des classes structurelles, un peu comme les fichiers include ou header. Les classes abstraites sont utilisées comme occupants de place, elles servent essentielles comme classes parentes ou superclasses afin de définir une hiérarchie et définir un système d'héritage entre les classes. Pour utiliser les objets relatifs aux configurations d'agent, il a été nécessaire de créer de nouveaux attributs et de nouvelles classes qui pourront être consultées en annexe "F".

5.3.3 Le "distinguish Name"

Chaque entrée est identifiée dans le directory Information Tree par son DN(distinguish Name). Celui-ci représente le nom de l'entrée sous forme d'un chemin d'accès depuis le sommet de l'arbre.Par exemple, le DN de la feuille gauche de l'arbre du schéma ci-dessus est : uid=mirtain,ou=people,dc=inria,dc=fr
On peut aussi utiliser des RDN (relatives distinguished names) pour accéder aux données. Par exemple, avec une position de base dc=inria,dc=fr on peut utiliser les RDN suivants :

- ou=people
- ou=groups
- cn=Semir,ou=groups
- uid=mirtain,ou=people

5.3.4 Les fichiers LDIF

LDIF ou LDAP Data Interchange Format permet de représenter les données LDAP dans un format texte standardisé. Ce fichier peut être utilisé pour ajouter ou modifier des données dans la base
Exemple de fichier LDIF pour modifier le champ alixdProcessName :

```
dn: alixdProcessEntryID=Zombies.<defunct>,version=1.0,configName=essai,
    o=RedEagle Corp.,c=be
changetype: modify
replace: alixdPrProcessName
alixdPrProcessName: defunct
```

Un exemple de fichier LDIF représentant une configuration particulière de chacun des agents pourra être trouvée en annexe "F".

5.3.5 Le service de réplication

Outre le service de partitionnement que nous n'aborderons, LDAP dispose d'un autres service nommé "réplication". Ce dernier consiste à recopier tout ou en partie le contenu d'un arbre sur un ou plusieurs serveurs afin de notamment, rapprocher le service des utilisateurs, de faire du load balancing, d'importer des entrées générées sur un autre serveur, de pallier aux pannes et aux coupures du réseau. Ce service permet d'améloiorer sans aucun doute les performances, la rapidité et la sûreté de l'annuaire.

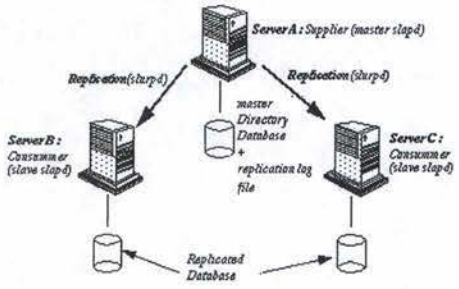

FIG. 16 – Service de réplication LDAP

5.4 Intégration des configurations dans LDAP

Le schéma ci-dessous représente la racine de l'arborescence. Le sommet correspond au nom d'une entreprise qui utilise des agents OpenMaster et qui souhaite en conserver les configurations sur un annuaire LDAP. Au second niveau, on trouve les différentes configurations des agents, le nom "configName" devant être considéré comme un identifiant. Nous avions aussi le choix d'avoir un identifiant propre à chaque agent comme OAC_configName, OAX_configName ou bien encore Coach_configName. L'avantage de cette autre solution était de pouvoir donner le nom souhaité à une configuration sans se soucier si ce nom avait déjà été donné pour un autre agent ; cette difficulté pouvant être contournée dans le premier cas par l'ajout d'un préfixe ou d'un suffixe propre à chaque agent. Cependant nous laisserons le soin au lecteur d'apprécier quelle solution lui semble la meilleure. Au dernier niveau nous avons les différentes versions relatives à une configuration donnée ; cela permet de garder un petit historique de l'évolution ainsi que de permettre des retours en arrière au cas où une modification s'avèrerait défectueuse.

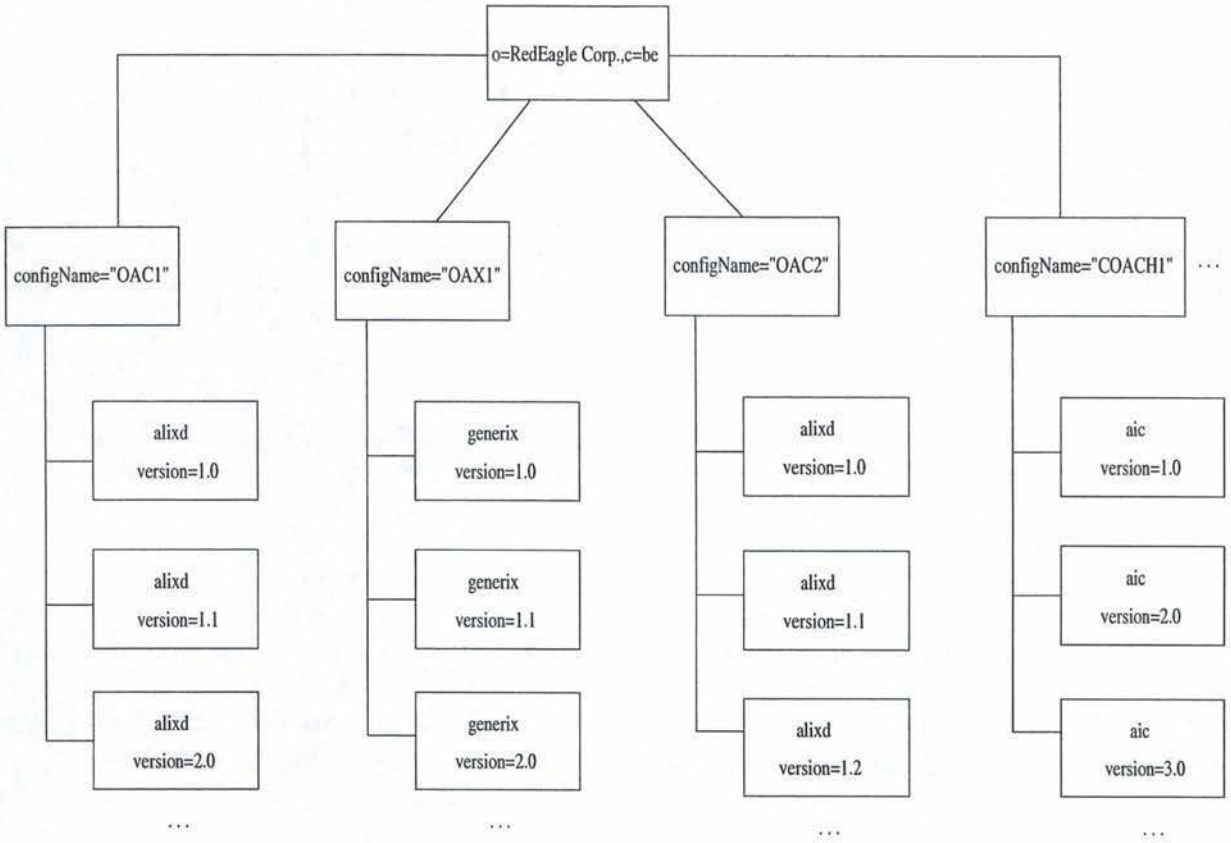

FIG. 17 – Structure LDAP générale

5.4.1 L'arborescence de l'agent de contrôle

Cette partie de l'arborescence vient se raccrocher à la base au niveau de la version. Le premier niveau étant la classe alixd, choisie pour contenir le champ "version" à cause de son unicité dans une configuration d'agent de contrôle ; et le second étant composé des autres tables qui constituent la configuration de cet agent. Un autre avantage pratique de choisir "alixd" pour contenir la version fut d'épargner un niveau dans l'arborescence, car ajouter une classe juste avec un champ "version" comme attribut paraissait être du gaspillage inutile de ressources.

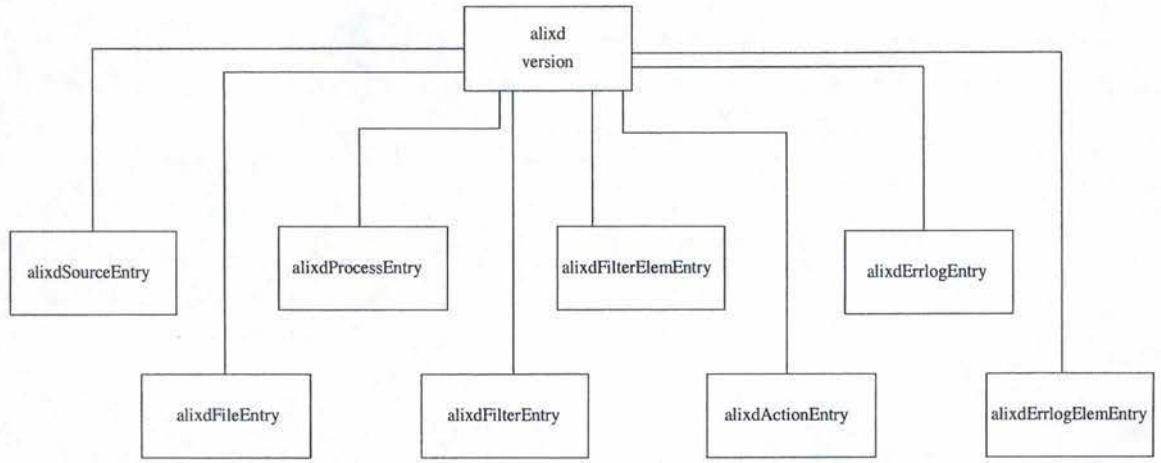

FIG. 18 – Structure LDAP de l'agent de contrôle

5.4.2 L'arborescence de l'agent d'extension

L'agent d'extension étant identique à l'agent de contrôle au niveau de la structure, le commentaire est le même qu'au point précédent.

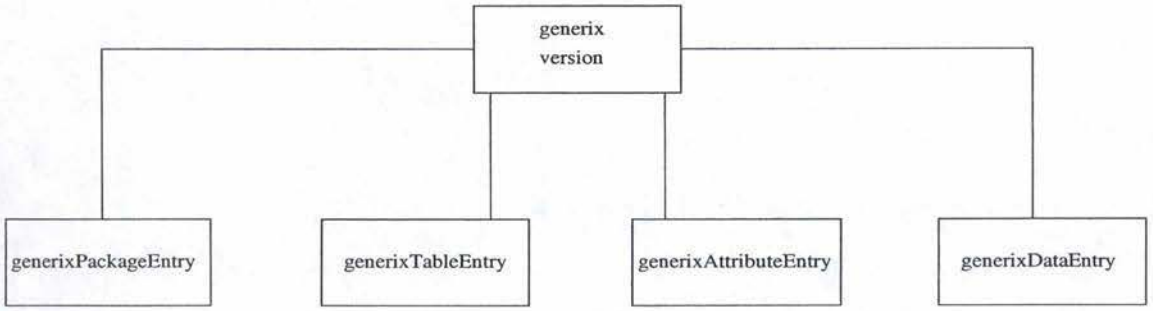

FIG. 19 – Structure LDAP de l'agent d'extension

5.4.3 L'arborescence de l'agent COACH

Cette arborescence se raccroche à la partie version comme dans les deux cas précédent. On remarquera ici la présence d'un troisième niveau. Cette décomposition s'est avéré nécessaire afin de faciliter la reconstruction de la feuille XML à partir de requêtes LDAP. En effet, ces éléments disposant eux-même d'attributs, ils devaient dès lors devenir des "objectclass". Plutôt que de les mettre au second niveau, on a préféré en créer un troisième afin de garder la même structure hiérarchisée qui existait déjà dans la feuille XML. Pour en revenir avec la remarque concernant la version, il faut savoir qu'ici aucune classe ne pouvait être une bonne condidate pour abriter le champ "version" ; c'est pour cette raison que la classe "aic" fut ajoutée.

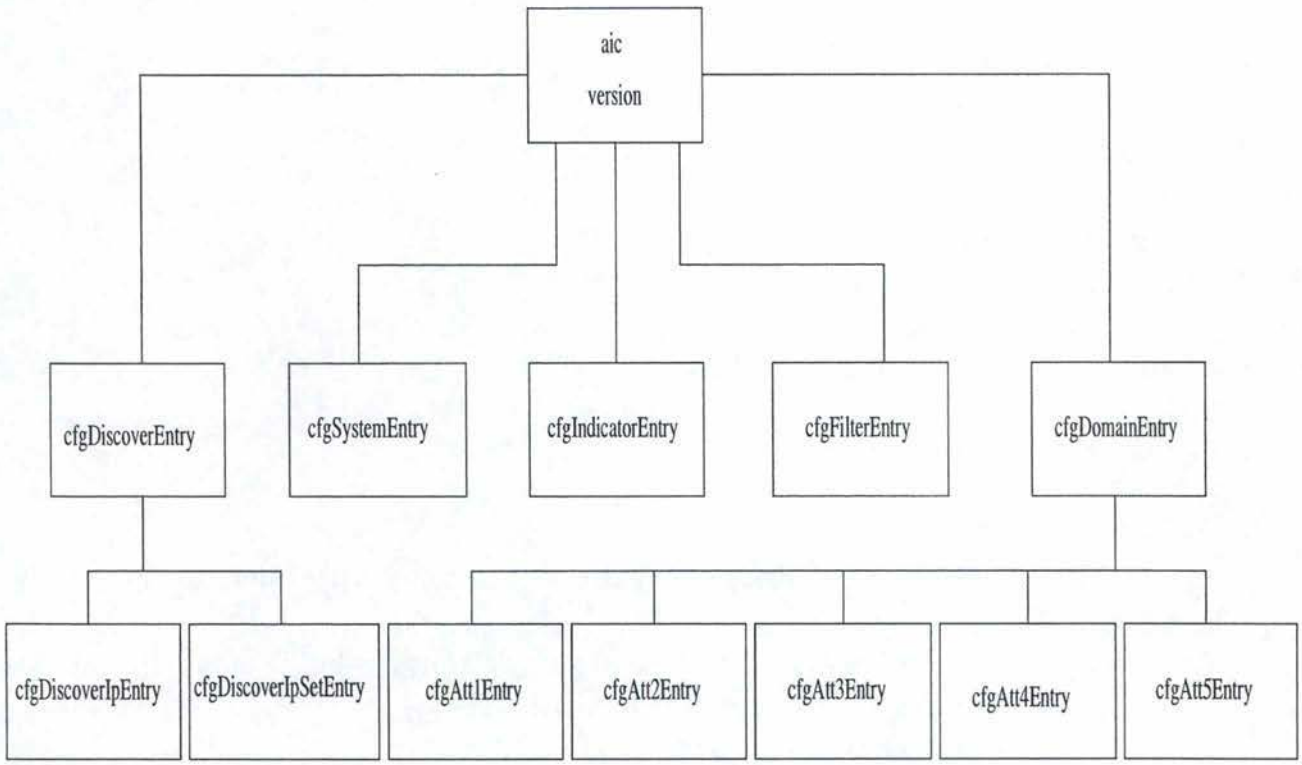

FIG. 20 – Structure LDAP de l'agent COACH

5.5 Les langages de programmation et LDAP

Nous allons aborder ici la problématique de l'interfaçage entre LDAP et les langages de programmation auquels nous nous intéressons depuis le début de ce travail.

5.5.1 Le langage C++

Les librairies ont été mise au point par Ralf Haferkamp dans le cadre de son travail de fin d'étude. Elles sont intitulées ldapcpplib et sont disponibles soit en source dans les dernières versions d'OpenLDAP(www.openladp.org), soit sous forme de package à l'adresse suivante : http ://at.rpmfind.net/opsys/linux/RPM/LByName.html(ou encore sur un des nombreux sites mirroirs Linux Suse).
Il serait inutile ici de détailler les différentes librairies nécessaires à leur installation ou à leur compilation, ainsi que les API disponibles, la documentation étant suffisamment explicite à ce sujet. Mais voici un petit exemple qui ajoute une entrée, lit l'ensemble des entrées disponibles et supprime l'entrée ajoutée dans l'annuaire LDAP :

```
#include <string.h>
#include <stdlib.h>
#include <iostream.h>
#include <strstream>
#include <iterator.h>
#include <LDAPConnection.h>
```

```
#include <LDAPConstraints.h>
#include <LDAPSearchReference.h>
#include <LDAPSearchResults.h>
#include <LDAPAttribute.h>
#include <LDAPAttributeList.h>
#include <LDAPEntry.h>
#include <LDAPException.h>
#include <LDAPModification.h>
#include <LDAPReferralException.h>

int main(int argC, char* argV[])
    {
    try {
        LDAPConnection* connexion=new LDAPConnection("redscorpion",389,new
            LDAPConstraints());
        connexion->bind("cn=ldap_admin,o=RedEagle_Corp.,c=be","secret",0);
        LDAPAttributeList* attrs=new LDAPAttributeList();
        StringList*  liste_cn =new StringList();
        liste_cn ->add("Verenne_Alain");
        liste_cn ->add("raptor");
        attrs->addAttribute(*new LDAPAttribute("cn",*liste_cn));
        attrs->addAttribute(*new LDAPAttribute("sn","Verenne"));
        attrs->addAttribute(*new LDAPAttribute("telephoneNumber","071/63.35.84"
            ));
        attrs->addAttribute(*new LDAPAttribute("objectclass","person"));
        LDAPEntry* entree=new LDAPEntry("cn=Verenne_Alain,o=RedEagle_Corp.,c
            =be",attrs);
        connexion->add(entree,0);
        LDAPSearchResults* resultat=connexion->search("o=RedEagle_Corp.,c=be",
            LDAPAsynConnection::SEARCH_SUB,"objectClass=*",*new StringList(),
            false,0);
        entree=resultat->getNext();
        while(entree!=0)
            {
            cout << "dn: " << entree->getDN() << endl;
            const LDAPAttributeList* list=entree->getAttributes();
            LDAPAttributeList::const_iterator attr=list->begin();
            for(long i=0;i<list->size();i++)
                {
                const StringList* values=&attr->getValues();
                StringList :: const_iterator  value=values->begin();
                for(int j=0;j<values->size();j++)
                    {cout << attr->getName() << ": " << *value++ << endl;}
```

```
            attr++;
            }
        cout << endl;
        entree=resultat->getNext();
        }
    connexion->del("cn=Verenne Alain,o=RedEagle Corp.,c=be",0);
    connexion->unbind();
    cout << "Fin de traitement" << endl;
    }
catch (const LDAPReferralException& toCatch)
    {cerr << toCatch.getResultMsg() << endl;}
catch (const LDAPException& toCatch)
    {cerr << toCatch.getResultMsg() << endl;}
}
```

5.5.2 Le langage perl

Perl dispose également d'un module d'interaction LDAP, perl-ldap. Inutile de passer en revue les dépendances de package, il suffira pour avoir des précisions de consulter la documentation fournie avec le module. En outre, il existe sur le site CPAN un petit document HTML décrivant de manière concise et précise les différentes API perl pour LDAP. Regardons juste l'exemple suivante qui effectue trois opérations successives :

- Lecture de l'ensemble des entrées de l'annuaire
- Ajout d'une entrée
- Suppression de cette entrée

```
use Net::LDAP;

$ldap=Net::LDAP->new('redscorpion:389') or die "$@";
$ldap->bind;
$mesg=$ldap->search (base => "o=RedEagle Corp.,c=be",filter => "(&(objectclass
    =*))");
$mesg->code && die $mesg->error;
foreach $entry ($mesg->all_entries) { print $entry; $entry->dump; }
$ldap->unbind;

$ldap=Net::LDAP->new('redscorpion:389');
$ldap->bind (dn => 'cn=ldap_admin,o=RedEagle Corp., c=be',password => 'secret')
    ;
$result=$ldap->add (dn => 'cn=Verenne Alain,o=RedEagle Corp.,c=be',
                    attr => [ 'cn' => ['Verenne Alain', 'raptor' ],
                              'sn' => 'Verenne',
                              'telephoneNumber' => '071/63.35.84',
```

```
                                'objectclass' => ['top', 'person',
                                'organizationalPerson' ]]) ;
$result->code && warn "failed to add entry: ", $result->error ;
$ldap->unbind;

$ldap=Net::LDAP->new('redscorpion:389');
$ldap->bind (dn => 'cn=ldap admin,o=RedEagle Corp.,c=be',password => 'secret')
    ;
$result=$ldap->delete('cn=Verenne Alain,o=RedEagle Corp.,c=be');
$result->code && warn "failed to add entry: ", $result->error ;
$ldap->unbind;
```

A titre illustratif, voici deux autres exemples :

```
#modification d'une entree

$ldap->modify($dn,changes => [
                    add => [ sn => 'Barr' ],              # Add sn=Barr
                 delete => [ faxNumber => []],            # Delete all fax
                    numbers
                 delete => [ telephoneNumber => ['911']], # delete phone number
                    911
                replace => [ email => 'gbarr@pobox.com']]); # change email address

#Utilisation de fichier LDIF (en lecture/ecriture)

use Net::LDAP::LDIF;
$ldif = Net::LDAP::LDIF->new( "file.ldif", "r", onerror => 'undef' );
while( not $ldif->eof() )
    {
    $entry = $ldif->read_entry();
    if ( $ldif->error() )
        {
        print "Error msg: ",$ldif->error(),"\n";
        print "Error lines:\n", $ldif->error_lines(),"\n";
        }
    else
        {# do stuff}
    }
$ldif->done();
```

5.5.3 Le langage java

Java dispose d'une interface capable d'interagir avec des services comme LDAP,RMI ou encore Corba. Cet interface porte le nom de "Java Naming and Directory Interface" ou

JNDI. Ces API fournissent des fonctionnalités permettant de gérer des services "naming" et "directory" en ne se servant que du langage de programmation Java. Ces interfaces ont été définies dans un objectif d'indépendance vis-à-vis des services avec lesquels elles peuvent interagir
L'architecture JNDI consiste en une API et une interface jouant un rôle de fournisseur de service(service provider interface ou SPI). L'API JNDI est utilisée pour accéder à une série de services de type "directory" et "naming", tandis que le SPI autorise une variété de services d'être connectée de manière transparente, utilisant de ce fait la couche API pour tirer parti de ces services. La figure ci-dessous donne une petite idée de l'architecture de JNDI :

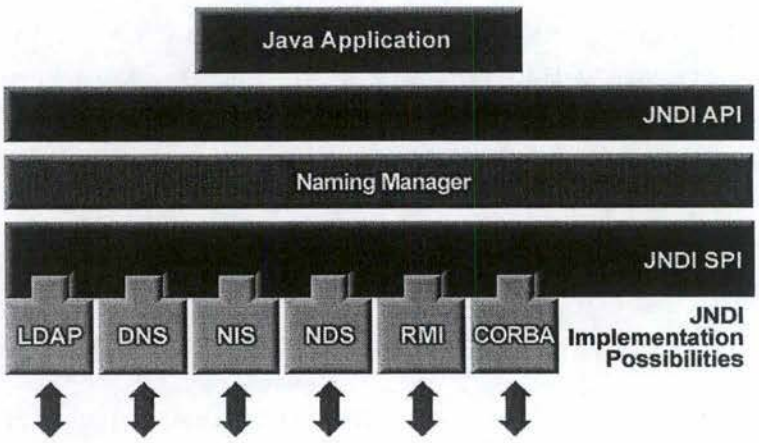

FIG. 21 – Structure de l'architecture JNDI

JNDI est donc une sorte de middelware entre le langage Java et les services concernés. Deux exemples d'application pourront être lus en annexe "F". Il s'agit de "ldapImport.java" et "ldapExport.java". Ces deux programmes importent et exportent respectivement une configuration XML représentant une configuration d'agent.
Le package JNDI peut être téléchargé sur le site sun "http ://java.sun.com".

5.6 Zope ou comment publier des données LDAP sur un Browser Web

Un dernier point que nous aborderons ici concerne la gestion des configurations contenues dans un annuaire LDAP. Comme nous l'avons vu au chapitre quatre, java dispose de toutes les librairies nécessaires pour traiter les feuilles XML et interagir avec LDAP. La question se pose maintenant de savoir comment envisager de mettre en oeuvre une plate-forme de gestion. Dans un premier temps, une application java munie d'une interface graphique pourrait sans aucun doute être réalisée, mais on se propose ici de présenter brièvement un autre outil de publication : Zope ou Z Object Publishing Environment.
Il s'agit en fait d'une plate-forme de développement d'application pour le web. On définit généralement une application "web" comme étant un programme informatique auquel les utilisateurs peuvent accéder par l'intermédiaire un navigateur comme Nestcape Navigator,

et ce à travers le réseau Internet. Dans ce genre d'application, les sites sont dynamiques et capables de fournir des informations sur les utilisateurs. Ils permettent également à ces derniers de faire interagir leurs applications avec des données issues de requêtes(SQL par exemple) grâce à des procédures stockées sur le serveur hébergeant le site. Autrement dit, vous pouvez consulter sur votre navigateur le résultat d'une requête SQL d'un serveur distant. Les différents composants utilisés par Zope sont les suivants :

Un serveur Web comme Apache ou Microsoft IIS(ou tout autre serveur supportant le Common Gateway Interface ou CGI) Zope en a besoin pour distribuer les contenus aux utilisateurs.

Une Interface Web Pour interagir avec le serveur Zope, vous utilisez une interface graphique de gestion à travers votre navigateur. Cette interface vous permet de créer des documents, des connexions à des bases de données externes ainsi que des scripts dans divers languages. C'est donc au travers de cette interface que vous gérez vos projets.

Un object d'interaction avec les bases de données Quand vous travaillez avec Zope, vous devez créer des objets de connexion vers la base de données cible ; ces objets étant stockés dans le système Zope de gestion de BD.

Des composants d'intégration relationnelle Zope fonctionne avec la plupart des systèmes de gestion de base de données relationnelles classiques comme Oracle, MySQL, PostgreSQL, Sybase,...

Support pour un certain nombres de langages de script Zope supporte perl, python mais aussi des documents natifs Zope DTML (Document Template Markup Language). Ces derniers sont des documents écrit dans un style HTML mais qui sont en réalités des scripts qui vont construire dynamiquement les pages résultats à renvoyer à l'utilisateur. Ces templates seront donc garnis des informations recueillies par la requête envoyée au serveur de base de données. Ce langage de script est un langage exécuté côté serveur comme PHP, ASP,...

Un exemple en annexe "G" illustrera l'interfaçage possible entre Zope et un annuaire LDAP. Cet exemple met en évidence la facilité avec laquelle on peut concevoir et construire une application Web capable d'interagir avec LDAP, et ce afin d'administrer les configurations des agents qu'on y aurait stocké.

5.7 Où en sommes-nous ?

Nous avons maintenant atteint notre second objectif de centraliser les configurations sur un serveur. Ce type de serveur possède en plus un service de réplication capable de faire du "load balancing". Cela permettrait de limiter le nombre de requêtes simultanées sur le même serveur et de répartir ainsi la mise à disposition sur plusieurs machines. Ce chapitre nous a également montré un outil intéressant pour administrer nos feuilles XML de

configuration. Il est d'autant plus intéressant que le programme client peut se limiter à un navigateur, ce qui permet d'accéder aux configurations à partir de n'importe quelle machine. Une solution en java, elle, aurait impliqué l'installation d'un client spécifique sur chaque station susceptible d'être concernée.

6 Conclusions

Nous venons de faire ici un petit parcours du monde XML appliqué dans un domaine particulier, celui de la création de fichiers de configuration. XML prend de plus en plus d'ampleur dans le monde informatique dû au fait de sa grande facilité d'utilisation et sa clarté dans la structure des documents. En outre il est relativement bien adapté pour les transmissions réseaux, ce qui le rend très intéressant pour des applications Internet.

Pour construire des feuilles XML de configuration, nous avons vu qu'il était nécessaire de les valider afin de vérifier si les documents XML représentent bien une configuration correcte. Bien sûr, il est difficile de mettre toutes les contraintes dans une DTD ou un schéma, on ne peut donc pas supprimer tous les tests du code de l'agent, mais d'un autre côté cela permet de grandement les simplifier. Nous avons passé en revue quelques moyens de valider des feuilles XML, depuis les DTD qui furent les premières dans ce domaine, jusqu'au divers types de schémas. Certaines personnes polémiquent déjà sur le fait de savoir si le langage du W3C pour la création de schéma ne modifierait pas la logique de la validation, et prône de ce fait plutôt l'utilisation du schématron ou de Relax NG. Mais à chacun de trouver le langage qui lui convient le mieux. Soit dit en passant, les schémas prendront sans nul doute le relais sur les DTD qui démontrent de plus en plus leurs limitations sur les contraintes de validation.

Nous nous sommes aussi intéressés à quelques langages de programmation et à la façon dont ces derniers permettent de travailler avec les divers parseurs comme SAX ou bien DOM. Le choix dépend bien sûr des goûts et des contraintes des développeurs, mais il pourrait aussi être guidé par des impératifs de portabilité. En effet, des petites expériences nous ont montrés que le langage le plus portable semblait être le Perl. Bien que ce langage dispose de tous les modules nécessaires pour aborder les différents points présentés ici, il faut savoir que certains de ces modules dépendent parfois eux-mêmes de librairies qui ne sont pas toujours aisées à trouver pour toutes les plateformes. Le gros problème provient souvent du fait que les codes sources sont optimisés pour le compilateur GNU cc(gcc) de Richard Stallman sous une plateforme i386 Linux. Pour les autres plateformes, il faut souvent modifier soi-même les scripts d'installation pour arriver au terme de la création des librairies, surtout si on souhaite utiliser gcc plutôt que d'utiliser le compilateur "dédié" comme xlc sous AIX. Malgré tout il faut savoir que gcc est portable sur plus de trente architectures différentes. Sinon, il existe aussi Java qui de part sa philosophie de machine virtuelle, apporte une solution acceptable et plus accessible que les autres langages. Il suffit de disposer d'une machine virtuelle adaptée au système d'exploitation et donc à l'architecture.
La dernière partie a porté sur la possibilité de centraliser les configurations sur un ou plusieurs serveurs afin d'augmenter la cohérence et la disponibilité de ces dernières. L'avantage certain des annuaires LDAP par rapport aux systèmes de gestion relationnelle est la similitude hiérarchique qui existe avec la structure des documents XML. En effet, il serait peu aisé de concevoir un système de tables pour conserver les configurations ; mais surtout il serait très complexe de construire des requêtes SQL permettant de les récupèrer ou de

les modifier. Ensuite nous nous sommes encore attardé sur les possibilités de construire des applications interagissant avec des annuaires suivant les mêmes langages utilisés pour XML. Bien qu'aucun test n'a pu être réalisé, les conclusions auraient été les mêmes : comment compiler les librairies sur toutes les plateformes. Les tests et exemples présentés dans le cadre de ce travail ont d'ailleurs été réalisé sur une plateforme Intel i386 Linux.

7 Perspectives

Pour terminer, nous avons examiné la façon dont nous pourrions consulter les configurations de manière graphique. Nous avons envisagé cela de deux manières. Soit nous utilisions le langage Java pour construire une interface graphique permettant d'interagir complètement avec LDAP et les feuilles XML, comprenant entre autre les fonctionnalités de lecture, mise à jour de la base LDAP, mais aussi des possibilités d'exportations et d'importations de documents XML. Soit nous utilisions un logiciel déjà préconçu dans cette optique : Zope. Le grand avantage de cette application réside dans son architecture orientée réseau voire Internet de manière plus générale. Plutôt que de créer un serveur et un client Java, on pourrait se contenter de concevoir un projet Zope disposant de toutes les méthodes nécessaires à la réalisation des divers tâches attendues ; le client se résumant à un simple Navigateur tel que Nestcape Navigator ou bien Internet Explorer. L'obstacle principal de cette solution n'est qu'une simple étude approfondie de Zope et de ses langages comme DTML, python ou encore perl.

Les éléments dont nous venons de parler pourraient à eux seuls faire partie d'un nouveau sujet de stage et pourquoi pas de mémoire. Une autre piste à suivre serait d'analyser les différents langages de schémas et voir si celui du W3C est le plus adapté au sujet étudié ici.

8 Bibliographie

Introduction à OpenMaster et ses agents

- Documentation fournie avec le logiciel OpenMaster, "Server and Station Installation Guide", Evidian S.A.
- Documentation fournie avec le logiciel OpenMaster, "OpenAgent Base User Guide", Evidian S.A.
- Documentation fournie avec le logiciel OpenMaster, "OpenAgent Control Module User Guide", Evidian S.A.
- Documentation fournie avec le logiciel OpenMaster, "OpenAgent Extension Module User Guide", Evidian S.A.
- Documentation fournie avec le logiciel OpenMaster, "OpenMaster COACH Installation and Configuration Guide", Evidian S.A.
- Documentation fournie avec le logiciel OpenMaster, "Station for Java User Guide", Evidian S.A.

Philosophie XML

- Recommandation du W3C du 10 février 1998 concernant XML 1.0 (REC-xml-19980210)
- "XML : une vue globale" par Olivier Perrin, Loria-ECOO, 14 juin 2001
- Recommandation du W3C du 14 janvier 1999 concernant les espaces de nommage XML (REC-xml-names-19990114)

Les DTD

- "Le langage SGML : vue d'ensemble et derniers progrès", George Charlebois, Flash Réseau numéro 3, décembre 1994
- "SGML (ISO 8879), Yves Marcoux, 1994-1996,
  http ://mapageweb.umontreal.ca/marcoux/INTRO/41.htm
  http ://mapageweb.umontreal.ca/marcoux/INTRO/41.htm

Les schémas XML du W3C

- Recommandation du W3C du 2 mai 2001 concernant XML Schema Part 0 : Primer (REC-xmlschema-0-20010502)
- Recommandation du W3C du 2 mai 2001 concernant XML Schema Part 1 : Structures (REC-xmlschema-1-20010502)
- Recommandation du W3C du 2 mai 2001 concernant XML Schema Part 2 : Datatypes (REC-xmlschema-2-20010502)

XPATH et XSL

- Recommandation du W3C du 16 novembre 1999 concernant le XML Path Language version 1.0 (REC-xpath-19991116)

- Recommandation du W3C du 15 octobre 2001 concernant XSL version 1.0 (REC-xsl-20011015)
- "XSL, Langage de feuilles de styles extensible", Antenna House XSL Formatter(Evaluation), www.mutu-xml.org/xml-base/shared/KEY-XSL.pdf

Schematron

- "Validating XML with schematron", Chimezie Ogbuji, 22 Novembre 2000, http ://www.xml.com/pub/a/2000/11/22/schematron.html
- "The schematron Assertion Language 1.5", Rick Jelliffe and Academia Sinica Computing Centre 2002-07-07, http ://www.ascc.net/xml/resource/schematron/Schematron2000.html
- "Schematron : validating XML using XSLT", Leigh Dodds, http ://www.ldodds.com/papers/schematron_xsltuk.html
- "Validating XML : schematron vs. XML schemas", Alx Dark, Ph.D., version 1.3(9 février 2001), http ://home.earthlink.net/ alxdark/software/schematron/index.html

TREX

- "TREX - Tree Regular Expressions for XML Language Specification", James Clark(Thai Open Source Software Center Ltd), http ://www.thaiopensource.com/trex/spec.html

Relax NG

- "Relax NG Specification", OASIS Committee Specification, 3 décembre 2001, http ://www.oasis-open.org/committees/relax-ng/spec-20011203.html
- "Implémentation RELAX NG", Edd Dumbill(auteur) Eric van der Vlist(traducteur), http ://xmlfr.org/actualites/tech/010614-0002
- "RELAX NG devient officiel", Michael Smith(auteur) Eric van der Vlist(traducteur), http ://xmlfr.org/actualites/tech/011206-0002

La validation

- "Page de manuel fournie avec le parseur xmlwf", Clark Cooper et Thai Open Source Software Center Ltd, 2000

Parseur SAX

- "Professional XML part 1 - SAX 1.0 : The simple API for XML", Wrox Books 16 février 2001, http ://tutorials.findtutorials.com/read/id/131
- "Parser du XML, les API SAX et DOM", http ://www.commentcamarche.net/xml/xmldomsax.php3
- "XML parsers - White Paper", California Software CO. Ltd.E-139A, http ://www.calsoft.co.in/techcenter/develop/xml.html

Parseur DOM

- "Parser du XML, les API SAX et DOM", http ://www.commentcamarche.net/xml/xmldomsax.php3
- "XML parsers - White Paper", California Software CO. Ltd.E-139A, http ://www.calsoft.co.in/techcenter/develop/xml.html
- Recommandation du W3C du 1 octobre 1998 concernant DOM Level 1 core version 1.0 (REC-DOM-Level-1-19981001)
- Recommandation du W3C du 13 novembre 2000 concernant DOM Level 2 core version 1.0 (REC-DOM-Level-2-Core-20001113)
- Recommandation du W3C du 13 novembre 2000 concernant DOM Level 2 Events version 1.0 (REC-DOM-Level-2-Events-20001113)
- Recommandation du W3C du 13 novembre 2000 concernant DOM Level 2 Style version 1.0 (REC-DOM-Level-2-Style-20001113)
- Recommandation du W3C du 13 novembre 2000 concernant DOM Level 2 Transversal and Range version 1.0 (REC-DOM-Level-2-Transversal-Range-20001113)
- Recommandation du W3C du 13 novembre 2000 concernant DOM Level 2 Views version 1.0 (REC-DOM-Level-2-Views-20001113)
- Recommandation du W3C du 9 avril 2002 concernant DOM Level 3 core version 1.0 (REC-DOM-Level-3-Core-20020409)
- Recommandation du W3C du 25 juillet 2002 concernant DOM Level 3 Load and Save version 1.0 (REC-DOM-Level-3-LS-20020725)
- Recommandation du W3C du 25 juillet 2002 concernant DOM Level 3 Validation version 1.0 (REC-DOM-Level-3-Val-20020725)

Interfaçage C++

- Documentation fournie avec le Package Xerces C++, The Apache Software Foundation, 2002,http ://xml.apache.org
- Documentation fournie avec le Package "ldapcpplib",Ralf Haferkamp, 1/09/2000

Interfaçage Perl

- "XML : :Parser : :Expat - Lowlevel access to James Clark's expat XML parser", Larry Wall and Clark Cooper, http ://www.perldoc.com/perl5.6.1/lib/XML/Parser/Expat.html
- "Using The Perl XML : :Parser Module", Clark Cooper,12/09/1998, http ://www.xml.com/pub/a/98/09/xml-perl.html
- "XML : :Parser", Larry Wall and Clark Cooper, http ://www.perldoc.com/perl5.6.1/lib/XML/Parser.html
- "XML : :Checker : :Parser", Enno Derksen, http ://www.nihongo.org/snowhare/utilities/perldoc2tree/example/XML /Checker/Parser.html
- "Using PerlSAX", David Megginson,

http ://www.durbanet.co.za/colin/xml/libxml-perl/UsingPerlSAX.html
- "XML : :DOM - A perl module for building DOM Level 1 compliant document structures", Enno Derksen Clark Cooper,
http ://jonas.liljegren.org/perl/libxml/XML/DOM.html
- "XML : :DOM - A perl module for building DOM Level 1 compliant document structures", Enno Derksen Clark Cooper,
http ://www.cgi101.com/modules/XML.dom.html
- "perl-ldap Documentation", ?,http ://perl-ldap.sourceforge.net/doc/

Interfaçage Java

- Documentation fournie avec le Package "Java XML Pack Summer 02 Bundle Release Notes", Sun Microsystems
- "The Java Web Services Tutorial. A beginner's guide to developing Web services and Web applications on the Java Web Services Developer Pack.", Eric Armstrong Stephanie Bodoff Debbie Carson Maydene Fisher Dale Green Kim Haase, Sun Microsystems,
http ://java.sun.com/webservices/docs/1.0/tutorial/index.html
- "Java API for XML Messaging (JAXM)", Sun Microsystems,
http ://java.sun.com/xml/jaxm/index.html
- "Java API for XML Processing (JAXP)", Sun Microsystems,
http ://java.sun.com/xml/jaxp/index.html
- "Java API for XML Registries (JAXR)", Sun Microsystems,
http ://java.sun.com/xml/jaxr/index.html
- "JavaTM API for XML-Based RPC (JAX-RPC)", Sun Microsystems,
http ://java.sun.com/xml/jaxrpc/index.html
- "The JNDI Tutorial. Building Directory-Enabled Java Applications", Rosanna Lee, Sun Microsystems, http ://java.sun.com

LDAP

- "LDAP", Laurent Mirtain -INRIA - octobre 1999,
http ://www-sop.inria.fr/semir/personnel/Laurent.Mirtain/LDAP.html
- "Introduction aux annuaires LDAP",
http ://www.commentcamarche.net/ldap/ldapintro.php3, Jean-François Pillou, 2001
- "Protocole LDAP",
http ://www.commentcamarche.net/ldap/ldapldap.php3, Jean-François Pillou, 2001
- "Le modèle d'information",
http ://www.commentcamarche.net/ldap/ldapinfo.php3, Jean-François Pillou, 2001
- "Le modèle de nommage",
http ://www.commentcamarche.net/ldap/ldapnomm.php3, Jean-François Pillou, 2001
- "Installation d'un serveur",
http ://www.commentcamarche.net/ldap/ldapinst.php3, Jean-François Pillou, 2001
- "Configuration d'un serveur",
http ://www.commentcamarche.net/ldap/ldapconf.php3, Jean-François Pillou, 2001

- Documentation fournie avec le Package OpenLDAP, The OpenLDAP foundation, 2000, http ://www.openldap.org

Zope

- "The Zope Book Covers Zope 2.5", Amos Latteier and Michel Pelletier, 2000 New Riders Publishing, http ://www.zope.org/Documentation/Books/ZopeBook/current/index_html
- "The Zope Developer's Guide", Amos Latteier Michel Pelletier and Chris McDonough, http ://www.zope.org/Documentation/ZDG

Annexes

A Documents DTD des agents OpenMaster

A.1 Agent de contrôle

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- DTD de la configuration de l'agent de controle -->

<!ELEMENT OACconfig (alixd,alixdSourcesEntry?,alixdFileEntry?,alixdProcessEntry?,
    alixdFilterEntry?,alixdFilterElemEntry?,alixdActionEntry?,alixdErrlogEntry?,
    alixdErrlogElemEntry?)>

    <!ELEMENT alixd EMPTY>
    <!ATTLIST alixd

        alixdTraceLevel (none|requests|functions|details) #IMPLIED
        alixdSaveConfigFlag (true|false) #IMPLIED
        alixdSaveConfigWhenExitFlag (true|false) #IMPLIED
        alixdOldPasswd CDATA #IMPLIED
        alixdNewPasswd CDATA #IMPLIED
        alixdPollingStatus (enabled|disabled) #IMPLIED
        alixdMaxTrapsPerMinut CDATA #IMPLIED
        alixdPath CDATA #IMPLIED
        alixActionLogType (overwrite|append|separate) #IMPLIED
    >

    <!ELEMENT alixdSourcesEntry (InstanceSourcesEntry)+>
    <!ELEMENT InstanceSourcesEntry EMPTY>
    <!ATTLIST InstanceSourcesEntry
        alixdSourcesName CDATA #REQUIRED
        alixdSourcesStatus (enabled|disabled|inError|pending) #IMPLIED
        alixdSourcesErrorInformation CDATA #IMPLIED
        alixdSourcesFrequency CDATA #IMPLIED
        alixdSourcesHighWaterMark CDATA #IMPLIED
        alixdSourcesLowWaterMark CDATA #IMPLIED
        alixdSourcesFile CDATA #IMPLIED
        alixdSourcesType (swap|filesystems|logFiles|fileGroup|processGroup|syslog|
            errlog|printers|program) #IMPLIED
        alixdSourcesSpecificName CDATA #IMPLIED
        alixdSourcesFilterName CDATA #IMPLIED
        alixdSourcesDefaultAction CDATA #IMPLIED
        alixdSourcesPasswd CDATA #IMPLIED
    >
```

```
<!ELEMENT alixdFileEntry (InstanceFileEntry)+>
<!ELEMENT InstanceFileEntry EMPTY>
<!ATTLIST InstanceFileEntry
    alixdFileGroupName CDATA #REQUIRED
    alixdFileIndex CDATA #REQUIRED
    alixdFileName CDATA #IMPLIED
    alixdFileDoesExist (true| false) #IMPLIED
    alixdFileSize CDATA #IMPLIED
    alixdFileMaxSize CDATA #IMPLIED
    alixdFileNormalSize CDATA #IMPLIED
>

<!ELEMENT alixdProcessEntry (InstanceProcessEntry)+>
<!ELEMENT InstanceProcessEntry EMPTY>
<!ATTLIST InstanceProcessEntry
    alixdPrGroupName CDATA #REQUIRED
    alixdPrProcessName CDATA #REQUIRED
    alixdPrProcessNumber CDATA #IMPLIED
    alixdPrPids CDATA #IMPLIED
    alixdPrMaxNumber CDATA #IMPLIED
    alixdPrMinNumber CDATA #IMPLIED
>

<!ELEMENT alixdFilterEntry (InstanceFilterEntry)+>
<!ELEMENT InstanceFilterEntry EMPTY>
<!ATTLIST InstanceFilterEntry
    alixdFilterName CDATA #REQUIRED
    alixdFilterType (lineMessage|separatorWithRegularExpressions|
        separatorWithAwkExpressions|ExtractMessagesWithAwk|
        ExtractMessagesWithProgram) #IMPLIED
    alixdFilterStartMessage CDATA #IMPLIED
    alixdFilterEndMessage CDATA #IMPLIED
    alixdFilterProgramName CDATA #IMPLIED
    alixdFilterStatus (ok|inError) #IMPLIED
    alixdFilterErrorInformation CDATA #IMPLIED
>

<!ELEMENT alixdFilterElemEntry (InstanceFilterElemEntry)+>
<!ELEMENT InstanceFilterElemEntry EMPTY>
<!ATTLIST InstanceFilterElemEntry
    alixdFilterElemFilterName CDATA #REQUIRED
    alixdFilterElemIndex CDATA #REQUIRED
    alixdFilterElemStatus (normal|overThreshold) #IMPLIED
```

```
    alixdFilterElemType CDATA #FIXED "regularExpression"
    alixdFilterElemExpression CDATA #IMPLIED
    alixdFilterElemCount CDATA #IMPLIED
    alixdFilterElemThreshold CDATA #IMPLIED
    alixdFilterElemDuration CDATA #IMPLIED
    alixdFilterElemLastMatchTime CDATA #IMPLIED
    alixdFilterElemLastMessage CDATA #IMPLIED
    alixdFilterElemTriggering (overMatch|overThreshold) #IMPLIED
    alixdFilterElemAction CDATA #IMPLIED
    alixdFilterElemSeverity (indeterminate| critical |major|minor|warning|clear) #
        IMPLIED
>

<!ELEMENT alixdActionEntry (InstanceActionEntry)+>
<!ELEMENT InstanceActionEntry EMPTY>
<!ATTLIST InstanceActionEntry
    alixdActionName CDATA #REQUIRED
    alixdActionType (doNothing|sendTrap|executeProgram) #IMPLIED
    alixdActionDescription CDATA #IMPLIED
    alixdActionProgramName CDATA #IMPLIED
    alixdActionArguments CDATA #IMPLIED
    alixdActionPasswd CDATA #IMPLIED
    alixdActionDoesSendTrap (true|false) #IMPLIED
>

<!ELEMENT alixdErrlogEntry (InstanceErrlogEntry)+>
<!ELEMENT InstanceErrlogEntry EMPTY>
<!ATTLIST InstanceErrlogEntry
    alErrlogClass (hard|soft|other|pend) #REQUIRED
     alErrlogIdentifier  CDATA #REQUIRED
    alErrlogLabel CDATA #IMPLIED
    alErrlogType (pend|perf|perm|temp|unkn|info) #IMPLIED
    alErrlogDescription CDATA #IMPLIED
>

<!ELEMENT alixdErrlogElemEntry (InstanceErrlogElemEntry)+>
<!ELEMENT InstanceErrlogElemEntry EMPTY>
<!ATTLIST InstanceErrlogElemEntry
    alixdErrlogElemId CDATA #REQUIRED
    alixdErrlogElemStatus (normal|overThreshold) #IMPLIED
    alixdErrlogElemCount CDATA #IMPLIED
    alixdErrlogElemThreshold CDATA #IMPLIED
    alixdErrlogElemDuration CDATA #IMPLIED
```

```
        alixdErrlogElemLastMatchTime CDATA #IMPLIED
        alixdErrlogElemLastMessage CDATA #IMPLIED
        alixdErrlogElemTriggering (everyTimes|overThreshold) #IMPLIED
        alixdErrlogElemAction CDATA #IMPLIED
    >
```

A.2 Agent d'extension

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT OAXconfig (generix,generixPackageEntry?,generixTableEntry?,
    generixAttributeEntry?,generixDataEntry?)>

    <!ELEMENT generix EMPTY>
    <!ATTLIST generix

        generixTraceLevel (none|requests|functions|details) #IMPLIED
        generixSaveConfigFlag (true|false) #IMPLIED
        generixSaveConfigWhenExitFlag (true|false) #IMPLIED
        generixOldPasswd CDATA #IMPLIED
        generixNewPasswd CDATA #IMPLIED
        generixPath CDATA #IMPLIED
    >

    <!ELEMENT generixPackageEntry (InstancePackageEntry)+>
    <!ELEMENT InstancePackageEntry EMPTY>
    <!ATTLIST InstancePackageEntry

        generixPackageName CDATA #REQUIRED
        generixPackageStatus (ok|notAvailable|inError) #IMPLIED
        generixPackageErrorInformation CDATA #IMPLIED
    >

    <!ELEMENT generixTableEntry (InstanceTableEntry)+>
    <!ELEMENT InstanceTableEntry EMPTY>
    <!ATTLIST InstanceTableEntry

        generixTableId CDATA #REQUIRED
        generixTableDescription CDATA #IMPLIED
        generixTableType (process|file) #IMPLIED
        generixTableCommand CDATA #IMPLIED
        generixTableStderr (stderrOnNull|stderrOnStdout) #IMPLIED
        generixTableFile CDATA #IMPLIED
        generixTableValidityDuration CDATA #IMPLIED
```

```
    generixTableErrorInformation CDATA #IMPLIED
    generixTablePasswd CDATA #IMPLIED
    generixTableStartLine CDATA #IMPLIED
    generixTableBeginListMeta CDATA #IMPLIED
    generixTableEndListMeta CDATA #IMPLIED
    generixTableEscapeMeta CDATA #IMPLIED
    generixTableCommentChar CDATA #IMPLIED
    generixTableSeparators CDATA #IMPLIED
    generixTableNullField CDATA #IMPLIED
    generixTableFieldNumber CDATA #IMPLIED
    generixTableFilterValue CDATA #IMPLIED
>

<!ELEMENT generixAttributeEntry (InstanceAttributeEntry)+>
<!ELEMENT InstanceAttributeEntry EMPTY>
<!ATTLIST InstanceAttributeEntry

    generixAttrTableId CDATA #REQUIRED
    generixAttrColumnNumber CDATA #REQUIRED
    generixAttrOid CDATA #IMPLIED
    generixAttrSyntax (Integer|octetString|ipAddress|counter|gauge|timeTicks|
        objectId) #IMPLIED
    generixAttrPasswd CDATA #IMPLIED
    generixAttrName CDATA #IMPLIED
>

<!ELEMENT generixDataEntry (InstanceDataEntry)+>
<!ELEMENT InstanceDataEntry EMPTY>
<!ATTLIST InstanceDataEntry

    generixDataTableId CDATA #REQUIRED
    generixDataName CDATA #REQUIRED
    generixDataLineCount CDATA #IMPLIED
    generixData1 CDATA #IMPLIED
    generixData2 CDATA #IMPLIED
    generixData3 CDATA #IMPLIED
    generixData4 CDATA #IMPLIED
    generixData5 CDATA #IMPLIED
    generixData6 CDATA #IMPLIED
    generixData7 CDATA #IMPLIED
    generixData8 CDATA #IMPLIED
    generixData9 CDATA #IMPLIED
    generixData10 CDATA #IMPLIED
```

```
        generixData11 CDATA #IMPLIED
        generixData12 CDATA #IMPLIED
        generixData13 CDATA #IMPLIED
        generixData14 CDATA #IMPLIED
        generixData15 CDATA #IMPLIED
        generixData16 CDATA #IMPLIED
        generixData17 CDATA #IMPLIED
        generixData18 CDATA #IMPLIED
        generixData19 CDATA #IMPLIED
        generixData20 CDATA #IMPLIED
        generixData21 CDATA #IMPLIED
    >
```

A.3 Agent COACH

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT COACHconfig (confmod.ini,aic.inst)>

<!ELEMENT confmod.ini (filter*,indicator*)>

<!ELEMENT indicator EMPTY>
<!ATTLIST indicator
    cfgIndicatorId CDATA #REQUIRED
    cfgIndicatorLabel CDATA #REQUIRED
    cfgIndicatorDomain CDATA #REQUIRED
    cfgIndicatorEquation CDATA #REQUIRED
    cfgIndicatorPeriodPolling CDATA #REQUIRED
    cfgIndicatorThreshold CDATA #REQUIRED
    cfgIndicatorCptMax CDATA #REQUIRED
    cfgIndicatorPeriodValid CDATA #REQUIRED
    cfgIndicatorComparaison (less|greater|equal| different ) #REQUIRED
    cfgIndicatorLogOn (on|off) #REQUIRED
    cfgIndicatorTrap (aicTrapCritical|aicTrapMajor|aicTrapMinor|aicTrapWarning|
        aicTrapClear|aicAutoTrapCritical|aicAutoTrapMajor|aicAutoTrapMinor|
        aicAutoTrapWarning|aicAutoTrapClear|aicMibTrapCritical|aicMibTrapMajor|
        aicMibTrapMinor|aicMibTrapWarning|aicMibTrapClear) #REQUIRED
>

<!ELEMENT filter EMPTY>
<!ATTLIST filter
    cfgFilterId CDATA #REQUIRED
    cfgFilterLabel CDATA #REQUIRED
    cfgFilterDomain CDATA #REQUIRED
```

```
    cfgFilterEnterprise  CDATA #REQUIRED
    cfgFilterGeneric  CDATA #REQUIRED
    cfgFilterSpecific  CDATA #REQUIRED
    cfgFilterCptMax CDATA #REQUIRED
    cfgFilterPeriodValid  CDATA #REQUIRED
>

<!ELEMENT aic.inst (cfgDiscovery,cfgDomain+,cfgSystem?)>

<!ELEMENT cfgDiscovery (cfgDiscoverIpEntry*,cfgDiscoverIpSetEntry*)>
<!ATTLIST cfgDiscovery
    cfgDiscoverId CDATA #FIXED "0"
    cfgDiscoverPeriod CDATA #REQUIRED
    cfgDiscoverSubnetMask CDATA #REQUIRED
    cfgDiscoverDomainAtNextTime (yes|no) #REQUIRED
    cfgDiscoverTimeOut CDATA #REQUIRED
    cfgDiscoverRedoPeriod CDATA #REQUIRED
>

<!ELEMENT cfgDiscoverIpEntry EMPTY>
<!ATTLIST cfgDiscoverIpEntry
    cfgDiscoverIpIndex CDATA #REQUIRED
    cfgDiscoverIp CDATA #REQUIRED
    cfgDiscoverIpDomain CDATA #IMPLIED
>

<!ELEMENT cfgDiscoverIpSetEntry EMPTY>
<!ATTLIST cfgDiscoverIpSetEntry
    cfgDiscoverIpSetIndex CDATA #REQUIRED
    cfgDiscoverIpLower CDATA #REQUIRED
    cfgDiscoverIpUpper CDATA #REQUIRED
>

<!ELEMENT cfgDomain (cfgAtt1,cfgAtt2?,cfgAtt3?,cfgAtt4?,cfgAtt5?)>
<!ATTLIST cfgDomain
    cfgDomainId CDATA #REQUIRED
    cfgDomainLabel CDATA #REQUIRED
>

<!ELEMENT cfgAtt1 EMPTY>
<!ATTLIST cfgAtt1
    cfgAtt1 CDATA #REQUIRED
    cfgAtt1Value CDATA #REQUIRED
```

```
    cfgAtt1Comparaison (exist|equal|greater| different |lessequal |greaterequal| less |
        include|notinclude|node) #REQUIRED
>
<!ELEMENT cfgAtt2 EMPTY>
<!ATTLIST cfgAtt2
    cfgAtt2 CDATA #REQUIRED
    cfgAtt2Value CDATA #REQUIRED
    cfgAtt2Comparaison (exist|equal|greater| different |lessequal |greaterequal| less |
        include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt3 EMPTY>
<!ATTLIST cfgAtt3
    cfgAtt3 CDATA #REQUIRED
    cfgAtt3Value CDATA #REQUIRED
    cfgAtt3Comparaison (exist|equal|greater| different |lessequal |greaterequal| less |
        include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt4 EMPTY>
<!ATTLIST cfgAtt4
    cfgAtt4 CDATA #REQUIRED
    cfgAtt4Value CDATA #REQUIRED
    cfgAtt4Comparaison (exist|equal|greater| different |lessequal |greaterequal| less |
        include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgAtt5 EMPTY>
<!ATTLIST cfgAtt5
    cfgAtt5 CDATA #REQUIRED
    cfgAtt5Value CDATA #REQUIRED
    cfgAtt5Comparaison (exist|equal|greater| different |lessequal |greaterequal| less |
        include|notinclude|node) #REQUIRED
>

<!ELEMENT cfgSystem EMPTY>
<!ATTLIST cfgSystem
    cfgSystemId CDATA #FIXED "0"
    systemSaveConfiguration (on|off) #IMPLIED
    detachTrapLogs (on|off) #IMPLIED
    detachIndicatorLogs (on|off) #IMPLIED
    logOnTraps (on|off) #IMPLIED
    logOnIndicators (on|off) #IMPLIED
```

```
    indicatorMaxTimePolling CDATA #IMPLIED
    indicatorPollingUnitPeriod CDATA #IMPLIED
    systemSnmpCommunity CDATA #IMPLIED
>
```

B Parseur Perl utilisant les DTD

B.1 XMLParseConfig.pl avec DOM

```
#!/usr/local/bin/perl -w

use XML::Parser::Expat;
use XML::Checker::Parser;
use XML::DOM;
use MyHandler;
use IO::Handle;

#############
## Main Procedure ##
#############

my $type="UNDEFINED";
my $path;

print "Checking parameter(s)... ";
if($#ARGV==0 && substr($ARGV[0],0,1) ne "-")
    {print "OK\n";$path=setOutput("./");parser();}
elsif ($#ARGV==1 && substr($ARGV[0],0,3) eq "-o=" && substr($ARGV[1],0,1) ne
    "-")
    {print "OK\n";$path=setOutput(substr($ARGV[0],3));parser();}
else
    {print "unknown or missing parameter(s) detected\n";HelpMessage();}

#############
## Parsing Procedure ##
#############

sub parser
    {
    my $parser = new XML::Checker::Parser(SkipInsignifWS => 1,ParseParamEnt
        => 1,SkipExternalDTD => 1);
    local $XML::Checker::FAIL;
    print "Searching agent type ... ";
    my $expatParser = new XML::Parser::Expat;
    $expatParser->setHandlers(Doctype=>\&returnDoctype);
    eval{$expatParser->parsefile($ARGV[$#ARGV]);};
    if($@){print "\nHmmm... this file seems not well-formed... problems could occur
        during the generation phase !... ";}
    if($type eq "OAC")
```

```
        {
        print "Control agent detected\n";
        $XML::Checker::FAIL=\&OACErrorHandler;
        $parser->setHandlers(Start=>\&OACstartHandler,End=>\&OACendHandler,
            Init=>\&OACinitHandler,Final=>\&OACfinalHandler);}
    elsif ($type eq "OAX")
        {
        print "Extension agent detected\n";
        $XML::Checker::FAIL=\&OAXErrorHandler;
        $parser->setHandlers(Start=>\&OAXstartHandler,End=>\&OAXendHandler,
            Init=>\&OAXinitHandler,Final=>\&OAXfinalHandler);}
    elsif ($type eq "COACH")
        {
        print "Coach agent detected\n";
        $XML::Checker::FAIL=\&COACHErrorHandler;
        $parser->setHandlers(Start=>\&COACHstartHandler,End=>\&
            COACHendHandler,Init=>\&COACHinitHandler,Final=>\&
            COACHfinalHandler);
        }
    elsif ($type eq "UNDEFINED")
        {die "\nDOCTYPE tag not defined or file not found...\n";}
    else
        {die "\nAgent type not recognized !\n";}
    eval{$parser->parsefile($ARGV[$#ARGV]);};
    if ($@)
        {errorProcedure($@);}
    }

#############
## Help Procedure ##
#############

sub HelpMessage
    {
    print "\nUsage : perl XMLParseConfig.pl [option] file\n\nThis program parses and
        generates configuration file for OpenMaster Agents.\n";
    print "\nOption:\n\t-o=xxx Define the directory destination of the output(s) file (
        s)\n";
    print "\n --> The current directory is used by default and all others output
        directories specified must exist\n";
    print "\n --> Type of agent is automatically detected\n\n";
    }
```

```
######################
## agent type detection procedure ##
######################

sub returnDoctype
    {
    my ($expat,$name,$sysid,$pubid,$internal)=@_;

    my @tmp=split('/',$sysid);
    if ($tmp[$#tmp] eq "OACconfig.dtd")
        {$type="OAC";}
    elsif ($tmp[$#tmp] eq "OAXconfig.dtd")
        {$type="OAX";}
    elsif ($tmp[$#tmp] eq "COACHconfig.dtd")
        {$type="COACH";}
    else
        {$type="OTHER";}
    }

sub errorProcedure
    {
    my ($error)=@_;

    print "\n$error\n";
    if ($type eq "OAC")
        {unlink($path."alixd.conf.dyn");}
    elsif ($type eq "OAX")
        {unlink($path."generixd.conf.dyn");}
    elsif ($type eq "COACH")
        {unlink($path."confmod.ini");unlink($path."aic.inst");}
    }
```

B.2 MyHandler.pm avec DOM

```
#################################
## Perl module for XML Open Agent configuration files ##
#################################

my $treeDoc;
my $indicator_index;
my $filter_index ;
my $output_directory;
my $ind_cfgAtt;
my $cfgAttseq;
```

```
sub built_structure
    {
    my ($element)=@_;
    my %index=();
    my $tmp = $treeDoc->getElementsByTagName($element)->item(0);
    foreach $pos ($tmp->getChildNodes)
        {$index{$pos->getAttributes->getNamedItem("pos")->getValue}=$pos->
            getNodeName unless $pos->getNodeName eq "#text";}
    return %index;
    }

sub setOutput
    {
    my ($directory)=@_;

    $output_directory=$directory;
    if(chop($directory) ne '/')
        {$output_directory.='/';}
    return $output_directory;
    }

##################################
## functions checking validity of attributes 'value' ##
##################################

sub inf
    {
    my ($borne,$valeur,$nom,$output)=@_;
    if($valeur < $borne)
        {
        close CONFIG;
        unlink($output);
        die "Invalid value : '$valeur' < '$borne' for the attribute '$nom'\n";
        }
    }

sub sup
    {
    my ($borne,$valeur,$nom,$output)=@_;
    if($valeur > $borne)
        {
        close CONFIG;
```

```
        unlink($output);
        die "Invalid value : '$valeur' > '$borne' for the attribute '$nom'\n";
        }
    }

sub checkValue
    {
    my ($node,$valeur,$nom,$file)=@_;

    my $valeur_int;
    if ($node->hasChildNodes == 1)
        {
        my $indice_max = $node->getChildNodes->getLength;
        my $val = $node->getFirstChild;
        for ($j=0;$j<$indice_max;$j++)
            {
            if ($val->getNodeName eq $valeur)
                {
                $valeur_int =$val->getFirstChild->getNodeValue;
                $j=$indice_max+1;
                }
            else
                {$val=$val->getNextSibling;}
            }
        }
    else
        {
        my $inf = $node->getAttributes->getNamedItem("inf");
        my $sup = $node->getAttributes->getNamedItem("sup");
        inf($inf->getValue,$valeur,$nom,$file) unless !defined($inf);
        sup($sup->getValue,$valeur,$nom,$file) unless !defined($sup);
        $valeur_int =$valeur;
        }
    return $valeur_int;
    }

##############################
## Functions used by AOC and AOX agents ##
##############################

sub initHandler
    {
    my ($filename)=@_;
```

```
    my $Domparser = new XML::DOM::Parser(ParseParamEnt => 0,SkipExternalDTD
        => 1);
    print "Reading configuration file ... ";
    $treeDoc = $Domparser->parsefile("data/config.xml");
    print "Done\n";
    print "Parsing file in progress ... ";
    my $path=$output_directory.$filename;
    open(CONFIG,">$path") || die "Error in opening file '$path' : $!\n";
    }

sub finalHandler
    {
    close CONFIG;
    print"\nParsing terminated with success...\n";
    }

sub startHandler
    {
    my ($prefix,$filename,$p,$element,%attributs)=@_;
    if ($element eq $prefix)
        {
        print CONFIG "CLASS $element {\n";
        while(($nom,$valeur)=each(%attributs))
            {
            my $indice_max = $treeDoc->getElementsByTagName($element)->item
                (0)->getChildNodes->getLength;
            my $node = $treeDoc->getElementsByTagName($element)->item(0)->
                getFirstChild;
            for ($i=0;$i<$indice_max;$i++)
                {
                if ($node->getNodeName eq $nom)
                    {
                    $i=$indice_max+1;
                    print "#";STDOUT->autoflush(1);
                    print CONFIG "\t $nom = ".checkValue($node,$valeur,$nom,
                        $output_directory.$filename)." ;\n";
                    }
                else
                    {$node = $node->getNextSibling;}
                }
            }
        }
    elsif (substr($element,0,5) eq $prefix)
```

```
    {print CONFIG "CLASS $element {\n";}
elsif (substr($element,0,8) eq "Instance" )
    {
    my %index=();
    my $node = $treeDoc->getElementsByTagName($element."Id")->item(0);
    foreach $id ($node->getChildNodes)
        {$index{$id->getAttributes->getNamedItem("id")->getValue}=$id->
            getNodeName unless $id->getNodeName eq "#text";}
    my $indice=1;
    my $cle="\"";
    $reference="$indice";
    while( exists $index{$reference})
        {
        $cle.=checkValue($treeDoc->getElementsByTagName($index{$reference})
            ->item(0),$attributs{$index{$reference}},$index{$reference},
            $output_directory.$filename).'.';
        delete $attributs{$index{$reference}};
        $indice++;
        $reference="$indice";
        }
    chop($cle);
    print CONFIG "\tINSTANCE $cle\" {\n";
    while(($nom,$valeur)=each(%attributs))
        {
        my $indice_max = $treeDoc->getElementsByTagName($element)->item
            (0)->getChildNodes->getLength;
        my $node = $treeDoc->getElementsByTagName($element)->item(0)->
            getFirstChild;
        for ($i=0;$i<$indice_max;$i++)
            {
            if ($node->getNodeName eq $nom)
                {
                $i=$indice_max+1;
                print "#";STDOUT->autoflush(1);
                print CONFIG "\t $nom = ".checkValue($node,$valeur,$nom,
                    $output_directory.$filename)." ;\n";
                }
            else
                {$node = $node->getNextSibling;}
            }
        }
    }
}
```

```
sub endHandler
    {
    my ($prefix,$p,$element)=@_;
    if(substr($element,0,5) eq $prefix)
        {print CONFIG "\t}\n\n";}
     elsif (substr($element,0,8) eq "Instance")
        {print CONFIG "\t }\n\n";}
    }

sub ErrorHandler
    {
    my $code=shift;
    if($code < 200)
        {
        close CONFIG;
        print "\n";
        die XML::Checker::error_string($code,@_);
        }
    }

############################
## Handling Procedures for OAC agents ##
############################

sub OACinitHandler
    {initHandler("alixd.conf.dyn");}

sub OACfinalHandler
    {finalHandler;}

sub OACstartHandler
    {startHandler("alixd","alixd.conf.dyn",@_);}

sub OACendHandler
    {endHandler("alixd",@_);}

sub OACErrorHandler
    {ErrorHandler(@_);}

############################
## Handling Procedures for OAX agents ##
############################
```

```
sub OAXinitHandler
    {initHandler("generixd.conf.dyn");}

sub OAXfinalHandler
    {finalHandler;}

sub OAXstartHandler
    {startHandler("generix","generixd.conf.dyn",@_);}

sub OAXendHandler
    {endHandler("generix",@_);}

sub OAXErrorHandler
    {ErrorHandler(@_);}

############################
## Handling procedures for COACH agents ##
############################

sub COACHinitHandler
    {
    my $Domparser = new XML::DOM::Parser(ParseParamEnt => 0,SkipExternalDTD
        => 1);
    print "Reading configuration file ... ";
    $treeDoc = $Domparser->parsefile("data/config.xml");
    print "Done\n";
    %indicator_index=built_structure("indicator");
    %filter_index=built_structure(" filter ");
    print "Parsing file in progress ... ";
    my $path=$output_directory."confmod.ini";
    open(CONFIG1,">$path") || die "Error in opening file '$path' : $!\n";
    $path=$output_directory."aic.inst";
    open(CONFIG2,">$path") || die "Error in opening file '$path' : $!\n";
    }

sub COACHfinalHandler
    {
    close CONFIG1;
    close CONFIG2;
    print"\nParsing terminated with success...\n";
    }
```

```
sub COACHstartHandler
    {
    my ($p,$element,%attributs)=@_;
    if ($element eq "indicator " || $element eq " filter ")
        {
        my $index;
        my $line;
        if ($element eq "indicator")
            {%index=%indicator_index;$line="IND ";}
        else
            {%index=%filter_index;$line="FIL ";}
        my $indice=1;
        my $reference="$indice";
        my $indice_max = $treeDoc->getElementsByTagName($element)->item(0)
            ->getChildNodes->getLength;
        while( exists  $index{$reference})
            {
            my $node = $treeDoc->getElementsByTagName($element)->item(0)->
                getFirstChild;
            for ($i=0;$i<$indice_max;$i++)
                {
                if ($node->getNodeName eq $index{$reference})
                    {
                    $i=$indice_max+1;
                    $line.=checkValue($node,$attributs{$index{$reference}},$index{
                        $reference},$output_directory."confmod.ini") .' ';
                    }
                else
                    {$node = $node->getNextSibling;}
                }
            $indice++;
            $reference="$indice";
            }
        print "#";STDOUT->autoflush(1);
        print  CONFIG1 "$line\n";
        }
     elsif ($element eq "cfgDiscovery" || $element eq "cfgDomain" || $element eq "
        cfgSystem" || $element eq "cfgDiscoverIpEntry" || $element eq "
        cfgDiscoverIpSetEntry" || substr($element,0,6) eq "cfgAtt")
        {
        my $ind=$ind_cfgAtt;
        my $indice_max = $treeDoc->getElementsByTagName($element)->item(0)
            ->getChildNodes->getLength;
```

```
my $node = $treeDoc->getElementsByTagName($element)->item(0)->
    getFirstChild;
if(substr($element,0,6) ne "cfgAtt")
    {
    $cfgAttseq=1;
    for($i=0;$i<$indice_max;$i++)
        {
        if(defined($node->getAttributes) && defined($node->getAttributes
            ->getNamedItem("id")))
            {
            $ind=checkValue($node,$attributs{$node->getNodeName},$node
                ->getNodeName,$output_directory."aic.inst");
            $ind_cfgAtt=$ind;
            delete $attributs{$node->getNodeName};
            $i=$indice_max+1;
            }
        else
            {$node = $node->getNextSibling;}
        }
    }
elsif ($element eq "cfgAtt".$cfgAttseq)
    {$cfgAttseq++;}
else
    {die "\nThe attribute $element doesn't respect order sequence in
        cfgDomainId = $ind !\n";}
$indice_max = $treeDoc->getElementsByTagName($element)->item(0)->
    getChildNodes->getLength;
while(($nom,$valeur)=each(%attributs))
    {
    my $node = $treeDoc->getElementsByTagName($element)->item(0)->
        getFirstChild;
    for($i=0;$i<$indice_max;$i++)
        {
        if($node->getNodeName eq $nom)
            {
            $i=$indice_max+1;
            print "#";STDOUT->autoflush(1);
            print CONFIG2 "$nom.$ind ".checkValue($node,$valeur,$nom,
                $output_directory."aic.inst")."\n";
            }
        else
            {$node = $node->getNextSibling;}
        }
```

```
            }
        }
    }

sub COACHendHandler
{}

sub COACHErrorHandler
    {
    my $code=shift;
    if ($code < 200)
        {
        close CONFIG1;
        close CONFIG2;
        print "\n";
        die XML::Checker::error_string($code,@_);
        }
    }

##############
## End of Module ##
##############

1;
```

B.3 XMLParseConfig.pl avec SAX

```
#!/usr/local/bin/perl -w

use XML::Parser::PerlSAX;
use XML::Checker::Parser;
use MyHandler;

##############
## Main Procedure ##
##############

my $type;
if ($#ARGV==0 && $ARGV[0] eq "-?")
    {HelpMessage();}
elsif ($#ARGV==1 && substr($ARGV[0],0,3) eq "-t=" && substr($ARGV[1],0,1) ne
    "-")
    {$type=substr($ARGV[0],3);parser();}
elsif ($#ARGV==2 && substr($ARGV[0],0,3) eq "-t=" && $ARGV[1] eq "-v" &&
```

```
    substr($ARGV[2],0,1) ne "-")
    {$type=substr($ARGV[0],3);valid();parser();}
elsif ($#ARGV==2 && substr($ARGV[1],0,3) eq "-t=" && $ARGV[0] eq "-v" &&
    substr($ARGV[2],0,1) ne "-")
    {$type=substr($ARGV[1],3);valid();parser();}
else
    {HelpMessage();}

################
## Validation Procedure ##
################

sub valid
    {
    print "Validation of file $ARGV[$#ARGV]...\n";
    my $parseur = new XML::Checker::Parser(SkipInsignifWS => 1,ParseParamEnt
        => 1,SkipExternalDTD => 1);
    local $XML::Checker::FAIL=\&myErrorHandler;
    $parseur->parsefile($ARGV[$#ARGV]);
    print "Validation ended without serious problems...\n";
    }

###############
## Parsing Procedure ##
###############

sub parser
    {
    my $handler;
    if ($type eq "OAC")
        {$handler = MyHandler->new;}
    elsif ($type eq "OAX" || $type eq "COACH")
        {die "Handler not available !";}
    else
        {die "Bad agent type !";}
    print "Creating File ...\ n";
    my $parser = XML::Parser::PerlSAX->new(Handler=> $handler);
    $parser->parse(Source => {SystemId => $ARGV[$#ARGV]});
    print "File created and parsing terminated with success...\ n";
    }
```

```
############
## Error Handler ##
############

sub myErrorHandler
    {
    my $code=shift;
    die XML::Checker::error_string($code,@_) if $code < 200;
    XML::Checker::print_error($code,@_);
    }

############
## Help Procedure ##
############

sub HelpMessage
    {
    print "\nUsage : perl XMLParseConfig.pl [option] file\nThis program parses and
        generates configuration file for OpenMaster Agents.\n";
    print "\nOption:\n\t-?      Show this help\n\t-v      Perform a validation of the
        document (recommended)\n\t-t=xxx Define the document type to be parsed [
        OAC|OAX|COACH]\n";
    print "\nDocument type must be provided explicitly : no default value available !\
        n\n";
    }
```

B.4 MyHandler.pm avec SAX

```
package MyHandler;

sub new
    {
    my ($type)=@_;
    return bless {},$type;
    }

sub start_document
    {
    open(CONFIG,'> config.dat') || die "Error in opening file : $!\n";
    print CONFIG "#\n# Open Agent Control Configuration\n#\n";
    }

sub start_element
    {
```

```
    my ($self,$element)=@_;
    if($element->{Name} eq "alixd")
        {
        print CONFIG "CLASS $element->{Name} {\n";
        my %t=%{$element->{Attributes}};
        while(( $attributs,$valeurs)=each(%t))
            {print CONFIG "\t$attributs = $valeurs ;\n";}
        }

    elsif (substr($element->{Name},0,5) eq "alixd")
        {
        print CONFIG "CLASS $element->{Name} {\n";
        }
    elsif (substr($element->{Name},0,5) eq "Index")
        {
        print CONFIG "\tINSTANCE \"";
        my $tmp="";
        my %t=%{$element->{Attributes}};
        while(( $attributs,$valeurs)=each(%t))
            {$tmp.="$valeurs.";}
        chop($tmp);
        print CONFIG "$tmp\" {\n";
        }
    else
        {
        if(substr($element->{Name},0,8) eq "Instance")
            {
            my %t=%{$element->{Attributes}};
            while(( $attributs,$valeurs)=each(%t))
                {print CONFIG "\t\t$attributs = $valeurs ;\n";}
            }
        }
    }

sub end_element
    {
    my ($self,$element)=@_;
    if(substr($element->{Name},0,5) eq "alixd")
        {print CONFIG "\t}\n";}
    elsif (substr($element->{Name},0,5) eq "Index")
        {print CONFIG "\t\t}\n";}
    }
```

```
sub end_document
    {close CONFIG;}
1;
```

C Le fichier "config.xml"

```
<?xml version="1.0" encoding="UTF-8" standalone='yes'?>

<AgentTypes>

<OAC>
    <alixd>
        <alixdTraceLevel>
            <none>0</none>
            <requests>1</requests>
            <functions>2</functions>
            <details>3</details>
        </alixdTraceLevel>
        <alixdSaveConfigFlag>
            <true>1</true>
            <false>2</false>
        </alixdSaveConfigFlag>
        <alixdSaveConfigWhenExitFlag>
            <true>1</true>
            <false>2</false>
        </alixdSaveConfigWhenExitFlag>
        <alixdOldPasswd />
        <alixdNewPasswd />
        <alixdPollingStatus>
            <enabled>1</enabled>
            <disabled>2</disabled>
        </alixdPollingStatus>
        <alixdMaxTrapsPerMinut inf="0" sup="500"/>
        <alixdPath />
        <alixActionLogType>
            <overwrite>1</overwrite>
            <append>2</append>
            <separate>3</separate>
        </alixActionLogType>
    </alixd>

    <InstanceSourcesEntry>
        <InstanceSourcesEntryId>
            <alixdSourcesName id="1"/>
        </InstanceSourcesEntryId>
        <alixdSourcesStatus>
            <enabled>1</enabled>
```

```
            <disabled>2</disabled>
            <inError>3</inError>
            <pending>4</pending>
        </alixdSourcesStatus>
        <alixdSourcesErrorInformation />
        <alixdSourcesFrequency />
        <alixdSourcesHighWaterMark inf="0" sup="100"/>
        <alixdSourcesLowWaterMark inf="0" sup="100"/>
        <alixdSourcesFile />
        <alixdSourcesType>
            <swap>1</swap>
            <filesystems>2</filesystems>
            <logFiles>3</logFiles>
            <fileGroup>4</fileGroup>
            <processGroup>5</processGroup>
            <syslog>6</syslog>
            <errlog>7</errlog>
            <printers>8</printers>
            <program>12</program>
        </alixdSourcesType>
        <alixdSourcesSpecificName />
        <alixdSourcesFilterName />
        <alixdSourcesDefaultAction />
        <alixdSourcesPasswd />
    </InstanceSourcesEntry>

    <InstanceFileEntry>
        <InstanceFileEntryId>
            <alixdFileGroupName id="1"/>
            <alixdFileIndex id="2"/>
        </InstanceFileEntryId>
        <alixdFileName />
        <alixdFileDoesExist>
            <true>1</true>
            <false>2</false>
        </alixdFileDoesExist>
        <alixdFileSize />
        <alixdFileMaxSize />
        <alixdFileNormalSize />
    </InstanceFileEntry>

    <InstanceProcessEntry>
        <InstanceProcessEntryId>
```

```
            <alixdPrGroupName id="1" />
            <alixdPrProcessName id="2" />
        </InstanceProcessEntryId>
        <alixdPrProcessNumber />
        <alixdPrPids />
        <alixdPrMaxNumber />
        <alixdPrMinNumber />
    </InstanceProcessEntry>

    <InstanceFilterEntry>
        <InstanceFilterEntryId>
            <alixdFilterName id="1" />
        </InstanceFilterEntryId>
        <alixdFilterType>
            <lineMessage>1</lineMessage>
            <separatorWithRegularExpressions>2</separatorWithRegularExpressions
                >
            <separatorWithAwkExpressions>3</separatorWithAwkExpressions>
            <ExtractMessagesWithAwk>4</ExtractMessagesWithAwk>
            <ExtractMessagesWithProgram>5</ExtractMessagesWithProgram>
        </alixdFilterType>
        <alixdFilterStartMessage />
        <alixdFilterEndMessage />
        <alixdFilterProgramName />
        <alixdFilterStatus>
            <ok>1</ok>
            <inError>2</inError>
        </alixdFilterStatus>
        <alixdFilterErrorInformation />
    </InstanceFilterEntry>

    <InstanceFilterElemEntry>
        <InstanceFilterElemEntryId>
            <alixdFilterElemFilterName id="1" />
            <alixdFilterElemIndex id="2" />
        </InstanceFilterElemEntryId>
        <alixdFilterElemStatus>
            <normal>1</normal>
            <overThreshold>2</overThreshold>
        </alixdFilterElemStatus>
        <alixdFilterElemType />
        <alixdFilterElemExpression />
        <alixdFilterElemCount />
```

```
    <alixdFilterElemThreshold />
    <alixdFilterElemDuration />
    <alixdFilterElemLastMatchTime />
    <alixdFilterElemLastMessage />
    <alixdFilterElemTriggering>
        <overMatch>1</overMatch>
        <overThreshold>2</overThreshold>
    </alixdFilterElemTriggering>
    <alixdFilterElemAction />
    <alixdFilterElemSeverity>
        <indeterminate>1</indeterminate>
        <critical>2</critical>
        <major>3</major>
        <minor>4</minor>
        <warning>5</warning>
        <clear>6</clear>
    </alixdFilterElemSeverity>
</InstanceFilterElemEntry>

<InstanceActionEntry>
    <InstanceActionEntryId>
        <alixdActionName id="1"/>
    </InstanceActionEntryId>
    <alixdActionType>
        <doNothing>1</doNothing>
        <sendTrap>2</sendTrap>
        <executeProgram>3</executeProgram>
    </alixdActionType>
    <alixdActionDescription />
    <alixdActionProgramName />
    <alixdActionArguments />
    <alixdActionPasswd />
    <alixdActionDoesSendTrap>
        <true>1</true>
        <false>2</false>
    </alixdActionDoesSendTrap>
</InstanceActionEntry>

<InstanceErrlogEntry>
    <InstanceErrlogEntryId>
        <alErrlogClass id="1">
            <hard>1</hard>
            <soft>2</soft>
```

```
                <other>3</other>
                <pend>4</pend>
            </alErrlogClass>
            <alErrlogIdentifier id="2"/>
        </InstanceErrlogEntryId>
        <alErrlogLabel />
        <alErrlogType>
            <pend>1</pend>
            <perf>2</perf>
            <perm>3</perm>
            <temp>4</temp>
            <unkn>5</unkn>
            <info>6</info>
        </alErrlogType>
        <alErrlogDescription />
    </InstanceErrlogEntry>

    <InstanceErrlogElemEntry>
        <InstanceErrlogElemEntryId>
            <alixdErrlogElemId id="1"/>
        </InstanceErrlogElemEntryId>
        <alixdErrlogElemStatus>
            <normal>1</normal>
            <overThreshold>2</overThreshold>
        </alixdErrlogElemStatus>
        <alixdErrlogElemCount />
        <alixdErrlogElemThreshold />
        <alixdErrlogElemDuration />
        <alixdErrlogElemLastMatchTime />
        <alixdErrlogElemLastMessage />
        <alixdErrlogElemTriggering>
            <everyTimes>1</everyTimes>
            <overThreshold>2</overThreshold>
        </alixdErrlogElemTriggering>
        <alixdErrlogElemAction />
    </InstanceErrlogElemEntry>
</OAC>

<OAX>
    <generix>
        <generixTraceLevel>
            <none>0</none>
            <requests>1</requests>
```

```
            <functions>2</functions>
            <details>3</details>
        </generixTraceLevel>
        <generixSaveConfigFlag>
            <true>1</true>
            <false>2</false>
        </generixSaveConfigFlag>
        <generixSaveConfigWhenExitFlag>
            <true>1</true>
            <false>2</false>
        </generixSaveConfigWhenExitFlag>
        <generixOldPasswd />
        <generixNewPasswd />
        <generixPath />
    </generix>

    <InstancePackageEntry>
        <InstancePackageEntryId>
            <generixPackageName id="1"/>
        </InstancePackageEntryId>
        <generixPackageStatus>
            <ok>1</ok>
            <notAvailable>2</notAvailable>
            <inError>3</inError>
        </generixPackageStatus>
        <generixPackageErrorInformation />
    </InstancePackageEntry>

    <InstanceTableEntry>
        <InstanceTableEntryId>
            <generixTableId id="1"/>
        </InstanceTableEntryId>
        <generixTableDescription />
        <generixTableType>
            <process>1</process>
            <file>2</file>
        </generixTableType>
        <generixTableCommand />
        <generixTableStderr>
            <stderrOnNull>1</stderrOnNull>
            <stderrOnStdout>2</stderrOnStdout>
        </generixTableStderr>
        <generixTableFile />
```

```
        <generixTableValidityDuration inf="11"/>
        <generixTableErrorInformation />
        <generixTablePasswd />
        <generixTableStartLine inf="1"/>
        <generixTableBeginListMeta />
        <generixTableEndListMeta />
        <generixTableEscapeMeta />
        <generixTableCommentChar />
        <generixTableSeparators />
        <generixTableNullField inf="0" sup="10"/>
        <generixTableFieldNumber />
        <generixTableFilterValue inf="0" sup="64"/>
    </InstanceTableEntry>

    <InstanceAttributeEntry>
        <InstanceAttributeEntryId>
            <generixAttrTableId id="1"/>
            <generixAttrColumnNumber id="2"/>
        </InstanceAttributeEntryId>
        <generixAttrOid />
        <generixAttrSyntax>
            <Integer>1</Integer>
            <octetString>2</octetString>
            <ipAddress>3</ipAddress>
            <counter>4</counter>
            <gauge>5</gauge>
            <timeTicks>6</timeTicks>
            <objectId>7</objectId>
        </generixAttrSyntax>
        <generixAttrPasswd />
        <generixAttrName />
    </InstanceAttributeEntry>

    <InstanceDataEntry>
        <generixDataTableId />
        <generixDataName />
        <generixDataLineCount />
        <generixData1 />
        <generixData2 />
        <generixData3 />
        <generixData4 />
        <generixData5 />
        <generixData6 />
```

```
        <generixData7 />
        <generixData8 />
        <generixData9 />
        <generixData10 />
        <generixData11 />
        <generixData12 />
        <generixData13 />
        <generixData14 />
        <generixData15 />
        <generixData16 />
        <generixData17 />
        <generixData18 />
        <generixData19 />
        <generixData20 />
        <generixData21 />
    </InstanceDataEntry>
</OAX>

<COACH>
    <indicator>
        <cfgIndicatorId pos="1"/>
        <cfgIndicatorLabel pos="2"/>
        <cfgIndicatorDomain pos="3"/>
        <cfgIndicatorEquation pos="4"/>
        <cfgIndicatorPeriodPolling pos="5"/>
        <cfgIndicatorThreshold pos="6"/>
        <cfgIndicatorCptMax pos="7" sup="20"/>
        <cfgIndicatorPeriodValid pos="8"/>
        <cfgIndicatorComparaison pos="9">
            <less>&lt;</less>
            <greater>&gt;</greater>
            <equal>=</equal>
            <different>!=</different>
        </cfgIndicatorComparaison>
        <cfgIndicatorLogOn pos="10">
            <on>LOG</on>
            <off>NLOG</off>
        </cfgIndicatorLogOn>
        <cfgIndicatorTrap pos="11">
            <aicTrapCritical>1</aicTrapCritical>
            <aicTrapMajor>2</aicTrapMajor>
            <aicTrapMinor>3</aicTrapMinor>
            <aicTrapWarning>4</aicTrapWarning>
```

```
            <aicTrapClear>5</aicTrapClear>
            <aicAutoTrapCritical>11</aicAutoTrapCritical>
            <aicAutoTrapMajor>12</aicAutoTrapMajor>
            <aicAutoTrapMinor>13</aicAutoTrapMinor>
            <aicAutoTrapWarning>14</aicAutoTrapWarning>
            <aicAutoTrapClear>15</aicAutoTrapClear>
            <aicMibTrapCritical>21</aicMibTrapCritical>
            <aicMibTrapMajor>22</aicMibTrapMajor>
            <aicMibTrapMinor>23</aicMibTrapMinor>
            <aicMibTrapWarning>24</aicMibTrapWarning>
            <aicMibTrapClear>25</aicMibTrapClear>
        </cfgIndicatorTrap>
    </indicator>

    <filter>
        <cfgFilterId pos="1"/>
        <cfgFilterLabel pos="2"/>
        <cfgFilterDomain pos="3"/>
        <cfgFilterEnterprise pos="4"/>
        <cfgFilterGeneric pos="5"/>
        <cfgFilterSpecific pos="6"/>
        <cfgFilterCptMax pos="7"/>
        <cfgFilterPeriodValid pos="8"/>
    </filter>

    <cfgDiscovery>
        <cfgDiscoverId id="0"/>
        <cfgDiscoverPeriod />
        <cfgDiscoverSubnetMask />
        <cfgDiscoverDomainAtNextTime>
            <yes>1</yes>
            <no>2</no>
        </cfgDiscoverDomainAtNextTime>
        <cfgDiscoverTimeOut />
        <cfgDiscoverRedoPeriod />
    </cfgDiscovery>

    <cfgDiscoverIpEntry>
        <cfgDiscoverIpIndex id="0"/>
        <cfgDiscoverIp />
        <cfgDiscoverIpDomain />
    </cfgDiscoverIpEntry>
```

```
<cfgDiscoverIpSetEntry>
    <cfgDiscoverIpSetIndex id="0"/>
    <cfgDiscoverIpLower />
    <cfgDiscoverIpUpper />
</cfgDiscoverIpSetEntry>

<cfgSystem>
    <cfgSystemId id="0"/>
    <systemSaveConfiguration>
        <on>1</on>
        <off>2</off>
    </systemSaveConfiguration>
    <detachTrapLogs>
        <on>1</on>
        <off>2</off>
    </detachTrapLogs>
    <detachIndicatorLogs>
        <on>1</on>
        <off>2</off>
    </detachIndicatorLogs>
    <logOnTraps>
        <on>1</on>
        <off>2</off>
    </logOnTraps>
    <logOnIndicators>
        <on>1</on>
        <off>2</off>
    </logOnIndicators>
    <indicatorMaxTimePolling />
    <indicatorPollingUnitPeriod />
    <systemSnmpCommunity />
</cfgSystem>

<cfgDomain>
    <cfgDomainId id="0"/>
    <cfgDomainLabel />
</cfgDomain>

<cfgAtt1>
    <cfgAtt1 />
    <cfgAtt1Value />
    <cfgAtt1Comparaison>
        <exist>0</exist>
```

```
            <equal>1</equal>
            <greater>2</greater>
            <different>3</different>
            <lessequal>4</lessequal>
            <greaterequal>5</greaterequal>
            <less>6</less>
            <include>7</include>
            <notinclude>8</notinclude>
            <node>9</node>
        </cfgAtt1Comparaison>
    </cfgAtt1>

    <cfgAtt2>
        <cfgAtt2 />
        <cfgAtt2Value />
        <cfgAtt2Comparaison>
            <exist>0</exist>
            <equal>1</equal>
            <greater>2</greater>
            <different>3</different>
            <lessequal>4</lessequal>
            <greaterequal>5</greaterequal>
            <less>6</less>
            <include>7</include>
            <notinclude>8</notinclude>
            <node>9</node>
        </cfgAtt2Comparaison>
    </cfgAtt2>

    <cfgAtt3>
        <cfgAtt3 />
        <cfgAtt3Value />
        <cfgAtt3Comparaison>
            <exist>0</exist>
            <equal>1</equal>
            <greater>2</greater>
            <different>3</different>
            <lessequal>4</lessequal>
            <greaterequal>5</greaterequal>
            <less>6</less>
            <include>7</include>
            <notinclude>8</notinclude>
            <node>9</node>
```

```
        </cfgAtt3Comparaison>
    </cfgAtt3>

    <cfgAtt4>
        <cfgAtt4 />
        <cfgAtt4Value />
        <cfgAtt4Comparaison>
            <exist>0</exist>
            <equal>1</equal>
            <greater>2</greater>
            <different>3</different>
            <lessequal>4</lessequal>
            <greaterequal>5</greaterequal>
            <less>6</less>
            <include>7</include>
            <notinclude>8</notinclude>
            <node>9</node>
        </cfgAtt4Comparaison>
    </cfgAtt4>

    <cfgAtt5>
        <cfgAtt5 />
        <cfgAtt5Value />
        <cfgAtt5Comparaison>
            <exist>0</exist>
            <equal>1</equal>
            <greater>2</greater>
            <different>3</different>
            <lessequal>4</lessequal>
            <greaterequal>5</greaterequal>
            <less>6</less>
            <include>7</include>
            <notinclude>8</notinclude>
            <node>9</node>
        </cfgAtt5Comparaison>
    </cfgAtt5>
</COACH>

</AgentTypes>
```

D Schémas XML du W3C des agents OpenMaster

D.1 Agent de contrôle

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:annotation>
    <xsd:documentation xml:lang="en">
        XML schema for Control Open Agent
    </xsd:documentation>
</xsd:annotation>

<xsd:element name="OACconfig" type="OACconfigType">
   <xsd:annotation>
      <xsd:appinfo>alixd|alixdSourcesEntry|alixdFileEntry|alixdProcessEntry|
          alixdFilterEntry|alixdFilterElemEntry|alixdActionEntry|alixdErrlogEntry|
          alixdErrlogElemEntry</xsd:appinfo>
   </xsd:annotation>
</xsd:element>

<xsd:complexType name="OACconfigType">
    <xsd:sequence>
        <xsd:element ref="alixd"/>
        <xsd:element ref="alixdSourcesEntry" minOccurs="0"/>
        <xsd:element ref="alixdFileEntry" minOccurs="0"/>
        <xsd:element ref="alixdProcessEntry" minOccurs="0"/>
        <xsd:element ref="alixdFilterEntry" minOccurs="0"/>
        <xsd:element ref="alixdFilterElemEntry" minOccurs="0"/>
        <xsd:element ref="alixdActionEntry" minOccurs="0"/>
        <xsd:element ref="alixdErrlogEntry" minOccurs="0"/>
        <xsd:element ref="alixdErrlogElemEntry" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="alixd">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdTraceLevel" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="none"><xsd:appInfo>
                            alixdTraceLevel:0</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="requests"><xsd:appInfo>
                            alixdTraceLevel:1</xsd:appInfo></xsd:enumeration>
```

```
                <xsd:enumeration value="functions"><xsd:appInfo>
                    alixdTraceLevel:2</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="details"><xsd:appInfo>
                    alixdTraceLevel:3</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="alixdSaveConfigFlag" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="true"><xsd:appInfo>
                    alixdSaveConfigFlag:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="false"><xsd:appInfo>
                    alixdSaveConfigFlag:2</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="alixdSaveConfigWhenExitFlag" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="true"><xsd:appInfo>
                    alixdSaveConfigWhenExitFlag:1</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="false"><xsd:appInfo>
                    alixdSaveConfigWhenExitFlag:2</xsd:appInfo></xsd:
                    enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="alixdOldPasswd" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="alixdNewPasswd" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="alixdPollingStatus" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="enabled"><xsd:appInfo>
                    alixdPollingStatus:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="disabled"><xsd:appInfo>
                    alixdPollingStatus:2</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
```

```
            <xsd:element name="alixdMaxTrapsPerMinut" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:integer">
                        <xsd:minInclusive value="0"/>
                        <xsd:maxInclusive value="200"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="alixdPath" type="xsd:string" minOccurs="0"/>
            <xsd:element name="alixActionLogType" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="overwrite"><xsd:appInfo>
                            alixActionLogType:1</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="append"><xsd:appInfo>
                            alixActionLogType:2</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="separate"><xsd:appInfo>
                            alixActionLogType:3</xsd:appInfo></xsd:enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="alixdSourcesEntry" type="alixdSourcesEntryType"/>

<xsd:complexType name="alixdSourcesEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceSourcesEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceSourcesEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdSourcesName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
```

```
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="alixdSourcesStatus" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="enabled"><xsd:appInfo>
                    alixdSourcesStatus:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="disabled"><xsd:appInfo>
                    alixdSourcesStatus:2</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="inError"><xsd:appInfo>
                    alixdSourcesStatus:3</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="pending"><xsd:appInfo>
                    alixdSourcesStatus:4</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="alixdSourcesErrorInformation" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="alixdSourcesFrequency" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="alixdSourcesHighWaterMark" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="alixdSourcesLowWaterMark" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="alixdSourcesFile" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="alixdSourcesType" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="swap"><xsd:appInfo>
                    alixdSourcesType:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="filesystems"><xsd:appInfo>
                    alixdSourcesType:2</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="logFiles"><xsd:appInfo>
                    alixdSourcesType:3</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="fileGroup"><xsd:appInfo>
                    alixdSourcesType:4</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="processGroup"><xsd:appInfo>
                    alixdSourcesType:5</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="syslog"><xsd:appInfo>
                    alixdSourcesType:6</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="errlog"><xsd:appInfo>
```

```
                        alixdSourcesType:7</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="printers"><xsd:appInfo>
                        alixdSourcesType:8</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="program"><xsd:appInfo>
                        alixdSourcesType:12</xsd:appInfo></xsd:enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="alixdSourcesSpecificName" type="xsd:string"
            minOccurs="0"/>
        <xsd:element name="alixdSourcesFilterName" type="xsd:string"
            minOccurs="0"/>
        <xsd:element name="alixdSourcesDefaultAction" type="xsd:string"
            minOccurs="0"/>
        <xsd:element name="alixdSourcesPasswd" type="xsd:string" minOccurs
            ="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

<xsd:element name="alixdFileEntry" type="alixdFileEntryType"/>

<xsd:complexType name="alixdFileEntryType">
  <xsd:sequence>
    <xsd:element ref="InstanceFileEntry" minOccurs="1" maxOccurs="unbounded
      "/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceFileEntry">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="alixdFileGroupName">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="index" fixed="1"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="alixdFileIndex">
        <xsd:complexType>
```

```
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alixdFileName" type="xsd:string" minOccurs="0"/>
            <xsd:element name="alixdFileDoesExist" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="true"><xsd:appInfo>
                            alixdFileDoesExist:1</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="false"><xsd:appInfo>
                            alixdFileDoesExist:2</xsd:appInfo></xsd:enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="alixdFileSize" type="xsd:string" minOccurs="0"/>
            <xsd:element name="alixdFileMaxSize" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="alixdFileNormalSize" type="xsd:string" minOccurs
                ="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="alixdProcessEntry" type="alixdProcessEntryType"/>

<xsd:complexType name="alixdProcessEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceProcessEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceProcessEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdPrGroupName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
```

```
                        <xsd:attribute name="index" fixed="1"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="alixdPrProcessName">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                        <xsd:attribute name="index" fixed="2"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="alixdPrProcessNumber" type="xsd:string" minOccurs
            ="0"/>
        <xsd:element name="alixdPrPids" type="xsd:string" minOccurs="0"/>
        <xsd:element name="alixdPrMaxNumber" type="xsd:string" minOccurs
            ="0"/>
        <xsd:element name="alixdPrMinNumber" type="xsd:string" minOccurs
            ="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

<xsd:element name="alixdFilterEntry" type="alixdFilterEntryType"/>

<xsd:complexType name="alixdFilterEntryType">
  <xsd:sequence>
    <xsd:element ref="InstanceFilterEntry" minOccurs="1" maxOccurs="
      unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceFilterEntry">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="alixdFilterName">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="index" fixed="1"/>
            </xsd:extension>
```

```
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="alixdFilterType" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="lineMessage"><xsd:appInfo>
                        alixdFilterType:1</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="separatorWithRegularExpressions
                        "><xsd:appInfo>alixdFilterType:2</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="separatorWithAwkExpressions"><
                        xsd:appInfo>alixdFilterType:3</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="ExtractMessagesWithAwk"><xsd:
                        appInfo>alixdFilterType:4</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="ExtractMessagesWithProgram"><
                        xsd:appInfo>alixdFilterType:5</xsd:appInfo></xsd:
                        enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="alixdFilterStartMessage" type="xsd:string"
            minOccurs="0"/>
        <xsd:element name="alixdFilterEndMessage" type="xsd:string" minOccurs
            ="0"/>
        <xsd:element name="alixdFilterProgramName" type="xsd:string"
            minOccurs="0"/>
        <xsd:element name="alixdFilterStatus" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="ok"><xsd:appInfo>alixdFilterStatus
                        :1</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="inError"><xsd:appInfo>
                        alixdFilterStatus:2</xsd:appInfo></xsd:enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="alixdFilterErrorInformation" type="xsd:string"
            minOccurs="0"/>
    </xsd:all>
</xsd:complexType>
```

```
</xsd:element>

<xsd:element name="alixdFilterElemEntry" type="alixdFilterElemEntryType"/>

<xsd:complexType name="alixdFilterElemEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceFilterElemEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceFilterElemEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdFilterElemFilterName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alixdFilterElemIndex">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alixdFilterElemStatus" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="normal"><xsd:appInfo>
                            alixdFilterElemStatus:1</xsd:appInfo></xsd:enumeration
                            >
                        <xsd:enumeration value="overThreshold"><xsd:appInfo>
                            alixdFilterElemStatus:2</xsd:appInfo></xsd:enumeration
                            >
                    </xsd:restriction>
                </xsd:simpleType>
```

```
</xsd:element>
<xsd:element name="alixdFilterElemExpression" type="xsd:string"
    minOccurs="0"/>
<xsd:element name="alixdFilterElemCount" type="xsd:string" minOccurs
    ="0"/>
<xsd:element name="alixdFilterElemThreshold" type="xsd:string"
    minOccurs="0"/>
<xsd:element name="alixdFilterElemDuration" type="xsd:string"
    minOccurs="0"/>
<xsd:element name="alixdFilterElemLastMatchTime" type="xsd:string"
    minOccurs="0"/>
<xsd:element name="alixdFilterElemLastMessage" type="xsd:string"
    minOccurs="0"/>
<xsd:element name="alixdFilterElemTriggering" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="overMatch"><xsd:appInfo>
                alixdFilterElemTriggering:1</xsd:appInfo></xsd:
                enumeration>
            <xsd:enumeration value="overThreshold"><xsd:appInfo>
                alixdFilterElemTriggering:2</xsd:appInfo></xsd:
                enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="alixdFilterElemAction" type="xsd:string" minOccurs
    ="0"/>
<xsd:element name="alixdFilterElemSeverity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="indeterminate"><xsd:appInfo>
                alixdFilterElemSeverity:1</xsd:appInfo></xsd:
                enumeration>
            <xsd:enumeration value="critical"><xsd:appInfo>
                alixdFilterElemSeverity:2</xsd:appInfo></xsd:
                enumeration>
            <xsd:enumeration value="major"><xsd:appInfo>
                alixdFilterElemSeverity:3</xsd:appInfo></xsd:
                enumeration>
            <xsd:enumeration value="minor"><xsd:appInfo>
                alixdFilterElemSeverity:4</xsd:appInfo></xsd:
                enumeration>
            <xsd:enumeration value="warning"><xsd:appInfo>
```

```
                        alixdFilterElemSeverity:5</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="clear"><xsd:appInfo>
                        alixdFilterElemSeverity:6</xsd:appInfo></xsd:
                        enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

<xsd:element name="alixdActionEntry" type="alixdActionEntryType"/>

<xsd:complexType name="alixdActionEntryType">
  <xsd:sequence>
    <xsd:element ref="InstanceActionEntry" minOccurs="1" maxOccurs="
      unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceActionEntry">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="alixdActionName">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="index" fixed="1"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="alixdActionType" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="doNothing"><xsd:appInfo>
              alixdActionType:1</xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="sendTrap"><xsd:appInfo>
              alixdActionType:2</xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="executeProgram"><xsd:appInfo>
              alixdActionType:3</xsd:appInfo></xsd:enumeration>
          </xsd:restriction>
```

```
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="alixdActionDescription" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="alixdActionProgramName" type="xsd:string"
                minOccurs="0"/>
            <xsd:element name="alixdActionArguments" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="alixdActionPasswd" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="alixdActionDoesSendTrap" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="true"><xsd:appInfo>
                            alixdActionDoesSendTrap:1</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="false"><xsd:appInfo>
                            alixdActionDoesSendTrap:2</xsd:appInfo></xsd:
                            enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="alixdErrlogEntry" type="alixdErrlogEntryType"/>


<xsd:complexType name="alixdErrlogEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceErrlogEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="alErrlogClassType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="hard"><xsd:appInfo>alErrlogClass:1</xsd:appInfo
            ></xsd:enumeration>
        <xsd:enumeration value="soft"><xsd:appInfo>alErrlogClass:2</xsd:appInfo
            ></xsd:enumeration>
        <xsd:enumeration value="other"><xsd:appInfo>alErrlogClass:3</xsd:appInfo
```

```
            ></xsd:enumeration>
        <xsd:enumeration value="pend"><xsd:appInfo>alErrlogClass:4</xsd:appInfo
            ></xsd:enumeration>                                    </xsd:restriction
            >
</xsd:simpleType>

<xsd:element name="InstanceErrlogEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alErrlogClass">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="alErrlogClassType">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alErrlogIdentifier">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alErrlogLabel" type="xsd:string" minOccurs="0"/>
            <xsd:element name="alErrlogType" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="pend"><xsd:appInfo>alErrlogType
                            :1</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="perf"><xsd:appInfo>alErrlogType
                            :2</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="perm"><xsd:appInfo>alErrlogType
                            :3</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="temp"><xsd:appInfo>alErrlogType
                            :4</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="unkn"><xsd:appInfo>alErrlogType
                            :5</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="info"><xsd:appInfo>alErrlogType
                            :6</xsd:appInfo></xsd:enumeration>
```

```
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="alErrlogDescription" type="xsd:string" minOccurs
                ="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="alixdErrlogElemEntry" type="alixdErrlogElemEntryType"/>

<xsd:complexType name="alixdErrlogElemEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceErrlogElemEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceErrlogElemEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="alixdErrlogElemId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="alixdErrlogElemStatus" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="normal"><xsd:appInfo>
                            alixdErrlogElemStatus:1</xsd:appInfo></xsd:enumeration
                            >
                        <xsd:enumeration value="overThreshold"><xsd:appInfo>
                            alixdErrlogElemStatus:2</xsd:appInfo></xsd:enumeration
                            >
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="alixdErrlogElemCount" type="xsd:string" minOccurs
```

```
            ="0" />
        <xsd:element name="alixdErrlogElemThreshold" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="alixdErrlogElemDuration" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="alixdErrlogElemLastMatchTime" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="alixdErrlogElemLastMessage" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="alixdErrlogElemTriggering" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="everyTimes"><xsd:appInfo>
                        alixdErrlogElemTriggering:1</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="overThreshold"><xsd:appInfo>
                        alixdErrlogElemTriggering:2</xsd:appInfo></xsd:
                        enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="alixdErrlogElemAction" type="xsd:string" minOccurs
            ="0" />
    </xsd:all>
  </xsd:complexType>
</xsd:element>

</xsd:schema>
```

D.2 Agent d'extension

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:annotation>
    <xsd:documentation xml:lang="en">
        XML schema for Extension Open Agent
    </xsd:documentation>
</xsd:annotation>

<xsd:element name="OAXconfig" type="OAXconfigType">
    <xsd:annotation>
      <xsd:appinfo>generix|generixPackageEntry|generixTableEntry|
```

```
            generixAttributeEntry|generixDataEntry</xsd:appinfo>
        </xsd:annotation>
</xsd:element>

<xsd:complexType name="OAXconfigType">
    <xsd:sequence>
        <xsd:element ref="generix"/>
        <xsd:element ref="generixPackageEntry" minOccurs="0"/>
        <xsd:element ref="generixTableEntry" minOccurs="0"/>
        <xsd:element ref="generixAttributeEntry" minOccurs="0"/>
        <xsd:element ref="generixDataEntry" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="generix">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="generixTraceLevel" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="none"><xsd:appInfo>
                            generixTraceLevel:0</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="requests"><xsd:appInfo>
                            generixTraceLevel:1</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="functions"><xsd:appInfo>
                            generixTraceLevel:2</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="details"><xsd:appInfo>
                            generixTraceLevel:3</xsd:appInfo></xsd:enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="generixSaveConfigFlag" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="true"><xsd:appInfo>
                            generixSaveConfigFlag:1</xsd:appInfo></xsd:enumeration
                            >
                        <xsd:enumeration value="false"><xsd:appInfo>
                            generixSaveConfigFlag:2</xsd:appInfo></xsd:enumeration
                            >
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
```

```
            <xsd:element name="generixSaveConfigWhenExitFlag" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="true"><xsd:appInfo>
                            generixSaveConfigWhenExitFlag:1</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="false"><xsd:appInfo>
                            generixSaveConfigWhenExitFlag:2</xsd:appInfo></xsd:
                            enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="generixOldPasswd" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="generixNewPasswd" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="generixPath" type="xsd:string" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="generixPackageEntry" type="generixPackageEntryType"/>

<xsd:complexType name="generixPackageEntryType">
    <xsd:sequence>
        <xsd:element ref="InstancePackageEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstancePackageEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="generixPackageName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="generixPackageStatus" minOccurs="0">
```

```
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="ok"><xsd:appInfo>
                            generixPackageStatus:1</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="notAvailable"><xsd:appInfo>
                            generixPackageStatus:2</xsd:appInfo></xsd:enumeration>
                        <xsd:enumeration value="inError"><xsd:appInfo>
                            generixPackageStatus:3</xsd:appInfo></xsd:enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="generixPackageErrorInformation" type="xsd:string"
                minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="generixTableEntry" type="generixTableEntryType"/>

<xsd:complexType name="generixTableEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceTableEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceTableEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="generixTableId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="generixTableDescription" type="xsd:string"
                minOccurs="0"/>
            <xsd:element name="generixTableType" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
```

```
                <xsd:enumeration value="process"><xsd:appInfo>
                    generixTableType:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="file"><xsd:appInfo>
                    generixTableType:2</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="generixTableCommand" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="generixTableStderr" minOccurs="0">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="stderrOnNull"><xsd:appInfo>
                    generixTableStderr:1</xsd:appInfo></xsd:enumeration>
                <xsd:enumeration value="stderrOnStdout"><xsd:appInfo>
                    generixTableStderr:2</xsd:appInfo></xsd:enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="generixTableFile" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="generixTableValidityDuration" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableErrorInformation" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTablePasswd" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="generixTableStartLine" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="generixTableBeginListMeta" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableEndListMeta" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableEscapeMeta" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableCommentChar" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableSeparators" type="xsd:string"
        minOccurs="0"/>
    <xsd:element name="generixTableNullField" type="xsd:string" minOccurs
        ="0"/>
    <xsd:element name="generixTableFieldNumber" type="xsd:string"
        minOccurs="0"/>
```

```
            <xsd:element name="generixTableFilterValue" type="xsd:string"
                minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="generixAttributeEntry" type="generixAttributeEntryType"/>

<xsd:complexType name="generixAttributeEntryType">
    <xsd:sequence>
        <xsd:element ref="InstanceAttributeEntry" minOccurs="1" maxOccurs="
            unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceAttributeEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="generixAttrTableId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="generixAttrColumnNumber">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="generixAttrOid" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="generixAttrSyntax" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="Integer"><xsd:appInfo>
                            generixAttrSyntax:1</xsd:appInfo></xsd:enumeration>
```

```
                    <xsd:enumeration value="octetString"><xsd:appInfo>
                        generixAttrSyntax:2</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="ipAddress"><xsd:appInfo>
                        generixAttrSyntax:3</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="counter"><xsd:appInfo>
                        generixAttrSyntax:4</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="gauge"><xsd:appInfo>
                        generixAttrSyntax:5</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="timeTicks"><xsd:appInfo>
                        generixAttrSyntax:6</xsd:appInfo></xsd:enumeration>
                    <xsd:enumeration value="objectId"><xsd:appInfo>
                        generixAttrSyntax:7</xsd:appInfo></xsd:enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="generixAttrPasswd" type="xsd:string" minOccurs
            ="0"/>
        <xsd:element name="generixAttrName" type="xsd:string" minOccurs
            ="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>

<xsd:element name="generixDataEntry" type="generixDataEntryType"/>

<xsd:complexType name="generixDataEntryType">
  <xsd:sequence>
    <xsd:element ref="InstanceDataEntry" minOccurs="1" maxOccurs="
        unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="InstanceDataEntry">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="generixDataTableId">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="index" fixed="1"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
```

```
            </xsd:element>
            <xsd:element name="generixDataName">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="generixDataLineCount" type="xsd:string" minOccurs
                ="0"/>
            <xsd:element name="generixData1" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData2" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData3" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData4" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData5" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData6" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData7" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData8" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData9" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData10" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData11" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData12" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData13" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData14" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData15" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData16" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData17" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData18" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData19" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData20" type="xsd:string" minOccurs="0"/>
            <xsd:element name="generixData21" type="xsd:string" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

</xsd:schema>
```

D.3 Agent COACH

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:annotation>
```

```
    <xsd:documentation xml:lang="en">
        XML schema for COACH Open Agent
    </xsd:documentation>
</xsd:annotation>

<xsd:element name="COACHconfig" type="COACHconfigType"/>

<xsd:complexType name="COACHconfigType">
    <xsd:sequence>
        <xsd:element ref="confmod.ini" minOccurs="0"/>
        <xsd:element ref="aic.inst" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="confmod.ini">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="filter" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="indicator" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:simpleType name="cfgIndicatorComparaisonType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="less"><xsd:appInfo>cfgIndicatorComparaison
            :&#60;</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="greater"><xsd:appInfo>cfgIndicatorComparaison
            :&#62;</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="equal"><xsd:appInfo>cfgIndicatorComparaison:=</
            xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="different"><xsd:appInfo>cfgIndicatorComparaison
            :!=</xsd:appInfo></xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="cfgIndicatorLogOnType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="on"><xsd:appInfo>cfgIndicatorLogOn:LOG </xsd:
            appInfo></xsd:enumeration>
        <xsd:enumeration value="off"><xsd:appInfo>cfgIndicatorLogOn:NLOG </xsd
            :appInfo></xsd:enumeration>
    </xsd:restriction>
```

```
</xsd:simpleType>

<xsd:simpleType name="cfgIndicatorTrapType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="aicTrapCritical"><xsd:appInfo>cfgIndicatorTrap
            :1</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicTrapMajor"><xsd:appInfo>cfgIndicatorTrap:2</
            xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicTrapMinor"><xsd:appInfo>cfgIndicatorTrap:3</
            xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicTrapWarning"><xsd:appInfo>cfgIndicatorTrap
            :4</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicTrapClear"><xsd:appInfo>cfgIndicatorTrap:5</
            xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicAutoTrapCritical"><xsd:appInfo>
            cfgIndicatorTrap:11</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicAutoTrapMajor"><xsd:appInfo>cfgIndicatorTrap
            :12</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicAutoTrapMinor"><xsd:appInfo>cfgIndicatorTrap
            :13</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicAutoTrapWarning"><xsd:appInfo>
            cfgIndicatorTrap:14</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicAutoTrapClear"><xsd:appInfo>cfgIndicatorTrap
            :15</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicMibTrapCritical"><xsd:appInfo>cfgIndicatorTrap
            :21</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicMibTrapMajor"><xsd:appInfo>cfgIndicatorTrap
            :22</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicMibTrapMinor"><xsd:appInfo>cfgIndicatorTrap
            :23</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicMibTrapWarning"><xsd:appInfo>
            cfgIndicatorTrap:24</xsd:appInfo></xsd:enumeration>
        <xsd:enumeration value="aicMibTrapClear"><xsd:appInfo>cfgIndicatorTrap
            :25</xsd:appInfo></xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:element name="indicator">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgIndicatorId">
                <xsd:complexType>
                    <xsd:simpleContent>
```

```
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="1"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorLabel">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="2"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorDomain">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="3"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorEquation">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="4"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorPeriodPolling">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="5"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorThreshold">
```

```
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="6"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorCptMax">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="7"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorPeriodValid">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="index" fixed="8"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorComparaison">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="cfgIndicatorComparaisonType">
                    <xsd:attribute name="index" fixed="9"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgIndicatorLogOn">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="cfgIndicatorLogOnType">
                    <xsd:attribute name="index" fixed="10"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
```

```
            </xsd:element>
            <xsd:element name="cfgIndicatorTrap">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="cfgIndicatorTrapType">
                            <xsd:attribute name="index" fixed="11"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="filter">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgFilterId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="1"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="cfgFilterLabel">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="2"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="cfgFilterDomain">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="index" fixed="3"/>
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
```

```
</xsd:element>
<xsd:element name="cfgFilterEnterprise">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="index" fixed="4"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="cfgFilterGeneric">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="index" fixed="5"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="cfgFilterSpecific">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="index" fixed="6"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="cfgFilterCptMax">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="index" fixed="7"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="cfgFilterPeriodValid">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="index" fixed="8"/>
            </xsd:extension>
```

```
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="aic.inst">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="cfgDiscovery"/>
            <xsd:element ref="cfgSystem" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="cfgDomain" minOccurs="1" maxOccurs="unbounded
                "/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="cfgDiscovery">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="cfgDiscoverIpEntry" minOccurs="0" maxOccurs="
                unbounded"/>
            <xsd:element ref="cfgDiscoverIpSetEntry" minOccurs="0" maxOccurs="
                unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="cfgDiscoverId" fixed="0"/>
        <xsd:attribute name="cfgDiscoverPeriod" type="xsd:string"/>
        <xsd:attribute name="cfgDiscoverSubnetMask" type="xsd:string"/>
        <xsd:attribute name="cfgDiscoverDomainAtNextTime">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="yes"><xsd:appInfo>
                        cfgDiscoverDomainAtNextTime:1</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="no"><xsd:appInfo>
                        cfgDiscoverDomainAtNextTime:2</xsd:appInfo></xsd:
                        enumeration>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="cfgDiscoverTimeOut" type="xsd:string"/>
        <xsd:attribute name="cfgDiscoverRedoPeriod" type="xsd:string"/>
```

```
    </xsd:complexType>
</xsd:element>

<xsd:element name="cfgDiscoverIpEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgDiscoverIpIndex" type="xsd:string"/>
            <xsd:element name="cfgDiscoverIp" type="xsd:string"/>
            <xsd:element name="cfgDiscoverIpDomain" type="xsd:string" minOccurs
                ="0"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="cfgDiscoverIpSetEntry">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgDiscoverIpSetIndex" type="xsd:string"/>
            <xsd:element name="cfgDiscoverIpLower" type="xsd:string"/>
            <xsd:element name="cfgDiscoverIpUpper" type="xsd:string"/>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="cfgDomain">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="cfgDomainId" type="xsd:string"/>
            <xsd:element name="cfgDomainLabel" type="xsd:string"/>
            <xsd:element name="cfgAtt1">
                <xsd:complexType>
                    <xsd:attribute name="cfgAtt1" type="xsd:string"/>
                    <xsd:attribute name="cfgAtt1Value" type="xsd:string"/>
                    <xsd:attribute name="cfgAtt1Comparaison">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="exist"><xsd:appInfo>
                                    cfgAtt1Comparaison:0</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="equal"><xsd:appInfo>
                                    cfgAtt1Comparaison:1</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="greater"><xsd:appInfo>
```

```
                    cfgAtt1Comparaison:2</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="different"><xsd:appInfo>
                    cfgAtt1Comparaison:3</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="lessequal"><xsd:appInfo>
                    cfgAtt1Comparaison:4</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="greaterequal"><xsd:appInfo
                    >cfgAtt1Comparaison:5</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="less"><xsd:appInfo>
                    cfgAtt1Comparaison:6</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="include"><xsd:appInfo>
                    cfgAtt1Comparaison:7</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="notinclude"><xsd:appInfo>
                    cfgAtt1Comparaison:8</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="node"><xsd:appInfo>
                    cfgAtt1Comparaison:9</xsd:appInfo></xsd:
                    enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="cfgAtt2" minOccurs="0">
    <xsd:complexType>
        <xsd:attribute name="cfgAtt2" type="xsd:string"/>
        <xsd:attribute name="cfgAtt2Value" type="xsd:string"/>
        <xsd:attribute name="cfgAtt2Comparaison">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="exist"><xsd:appInfo>
                        cfgAtt2Comparaison:0</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="equal"><xsd:appInfo>
                        cfgAtt2Comparaison:1</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="greater"><xsd:appInfo>
                        cfgAtt2Comparaison:2</xsd:appInfo></xsd:
```

```
                    enumeration>
                <xsd:enumeration value="different"><xsd:appInfo>
                    cfgAtt2Comparaison:3</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="lessequal"><xsd:appInfo>
                    cfgAtt2Comparaison:4</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="greaterequal"><xsd:appInfo
                    >cfgAtt2Comparaison:5</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="less"><xsd:appInfo>
                    cfgAtt2Comparaison:6</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="include"><xsd:appInfo>
                    cfgAtt2Comparaison:7</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="notinclude"><xsd:appInfo>
                    cfgAtt2Comparaison:8</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="node"><xsd:appInfo>
                    cfgAtt2Comparaison:9</xsd:appInfo></xsd:
                    enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="cfgAtt3" minOccurs="0">
    <xsd:complexType>
        <xsd:attribute name="cfgAtt3" type="xsd:string"/>
        <xsd:attribute name="cfgAtt3Value" type="xsd:string"/>
        <xsd:attribute name="cfgAtt3Comparaison">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="exist"><xsd:appInfo>
                        cfgAtt3Comparaison:0</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="equal"><xsd:appInfo>
                        cfgAtt3Comparaison:1</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="greater"><xsd:appInfo>
                        cfgAtt3Comparaison:2</xsd:appInfo></xsd:
                        enumeration>
```

```
                        <xsd:enumeration value="different"><xsd:appInfo>
                            cfgAtt3Comparaison:3</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="lessequal"><xsd:appInfo>
                            cfgAtt3Comparaison:4</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="greaterequal"><xsd:appInfo
                            >cfgAtt3Comparaison:5</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="less"><xsd:appInfo>
                            cfgAtt3Comparaison:6</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="include"><xsd:appInfo>
                            cfgAtt3Comparaison:7</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="notinclude"><xsd:appInfo>
                            cfgAtt3Comparaison:8</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="node"><xsd:appInfo>
                            cfgAtt3Comparaison:9</xsd:appInfo></xsd:
                            enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:attribute>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="cfgAtt4" minOccurs="0">
        <xsd:complexType>
            <xsd:attribute name="cfgAtt4" type="xsd:string"/>
            <xsd:attribute name="cfgAtt4Value" type="xsd:string"/>
            <xsd:attribute name="cfgAtt4Comparaison">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="exist"><xsd:appInfo>
                            cfgAtt4Comparaison:0</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="equal"><xsd:appInfo>
                            cfgAtt4Comparaison:1</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="greater"><xsd:appInfo>
                            cfgAtt4Comparaison:2</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="different"><xsd:appInfo>
```

```
                    cfgAtt4Comparaison:3</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="lessequal"><xsd:appInfo>
                    cfgAtt4Comparaison:4</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="greaterequal"><xsd:appInfo
                    >cfgAtt4Comparaison:5</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="less"><xsd:appInfo>
                    cfgAtt4Comparaison:6</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="include"><xsd:appInfo>
                    cfgAtt4Comparaison:7</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="notinclude"><xsd:appInfo>
                    cfgAtt4Comparaison:8</xsd:appInfo></xsd:
                    enumeration>
                <xsd:enumeration value="node"><xsd:appInfo>
                    cfgAtt4Comparaison:9</xsd:appInfo></xsd:
                    enumeration>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="cfgAtt5" minOccurs="0">
    <xsd:complexType>
        <xsd:attribute name="cfgAtt5" type="xsd:string"/>
        <xsd:attribute name="cfgAtt5Value" type="xsd:string"/>
        <xsd:attribute name="cfgAtt5Comparaison">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="exist"><xsd:appInfo>
                        cfgAtt5Comparaison:0</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="equal"><xsd:appInfo>
                        cfgAtt5Comparaison:1</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="greater"><xsd:appInfo>
                        cfgAtt5Comparaison:2</xsd:appInfo></xsd:
                        enumeration>
                    <xsd:enumeration value="different"><xsd:appInfo>
                        cfgAtt5Comparaison:3</xsd:appInfo></xsd:
```

```
                                    enumeration>
                                <xsd:enumeration value="lessequal"><xsd:appInfo>
                                    cfgAtt5Comparaison:4</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="greaterequal"><xsd:appInfo
                                    >cfgAtt5Comparaison:5</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="less"><xsd:appInfo>
                                    cfgAtt5Comparaison:6</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="include"><xsd:appInfo>
                                    cfgAtt5Comparaison:7</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="notinclude"><xsd:appInfo>
                                    cfgAtt5Comparaison:8</xsd:appInfo></xsd:
                                    enumeration>
                                <xsd:enumeration value="node"><xsd:appInfo>
                                    cfgAtt5Comparaison:9</xsd:appInfo></xsd:
                                    enumeration>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:attribute>
                </xsd:complexType>
            </xsd:element>
        </xsd:all>
    </xsd:complexType>
</xsd:element>

<xsd:element name="cfgSystem">
    <xsd:complexType>
        <xsd:all>
            <xsd:element name="systemSaveConfiguration" minOccurs="0">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="on"><xsd:appInfo>
                            systemSaveConfiguration:1</xsd:appInfo></xsd:
                            enumeration>
                        <xsd:enumeration value="off"><xsd:appInfo>
                            systemSaveConfiguration:2</xsd:appInfo></xsd:
                            enumeration>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
```

```
<xsd:element name="detachTrapLogs" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"><xsd:appInfo>detachTrapLogs
                :1</xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="off"><xsd:appInfo>detachTrapLogs
                :2</xsd:appInfo></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="detachIndicatorLogs" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"><xsd:appInfo>
                detachIndicatorLogs:1</xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="off"><xsd:appInfo>
                detachIndicatorLogs:2</xsd:appInfo></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="logOnTraps" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"><xsd:appInfo>logOnTraps:1</
                xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="off"><xsd:appInfo>logOnTraps:2</
                xsd:appInfo></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="logOnIndicators" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="on"><xsd:appInfo>logOnIndicators
                :1</xsd:appInfo></xsd:enumeration>
            <xsd:enumeration value="off"><xsd:appInfo>logOnIndicators
                :2</xsd:appInfo></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="indicatorMaxTimePolling" minOccurs="0"/>
<xsd:element name="indicatorPollingUnitPeriod" minOccurs="0"/>
<xsd:element name="systemSnmpCommunity" minOccurs="0"/>
```

```
        </xsd:all>
        <xsd:attribute name="cfgSystemId" fixed="0" />
    </xsd:complexType>
</xsd:element>

</xsd:schema>
```

E Exemples de parseur Java utilisant les schémas XML du W3C

E.1 Parseur SAX

```
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;

import java.util.*;
import java.io.*;

public class saxParserClass extends DefaultHandler{

    public void startDocument() throws SAXException
    {}

    public void startElement(String namespaceURI, String localName,String qName,
        Attributes atts) throws SAXException
    {}

    public void endElement(String namespaceURI, String localName,String qName)
        throws SAXException
    {}

    public void endDocument() throws SAXException
    {}

    public static void main(String argv[])
        {
        SAXParserFactory spf = SAXParserFactory.newInstance();
        spf.setNamespaceAware(true);
        spf.setValidating(true);
        try {
            SAXParser saxParser = spf.newSAXParser();
            saxParser.setProperty("http://java.sun.com/xml/jaxp/properties/
                schemaLanguage","http://www.w3.org/2001/XMLSchema");
            XMLReader xmlReader = saxParser.getXMLReader();
            xmlReader.setContentHandler(new saxParserClass());
            xmlReader.setErrorHandler(new MyErrorHandler(System.err));
            xmlReader.parse("/home/raptor/Memoire/data/schematron/OACconfig.sch
                ");
            System.out.println("Fin du traitement\n");
```

```
            }
        catch(SAXException e){System.out.println(e);}
        catch(IOException e){System.out.println(e);}
        catch(ParserConfigurationException e){System.out.println(e);}
        }

    private static class MyErrorHandler implements ErrorHandler {

        private PrintStream out;

        MyErrorHandler(PrintStream out) {
            this.out = out;
        }

        private String getParseExceptionInfo(SAXParseException spe) {
            String systemId = spe.getSystemId();
            if (systemId == null) {
                systemId = "null";
            }
            String info = "URI=" + systemId +
                " Line=" + spe.getLineNumber() +
                ": " + spe.getMessage();
            return info;
        }

        public void warning(SAXParseException spe) throws SAXException {
            out.println("Warning: " + getParseExceptionInfo(spe));
        }

        public void error(SAXParseException spe) throws SAXException {
            String message = "Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }

        public void fatalError(SAXParseException spe) throws SAXException {
            String message = "Fatal Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }
    }

}
```

E.2 Parseur DOM transformant des feuilles XML en configuration d'agent

```
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import org.w3c.dom.*;

import java.util.*;
import java.io.*;

public class ConfigurationParser extends DefaultHandler{

    private static Hashtable Htable = new Hashtable();
    private static PrintWriter out;

    private static void init_Hashtable(String schemaFile)
        {
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        dbf.setNamespaceAware(true);
        try {
            DocumentBuilder db = dbf.newDocumentBuilder();
            db.setErrorHandler(new MyErrorHandler(System.err));
            Document doc = db.parse(schemaFile);
            NodeList n = doc.getElementsByTagName("xsd:appInfo");
            for(int i=0;i<n.getLength();i++)
                {
                if(n.item(i).getParentNode().getNodeName().equals("xsd:enumeration
                    "))
                    {
                    StringTokenizer st = new StringTokenizer(n.item(i).getFirstChild().
                        getNodeValue(),":");
                    Htable.put(st.nextToken()+n.item(i).getParentNode().getAttributes
                        ().getNamedItem("value").getNodeValue(),st.nextToken());
                    }
                }
            }
        catch(SAXException e){System.out.println(e);}
        catch(IOException e){System.out.println(e);}
        catch(ParserConfigurationException e){System.out.println(e);}
        }

    private static Document init_Parsing(String file)
```

```
    {
    Document doc=null;
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    dbf.setNamespaceAware(true);
    dbf.setValidating(true);
    dbf.setAttribute("http://java.sun.com/xml/jaxp/properties/schemaLanguage
        ","http://www.w3.org/2001/XMLSchema");
    try {
        DocumentBuilder db = dbf.newDocumentBuilder();
        db.setErrorHandler(new MyErrorHandler(System.err));
        doc = db.parse(file);
        }
    catch(SAXException e){System.out.println(e);}
    catch(IOException e){System.out.println(e);}
    catch(ParserConfigurationException e){System.out.println(e);}
    return doc;
    }

private static void traitement_attribut(NodeList element,String ecart)
    {
    for(int i=1;i<element.getLength();i++)
        {
        if (!element.item(i).getNodeName().equals("#text"))
            {
            String tmp = element.item(i).getFirstChild().getNodeValue();
            if(Htable.containsKey(element.item(i).getNodeName()+tmp)){tmp=(
                String)Htable.get(element.item(i).getNodeName()+tmp);}
            out.println(ecart+element.item(i).getNodeName()+" = "+tmp+" ;");
            }
        }
    }

private static String recherche_cle(NodeList element,Node parent)
    {
    Hashtable t = new Hashtable();
    String cle="";
    for(int i=1;i<element.getLength();i++)
        {
        if(element.item(i).hasAttributes())
         {
         String tmp = element.item(i).getFirstChild().getNodeValue();
         if(Htable.containsKey(element.item(i).getNodeName()+tmp)){tmp=(
             String)Htable.get(element.item(i).getNodeName()+tmp);}
```

```
            t.put(element.item(i).getAttributes().getNamedItem("index").
                getNodeValue(),tmp);parent.removeChild(element.item(i));}
        }
    for(int j=1;j<=t.size();j++)
        {cle=cle+"."+t.get(Integer.toString(j));}

    return cle.substring(1);
    }

private static void Parsing(Document doc,String prefixe)
    {
    NodeList elt1 = doc.getDocumentElement().getChildNodes();
    for(int i=1;i<elt1.getLength();i++)
        {
        if(elt1.item(i).getNodeName().equals(prefixe))
            {
            out.println("CLASS "+elt1.item(i).getNodeName()+" {");
            NodeList elt2 = elt1.item(i).getChildNodes();
            traitement_attribut(elt2,"\t");
            out.println("\t}\n");
            }
        else if(!elt1.item(i).getNodeName().equals("#text"))
            {
            out.println("CLASS "+elt1.item(i).getNodeName()+" {");
            NodeList elt2 = elt1.item(i).getChildNodes();
            for(int j=1;j<elt2.getLength();j++)
                {
                if(!elt2.item(j).getNodeName().equals("#text"))
                    {
                    NodeList elt3 = elt2.item(j).getChildNodes();
                    String cle = recherche_cle(elt3,elt2.item(j));
                    out.println("\tINSTANCE \""+cle+"\" {");
                    traitement_attribut(elt3,"\t\t");
                    out.println("\t\t}\n");
                    }
                }
            out.println("\t}\n");
            }
        }
    }

private static void confmod(NodeList elt,String entete)
    {
```

```
for(int i=0;i<elt.getLength();i++)
    {
    NodeList elt_childs = elt.item(i).getChildNodes();
    Hashtable index = new Hashtable();
    for(int j=0;j<elt_childs.getLength();j++)
        {
        if (! elt_childs .item(j).getNodeName().equals("#text"))
            {
            String cle=elt_childs.item(j).getNodeName()+elt_childs.item(j).
                getFirstChild().getNodeValue();
            if(Htable.containsKey(cle))
                {
                index.put((String) elt_childs .item(j).getAttributes().
                    getNamedItem("index").getNodeValue(),Htable.get(cle));
                }
            else
                {
                index.put((String) elt_childs .item(j).getAttributes().
                    getNamedItem("index").getNodeValue(),(String)elt_childs.
                    item(j).getFirstChild().getNodeValue());
                }
            }
        }
    String chaine=entete;
    for(int j=1;j<=index.size();j++)
        {
        chaine=chaine+" "+((String)index.get(Integer.toString(j))).trim();
        }
    out.println(chaine);
    }
}

private static void init_aic (NodeList cfgDiscovery, NodeList cfgSystem)
    {
    NamedNodeMap attr = cfgDiscovery.item(0).getAttributes();
    String ID=attr.getNamedItem("cfgDiscoverId").getFirstChild().getNodeValue();
    attr.removeNamedItem("cfgDiscoverId");
    for(int i=0;i<attr.getLength();i++)
        {
        String cle=attr.item(i).getNodeName()+attr.item(i).getFirstChild().
            getNodeValue();
        if(Htable.containsKey(cle))
            {out.println(attr.item(i).getNodeName()+"."+ID+" "+Htable.get(cle)
```

```
            );}
        else
            {out.println(attr.item(i).getNodeName()+"."+ID+" "+attr.item(i).
                getFirstChild().getNodeValue());}
        }
    NodeList childDiscovery = cfgDiscovery.item(0).getChildNodes();
    for(int i=0;i<childDiscovery.getLength();i++)
        {
        if (!childDiscovery.item(i).getNodeName().equals("#text"))
            {
            String index="";
            NodeList child = childDiscovery.item(i).getChildNodes();
            for(int j=0;j<child.getLength();j++)
                {
                if(child.item(j).getNodeName().endsWith("Index"))
                    {
                    index=child.item(j).getFirstChild().getNodeValue();
                    childDiscovery.item(i).removeChild(child.item(j));
                    }
                }
            for(int j=0;j<child.getLength();j++)
                {
                if (!child.item(j).getNodeName().equals("#text"))
                    {
                    String cle=child.item(j).getNodeName()+child.item(j).
                        getFirstChild().getNodeValue();
                    if(Htable.containsKey(cle))
                        {out.println(child.item(j).getNodeName()+"."+index
                            +" "+Htable.get(cle));}
                    else
                        {out.println(child.item(j).getNodeName()+"."+index
                            +" "+child.item(j).getFirstChild().getNodeValue());}
                    }
                }
            }
        }
    if(cfgSystem!=null)
        {
        String index=cfgSystem.item(0).getAttributes().getNamedItem("
            cfgSystemId").getFirstChild().getNodeValue();
        NodeList childSystem = cfgSystem.item(0).getChildNodes();
        for(int j=0;j<childSystem.getLength();j++)
            {
```

```
            if (!childSystem.item(j).getNodeName().equals("#text"))
                {
                String cle=childSystem.item(j).getNodeName()+childSystem.item(j).
                    getFirstChild().getNodeValue();
                if (Htable.containsKey(cle))
                   {out.println(childSystem.item(j).getNodeName()+"."+index
                       +" "+Htable.get(cle));}
                else
                   {out.println(childSystem.item(j).getNodeName()+"."+index
                       +" "+childSystem.item(j).getFirstChild().getNodeValue());}
                }
            }
        }
    }

private static void gestionDomain_aic(NodeList cfgDomain)
    {
    for(int i=0;i<cfgDomain.getLength();i++)
        {
        String index="";
        NodeList childcfgDomain = cfgDomain.item(i).getChildNodes();
        for(int j=0;j<childcfgDomain.getLength();j++)
            {
            if(childcfgDomain.item(j).getNodeName().endsWith("Id"))
                {
                index=childcfgDomain.item(j).getFirstChild().getNodeValue();
                cfgDomain.item(i).removeChild(childcfgDomain.item(j));
                }
            }
        for(int j=0;j<childcfgDomain.getLength();j++)
            {
            if (!childcfgDomain.item(j).getNodeName().equals("#text") && !
                childcfgDomain.item(j).getNodeName().startsWith("cfgAtt"))
                {
                String cle=childcfgDomain.item(j).getNodeName()+childcfgDomain
                    .item(j).getFirstChild().getNodeValue();
                if(Htable.containsKey(cle))
                    {out.println(childcfgDomain.item(j).getNodeName()+"."+index
                        +" "+Htable.get(cle));}
                else
                    {out.println(childcfgDomain.item(j).getNodeName()+"."+index
                        +" "+childcfgDomain.item(j).getFirstChild().getNodeValue
                        ());}
```

```
                }
            else  if(childcfgDomain.item(j).getNodeName().startsWith("cfgAtt"))
                {
                NamedNodeMap attr = childcfgDomain.item(j).getAttributes();
                for(int  k=0;k<attr.getLength();k++)
                    {
                    String  cle=attr.item(k).getNodeName()+attr.item(k).
                        getFirstChild().getNodeValue();
                    if(Htable.containsKey(cle))
                        {out.println(attr.item(k).getNodeName()+"."+index
                            +" "+Htable.get(cle));}
                    else
                        {out.println(attr.item(k).getNodeName()+"."+index
                            +" "+attr.item(k).getFirstChild().getNodeValue());}
                    }
                }
            }
        }

public  static  void main(String[] argv)
    {
    try {
        Document doc = init_Parsing(argv[0]);
        if(doc==null){System.out.println(argv[0]+" seems to be not well-formed,
            loading failed...");}
        else {
            String agentType=doc.getFirstChild().getNodeName();
            String schemaFile=doc.getFirstChild().getAttributes().getNamedItem
                ("xsi:noNamespaceSchemaLocation").getNodeValue();
            init_Hashtable(schemaFile);
            if(agentType.equals("OACconfig"))
                {
                try {
                    System.out.println("Configuration of Control open agent ;-)")
                        ;
                    out = new PrintWriter(new FileWriter("alixd.conf.dyn"));
                    Parsing(doc,"alixd");
                    out.close();
                    System.out.println("File \"alixd.conf.dyn\" generated ;-)");
                    }
                catch(IOException e){System.out.println(e);}
                }
```

```
            else  if(agentType.equals("OAXconfig"))
                {
                try {
                    System.out.println("Configuration of Extension open agent ;-)
                        ");
                    out = new PrintWriter(new FileWriter("generix.conf.dyn"));
                    Parsing(doc,"generix");
                    out.close();
                    System.out.println("File \"generix.conf.dyn\" generated ;-)")
                        ;
                    }
                catch(IOException e){System.out.println(e);}
                }
            else  if(agentType.equals("COACHconfig"))
                {
                try {
                    System.out.println("Configuration of COACH open agent ;-)
                        ");
                    out = new PrintWriter(new FileWriter("confmod.ini"));
                    confmod(doc.getElementsByTagName("filter"),"FIL");
                    confmod(doc.getElementsByTagName("indicator"),"IND");
                    out.close();
                    System.out.println("File \"confmod.ini\" generated ;-)");
                    out = new PrintWriter(new FileWriter("aic.inst"));
                     init_aic (doc.getElementsByTagName("cfgDiscovery"),doc.
                        getElementsByTagName("cfgSystem"));
                    gestionDomain_aic(doc.getElementsByTagName("cfgDomain"))
                        ;
                    out.close();
                    System.out.println("File \"aic.inst\" generated ;-)");
                    }
                catch(IOException e){System.out.println(e);}
                }
            else
                {
                System.out.println("Unknown agent type...");
                }
            }
        }
    catch(ArrayIndexOutOfBoundsException e){System.out.println("File name
        missing :-(");}
    }
```

```
    private static class MyErrorHandler implements ErrorHandler {

        private PrintStream out;

        MyErrorHandler(PrintStream out) {
            this.out = out;
        }

        private String getParseExceptionInfo(SAXParseException spe) {
            String systemId = spe.getSystemId();
            if (systemId == null) {
                systemId = "null";
            }
            String info = "URI=" + systemId +
                " Line=" + spe.getLineNumber() +
                ": " + spe.getMessage();
            return info;
        }

        public void warning(SAXParseException spe) throws SAXException {
            out.println("Warning: " + getParseExceptionInfo(spe));
        }

        public void error(SAXParseException spe) throws SAXException {
            String message = "Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }

        public void fatalError(SAXParseException spe) throws SAXException {
            String message = "Fatal Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }
    }

}
```

F LDAP

F.1 Schéma LDAP de l'agent de contrôle

attributetype (1.3.6.1.4.1.107.118.503.1 NAME 'OACname'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{200})

attributetype (1.3.6.1.4.1.107.118.503.2 NAME 'configName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.3 NAME 'version' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.4 NAME 'alixdSourcesEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.5 NAME 'alixdFileEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.6 NAME 'alixdProcessEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.7 NAME 'alixdFilterEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.8 NAME 'alixdFilterElemEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.9 NAME 'alixdActionEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.10 NAME 'alixdErrlogEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.503.11 NAME 'alixdErrlogElemEntryID' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.4 NAME 'alixdTraceLevel' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.5 NAME 'alixdSaveConfigFlag' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.6 NAME 'alixdSaveConfigWhenExitFlag' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.7 NAME 'alixdOldPasswd' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.8 NAME 'alixdNewPasswd' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.9 NAME 'alixdPollingStatus' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.10 NAME 'alixdMaxTrapsPerMinut' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.13 NAME 'alixdPath' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.14 NAME 'alixActionLogType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.1 NAME 'alixdSourcesName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.2 NAME 'alixdSourcesStatus' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.3 NAME 'alixdSourcesErrorInformation' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.4 NAME 'alixdSourcesFrequency' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.5 NAME 'alixdSourcesHighWaterMark' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.6 NAME 'alixdSourcesLowWaterMark' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.7 NAME 'alixdSourcesFile' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.8 NAME 'alixdSourcesType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.9 NAME 'alixdSourcesSpecificName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.10 NAME 'alixdSourcesFilterName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.11 NAME 'alixdSourcesDefaultAction' SUP

OACname)

attributetype (1.3.6.1.4.1.107.118.502.1.12 NAME 'alixdSourcesPasswd' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.1 NAME 'alixdFileGroupName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.2 NAME 'alixdFileIndex' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.3 NAME 'alixdFileName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.4 NAME 'alixdFileDoesExist' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.5 NAME 'alixdFileSize' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.6 NAME 'alixdFileMaxSize' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.101.1.7 NAME 'alixdFileNormalSize' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.1 NAME 'alixdPrGroupName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.2 NAME 'alixdPrProcessName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.3 NAME 'alixdPrProcessNumber' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.4 NAME 'alixdPrPids' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.5 NAME 'alixdPrMaxNumber' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.100.1.6 NAME 'alixdPrMinNumber' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.1 NAME 'alixdFilterName' SUP

OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.2 NAME 'alixdFilterType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.3 NAME 'alixdFilterStartMessage' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.4 NAME 'alixdFilterEndMessage' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.5 NAME 'alixdFilterProgramName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.6 NAME 'alixdFilterStatus' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.110.1.7 NAME 'alixdFilterErrorInformation' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.1 NAME 'alixdFilterElemFilterName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.2 NAME 'alixdFilterElemIndex' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.3 NAME 'alixdFilterElemStatus' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.4 NAME 'alixdFilterElemType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.5 NAME 'alixdFilterElemExpression' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.6 NAME 'alixdFilterElemCount' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.7 NAME 'alixdFilterElemThreshold' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.8 NAME 'alixdFilterElemDuration' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.9 NAME 'alixdFilterElemLastMatchTime ' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.10 NAME 'alixdFilterElemLastMessage' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.11 NAME 'alixdFilterElemTriggering' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.12 NAME 'alixdFilterElemAction' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.111.1.13 NAME 'alixdFilterElemSeverity' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.1 NAME 'alixdActionName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.2 NAME 'alixdActionType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.3 NAME 'alixdActionDescription' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.4 NAME 'alixdActionProgramName' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.5 NAME 'alixdActionArguments' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.6 NAME 'alixdActionPasswd' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.112.1.7 NAME 'alixdActionDoesSendTrap' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.1.1.1 NAME 'alErrlogClass' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.1.1.2 NAME 'alErrlogIdentifier' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.1.1.3 NAME 'alErrlogLabel' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.1.1.4 NAME 'alErrlogType' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.1.1.5 NAME 'alErrlogDescription' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.1 NAME 'alixdErrlogElemId' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.2 NAME 'alixdErrlogElemStatus' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.3 NAME 'alixdErrlogElemCount' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.4 NAME 'alixdErrlogElemThreshold' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.5 NAME 'alixdErrlogElemDuration' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.6 NAME ' alixdErrlogElemLastMatchTime' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.7 NAME 'alixdErrlogElemLastMessage ' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.8 NAME 'alixdErrlogElemTriggering' SUP OACname)

attributetype (1.3.6.1.4.1.107.118.501.120.2.1.9 NAME 'alixdErrlogElemAction' SUP OACname)

objectclass (1.3.6.1.4.1.107.118.503 NAME 'OACconfig' SUP top STRUCTURAL
MUST (configName))

objectclass (1.3.6.1.4.1.107.118.501 NAME 'alixd' SUP top STRUCTURAL
MUST (version)
MAY (alixdTraceLevel $ alixdSaveConfigFlag $ alixdSaveConfigWhenExitFlag $ alixdOldPasswd $ alixdNewPasswd $ alixdPollingStatus $

```
        alixdMaxTrapsPerMinut $ alixdPath $ alixActionLogType ))

objectclass   ( 1.3.6.1.4.1.107.118.502.1   NAME 'alixdSourcesEntry' SUP top
    STRUCTURAL
        MUST ( alixdSourcesEntryID $ alixdSourcesName )
        MAY ( alixdSourcesStatus $ alixdSourcesErrorInformation $
            alixdSourcesFrequency $ alixdSourcesHighWaterMark $
            alixdSourcesLowWaterMark $ alixdSourcesFile $ alixdSourcesType $
            alixdSourcesSpecificName $ alixdSourcesFilterName $
            alixdSourcesDefaultAction $ alixdSourcesPasswd ))

objectclass   ( 1.3.6.1.4.1.107.118.501.101.1   NAME 'alixdFileEntry' SUP top
    STRUCTURAL
        MUST ( alixdFileEntryID $ alixdFileGroupName $ alixdFileIndex )
        MAY ( alixdFileName $ alixdFileDoesExist $ alixdFileSize $ alixdFileMaxSize $
            alixdFileNormalSize ))

objectclass   ( 1.3.6.1.4.1.107.118.501.100.1   NAME 'alixdProcessEntry' SUP top
    STRUCTURAL
        MUST ( alixdProcessEntryID $ alixdPrGroupName $ alixdPrProcessName )
        MAY ( alixdPrProcessNumber $ alixdPrPids $ alixdPrMaxNumber $
            alixdPrMinNumber ))

objectclass   ( 1.3.6.1.4.1.107.118.501.110.1   NAME 'alixdFilterEntry' SUP top
    STRUCTURAL
        MUST ( alixdFilterEntryID $ alixdFilterName )
        MAY ( alixdFilterType $ alixdFilterStartMessage $ alixdFilterEndMessage $
            alixdFilterProgramName $ alixdFilterStatus $ alixdFilterErrorInformation ))

objectclass   ( 1.3.6.1.4.1.107.118.501.111.1   NAME 'alixdFilterElemEntry' SUP top
    STRUCTURAL
        MUST ( alixdFilterElemEntryID $ alixdFilterElemFilterName $
            alixdFilterElemIndex )
        MAY ( alixdFilterElemStatus $ alixdFilterElemType $
            alixdFilterElemExpression $ alixdFilterElemCount $
            alixdFilterElemThreshold $ alixdFilterElemDuration $
            alixdFilterElemLastMatchTime $ alixdFilterElemLastMessage $
            alixdFilterElemTriggering $ alixdFilterElemAction $ alixdFilterElemSeverity
            ))

objectclass   ( 1.3.6.1.4.1.107.118.501.112.1   NAME 'alixdActionEntry' SUP top
    STRUCTURAL
        MUST ( alixdActionEntryID $ alixdActionName )
```

MAY (alixdActionType $ alixdActionDescription $ alixdActionProgramName $
alixdActionArguments $ alixdActionPasswd $ alixdActionDoesSendTrap))

objectclass (1.3.6.1.4.1.107.118.501.120.1.1 NAME 'alixdErrlogEntry' SUP top
STRUCTURAL
MUST (alixdErrlogEntryID $ alErrlogClass $ alErrlogIdentifier)
MAY (alErrlogLabel $ alErrlogType $ alErrlogDescription))

objectclass (1.3.6.1.4.1.107.118.501.120.2.1 NAME 'alixdErrlogElemEntry' SUP top
STRUCTURAL
MUST (alixdErrlogElemEntryID $ alixdErrlogElemId)
MAY (alixdErrlogElemStatus $ alixdErrlogElemCount $
alixdErrlogElemThreshold $ alixdErrlogElemDuration $
alixdErrlogElemLastMatchTime $ alixdErrlogElemLastMessage $
alixdErrlogElemTriggering $ alixdErrlogElemAction))

F.2 Schéma LDAP de l'agent d'extension

attributetype (1.3.6.1.4.1.107.157.24.1 NAME 'OAXname'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{200})

attributetype (1.3.6.1.4.1.107.157.20.1 NAME 'generixPackageEntryID' SUP
OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1 NAME 'generixTableEntryID' SUP OAXname
)

attributetype (1.3.6.1.4.1.107.157.22.1 NAME 'generixAttributeEntryID' SUP
OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1 NAME 'generixDataEntryID' SUP OAXname
)

attributetype (1.3.6.1.4.1.107.157.3 NAME 'generixTraceLevel' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.4 NAME 'generixSaveConfigFlag' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.5 NAME 'generixSaveConfigWhenExitFlag' SUP
OAXname)

attributetype (1.3.6.1.4.1.107.157.6 NAME 'generixOldPasswd' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.7 NAME 'generixNewPasswd' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.8 NAME 'generixPath' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.20.1.1 NAME 'generixPackageName' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.20.1.2 NAME 'generixPackageStatus' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.20.1.3 NAME 'generixPackageErrorInformation' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.1 NAME 'generixTableId' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.2 NAME 'generixTableDescription' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.3 NAME 'generixTableType' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.4 NAME 'generixTableCommand' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.5 NAME 'generixTableStderr' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.6 NAME 'generixTableFile' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.7 NAME 'generixTableValidityDuration' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.8 NAME 'generixTableErrorInformation' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.9 NAME 'generixTablePasswd' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.10 NAME 'generixTableStartLine' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.11 NAME 'generixTableBeginListMeta' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.12 NAME 'generixTableEndListMeta' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.13 NAME 'generixTableEscapeMeta' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.14 NAME 'generixTableCommentChar' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.15 NAME 'generixTableSeparators' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.16 NAME 'generixTableNullField' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.17 NAME 'generixTableFieldNumber' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.21.1.18 NAME 'generixTableFilterValue' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.1 NAME 'generixAttrTableId' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.2 NAME 'generixAttrColumnNumber' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.3 NAME 'generixAttrOid' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.4 NAME 'generixAttrSyntax' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.5 NAME 'generixAttrPasswd' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.22.1.6 NAME 'generixAttrName' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.1 NAME 'generixDataTableId' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.2 NAME 'generixDataName' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.3 NAME 'generixDataLineCount' SUP

OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.4 NAME 'generixData1' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.5 NAME 'generixData2' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.6 NAME 'generixData3' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.7 NAME 'generixData4' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.8 NAME 'generixData5' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.9 NAME 'generixData6' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.10 NAME 'generixData7' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.11 NAME 'generixData8' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.12 NAME 'generixData9' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.13 NAME 'generixData10' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.14 NAME 'generixData11' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.15 NAME 'generixData12' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.16 NAME 'generixData13' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.17 NAME 'generixData14' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.18 NAME 'generixData15' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.19 NAME 'generixData16' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.20 NAME 'generixData17' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.21 NAME 'generixData18' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.22 NAME 'generixData19' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.23 NAME 'generixData20' SUP OAXname)

attributetype (1.3.6.1.4.1.107.157.23.1.24 NAME 'generixData21' SUP OAXname)

```
objectclass   ( 1.3.6.1.4.1.107.157.24   NAME 'OAXconfig' SUP top STRUCTURAL
        MUST ( configName ))

objectclass   ( 1.3.6.1.4.1.107.157   NAME 'generix' SUP top STRUCTURAL
        MUST ( version )
        MAY ( generixTraceLevel $ generixSaveConfigFlag $
            generixSaveConfigWhenExitFlag $ generixOldPasswd $ generixNewPasswd $
             generixPath ))

objectclass   ( 1.3.6.1.4.1.107.157.20.1   NAME 'generixPackageEntry' SUP top
    STRUCTURAL
        MUST ( generixPackageEntryID $ generixPackageName )
        MAY ( generixPackageStatus $ generixPackageErrorInformation ))

objectclass   ( 1.3.6.1.4.1.107.157.21.1   NAME 'generixTableEntry' SUP top
    STRUCTURAL
        MUST ( generixTableEntryID $ generixTableId )
        MAY ( generixTableDescription $ generixTableType $ generixTableCommand $
            generixTableStderr $ generixTableFile $ generixTableValidityDuration $
            generixTableErrorInformation $ generixTablePasswd $
            generixTableStartLine $ generixTableBeginListMeta $
            generixTableEndListMeta $ generixTableEscapeMeta $
            generixTableCommentChar $ generixTableSeparators $
            generixTableNullField $ generixTableFilterValue ))

objectclass   ( 1.3.6.1.4.1.107.157.22.1   NAME 'generixAttributeEntry' SUP top
    STRUCTURAL
        MUST ( generixAttributeEntryID $ generixAttrTableId $
            generixAttrColumnNumber )
        MAY ( generixAttrOid $ generixAttrSyntax $ generixAttrPasswd $
            generixAttrName ))

objectclass   ( 1.3.6.1.4.1.107.157.23.1   NAME 'generixDataEntry' SUP top
    STRUCTURAL
        MUST ( generixDataEntryID $ generixDataTableId $ generixDataName )
        MAY ( generixDataLineCount $ generixData1 $ generixData2 $ generixData3 $
            generixData4 $ generixData5 $ generixData6 $ generixData7 $ generixData8
             $ generixData9 $ generixData10 $ generixData11 $ generixData12 $
            generixData13 $ generixData14 $ generixData15 $ generixData16 $
            generixData17 $ generixData18 $ generixData19 $ generixData20 $
            generixData21 ))
```

F.3 Schéma LDAP de l'agent COACH

attributetype (1.3.6.1.4.1.107.147.1 NAME 'COACHname'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{200})

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.1 NAME 'cfgIndicatorId' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.2 NAME 'cfgIndicatorLabel' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.3 NAME 'cfgIndicatorDomain' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.4 NAME 'cfgIndicatorEquation' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.5 NAME 'cfgIndicatorPeriodPolling' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.6 NAME 'cfgIndicatorThreshold' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.7 NAME 'cfgIndicatorCptMax' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.8 NAME 'cfgIndicatorPeriodValid' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.9 NAME 'cfgIndicatorComparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.10 NAME 'cfgIndicatorLogOn' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.1.1.11 NAME 'cfgIndicatorTrap' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.1 NAME 'cfgFilterId' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.2 NAME 'cfgFilterLabel' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.3 NAME 'cfgFilterDomain' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.4 NAME 'cfgFilterEnterprise' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.5 NAME 'cfgFilterGeneric' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.6 NAME 'cfgFilterSpecific' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.7 NAME 'cfgFilterCptMax' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.2.1.8 NAME 'cfgFilterPeriodValid' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.5 NAME 'cfgDiscoverId' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.1 NAME 'cfgDiscoverPeriod' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.3 NAME 'cfgDiscoverSubnetMask' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.4 NAME 'cfgDiscoverDomainAtNextTime' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.8 NAME 'cfgDiscoverTimeOut' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.9 NAME 'cfgDiscoverRedoPeriod' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.12.1.1 NAME 'cfgDiscoverIp' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.12.1.2 NAME 'cfgDiscoverIpDomain' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.12.1.3 NAME 'cfgDiscoverIpIndex' SUP

COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.11.1.1 NAME 'cfgDiscoverIpLower' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.11.1.2 NAME 'cfgDiscoverIpUpper' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.4.11.1.3 NAME 'cfgDiscoverIpSetIndex' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.1 NAME 'cfgDomainId' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.2 NAME 'cfgDomainLabel' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.3 NAME 'cfgAtt1' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.10 NAME 'cfgAtt1Value' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.11 NAME 'cfgAtt1Comparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.4 NAME 'cfgAtt2' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.12 NAME 'cfgAtt2Value' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.13 NAME 'cfgAtt2Comparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.5 NAME 'cfgAtt3' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.14 NAME 'cfgAtt3Value' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.15 NAME 'cfgAtt3Comparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.6 NAME 'cfgAtt4' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.16 NAME 'cfgAtt4Value' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.17 NAME 'cfgAtt4Comparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.7 NAME 'cfgAtt5' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.18 NAME 'cfgAtt5Value' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.3.1.19 NAME 'cfgAtt5Comparaison' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.1.6.2 NAME 'cfgSystemId' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.6.1 NAME 'systemSaveConfiguration' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.4.1 NAME 'detachTrapLogs' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.4.2 NAME 'detachIndicatorLogs' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.4.3 NAME 'logOnTraps' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.4.4 NAME 'logOnIndicators' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.1.4 NAME 'indicatorMaxTimePolling' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.1.5 NAME 'indicatorPollingUnitPeriod' SUP COACHname)

attributetype (1.3.6.1.4.1.107.146.1.3.5.2 NAME 'systemSnmpCommunity' SUP COACHname)

objectclass (1.3.6.1.4.1.107.146 NAME 'COACHconfig' SUP top STRUCTURAL MUST (configName))

objectclass (1.3.6.1.4.1.107.146.1 NAME 'aic' SUP top STRUCTURAL

```
        MUST ( version ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.1.1   NAME 'cfgIndicatorEntry' SUP top
    STRUCTURAL
        MUST ( cfgIndicatorId $ cfgIndicatorLabel $ cfgIndicatorDomain $
            cfgIndicatorEquation $ cfgIndicatorPeriodPolling $ cfgIndicatorThreshold $
            cfgIndicatorCptMax $ cfgIndicatorPeriodValid $ cfgIndicatorComparaison $
            cfgIndicatorLogOn $ cfgIndicatorTrap ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.2.1   NAME 'cfgFilterEntry' SUP top
    STRUCTURAL
        MUST ( cfgFilterId $ cfgFilterLabel $ cfgFilterDomain $ cfgFilterEnterprise $
            cfgFilterGeneric $  cfgFilterSpecific  $ cfgFilterCptMax $
            cfgFilterPeriodValid ) )

objectclass   ( 1.3.6.1.4.1.107.146.1.1.4   NAME 'cfgDiscoveryEntry' SUP top
    STRUCTURAL
        MUST ( cfgDiscoverId $ cfgDiscoverPeriod $ cfgDiscoverSubnetMask $
            cfgDiscoverDomainAtNextTime $ cfgDiscoverTimeOut $
            cfgDiscoverRedoPeriod ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.4.12.1   NAME 'cfgDiscoverIpEntry' SUP top
    STRUCTURAL
        MUST ( cfgDiscoverIpIndex $ cfgDiscoverIp )
        MAY ( cfgDiscoverIpDomain ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.4.11.1   NAME 'cfgDiscoverIpSetEntry' SUP top
    STRUCTURAL
        MUST ( cfgDiscoverIpSetIndex $ cfgDiscoverIpLower $ cfgDiscoverIpUpper ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.3.1   NAME 'cfgDomainEntry' SUP top
    STRUCTURAL
        MUST ( cfgDomainId $ cfgDomainLabel ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.3.1.20   NAME 'cfgAtt1Entry' SUP top
    STRUCTURAL
        MUST ( cfgAtt1 $ cfgAtt1Value $ cfgAtt1Comparaison ))

objectclass   ( 1.3.6.1.4.1.107.146.1.1.3.1.21   NAME 'cfgAtt2Entry' SUP top
    STRUCTURAL
        MUST ( cfgAtt2 $ cfgAtt2Value $ cfgAtt2Comparaison )).

objectclass   ( 1.3.6.1.4.1.107.146.1.1.3.1.22   NAME 'cfgAtt3Entry' SUP top
```

```
    STRUCTURAL
        MUST ( cfgAtt3 $ cfgAtt3Value $ cfgAtt3Comparaison ))

objectclass    ( 1.3.6.1.4.1.107.146.1.1.3.1.23    NAME 'cfgAtt4Entry' SUP top
    STRUCTURAL
        MUST ( cfgAtt4 $ cfgAtt4Value $ cfgAtt4Comparaison ))

objectclass    ( 1.3.6.1.4.1.107.146.1.1.3.1.24    NAME 'cfgAtt5Entry' SUP top
    STRUCTURAL
        MUST ( cfgAtt5 $ cfgAtt5Value $ cfgAtt5Comparaison ))

objectclass    ( 1.3.6.1.4.1.107.146.2    NAME 'cfgSystemEntry' SUP top STRUCTURAL
        MUST ( cfgSystemId )
        MAY ( systemSaveConfiguration $ detachTrapLogs $ detachIndicatorLogs $
            logOnTraps $ logOnIndicators $ indicatorMaxTimePolling $
            indicatorPollingUnitPeriod $ systemSnmpCommunity ))
```

F.4 Exemple de fichier LDIF pour l'agent de contrôle

```
dn: configName=essai,o=RedEagle Corp.,c=be
configName: essai
objectclass : OACconfig

dn: version=1.0,configName=essai,o=RedEagle Corp.,c=be
version : 1.0
alixdTraceLevel: none
alixdSaveConfigFlag: false
alixdPollingStatus : enabled
alixdPath: icmagent/etc/agix:/icmagent/etc/generix:/icmagent/etc:/icmagent/install/
    install_script :/etc:/bin:/usr/bin:/usr/local/bin:/usr/sbin:/usr/ucb:/usr/dt/bin:/usr
    /bin/X11:/sbin:/home/alain/bin:.:/icmagent/install/instAgent:/icmagent/perl/bin:/
    icmagent/install_tmp/bin:/icmagent:/icmagent/bin:/usr/ccs/bin:/home/ismdev/bin"
alixdMaxTrapsPerMinut: 200
alixActionLogType: overwrite
objectclass : alixd

dn: alixdSourcesEntryID=errlog,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdSourcesEntryID: errlog
alixdSourcesType: errlog
alixdSourcesStatus: enabled
alixdSourcesName: errlog
alixdSourcesLowWaterMark: 0
alixdSourcesHighWaterMark: 0
alixdSourcesFrequency: 300
```

alixdSourcesDefaultAction: Send a trap
objectclass : alixdSourcesEntry

dn: alixdSourcesEntryID=filesystem_/dev/hd1,version=1.0,configName=essai,o=
RedEagle Corp.,c=be
alixdSourcesEntryID: filesystem_/dev/hd1
alixdSourcesType: filesystems
alixdSourcesStatus: enabled
alixdSourcesName: filesystem_/dev/hd1
alixdSourcesLowWaterMark: 90
alixdSourcesHighWaterMark: 90
alixdSourcesFrequency: 180
alixdSourcesDefaultAction: Send a trap
objectclass : alixdSourcesEntry

dn: alixdSourcesEntryID=filesystem_/dev/lv00,version=1.0,configName=essai,o=
RedEagle Corp.,c=be
alixdSourcesEntryID: filesystem_/dev/lv00
alixdSourcesType: filesystems
alixdSourcesStatus: enabled
alixdSourcesName: filesystem_/dev/lv00
alixdSourcesLowWaterMark: 90
alixdSourcesHighWaterMark: 90
alixdSourcesFrequency: 180
alixdSourcesDefaultAction: Send a trap
objectclass : alixdSourcesEntry

dn: alixdProcessEntryID=Zombies.<defunct>,version=1.0,configName=essai,o=
RedEagle Corp.,c=be
alixdProcessEntryID: Zombies.<defunct>
alixdPrProcessNumber: 0
alixdPrProcessName: <defunct>
alixdPrMinNumber: 0
alixdPrMaxNumber: 5
alixdPrGroupName: Zombies
objectclass : alixdProcessEntry

dn: alixdProcessEntryID=agovin.named,version=1.0,configName=essai,o=RedEagle
Corp.,c=be
alixdProcessEntryID: agovin.named
alixdPrProcessNumber: 0
alixdPrProcessName: named
alixdPrMinNumber: 1

alixdPrMaxNumber: 1
alixdPrGroupName: agovin
objectclass : alixdProcessEntry

dn: alixdActionEntryID=Do nothing,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdActionEntryID: Do nothing
alixdActionType: doNothing
alixdActionName: Do nothing
alixdActionDoesSendTrap: false
alixdActionDescription: Do nothing
objectclass : alixdActionEntry

dn: alixdActionEntryID=Send a trap,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdActionEntryID: Send a trap
alixdActionType: sendTrap
alixdActionName: Send a trap
alixdActionDoesSendTrap: true
alixdActionDescription: Send a trap
objectclass : alixdActionEntry

dn: alixdErrlogEntryID=hard.00D2B9FE,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdErrlogEntryID: hard.00D2B9FE
alErrlogType: temp
alErrlogLabel: MPS_RCVRY_EXIT
alErrlogIdentifier : 00D2B9FE
alErrlogDescription: 00D2B9FE MPS_RCVRY_EXIT TEMP H PROBLEM RESOLVED
alErrlogClass: hard
objectclass : alixdErrlogEntry

dn: alixdErrlogEntryID=hard.00E6300A,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdErrlogEntryID: hard.00E6300A
alErrlogType: perm
alErrlogLabel: CFDDI_SELFT_ERR
alErrlogIdentifier : 00E6300A
alErrlogDescription: 00E6300A CFDDI_SELFT_ERR PERM H FDDI SELF TEST ERROR
alErrlogClass: hard
objectclass : alixdErrlogEntry

dn: alixdErrlogEntryID=hard.01D7FC25,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdErrlogEntryID: hard.01D7FC25
alErrlogType: pend
alErrlogLabel: FDDI_RCVRY_ENTER
alErrlogIdentifier : 01D7FC25
alErrlogDescription: 01D7FC25 FDDI_RCVRY_ENTER PEND H RECOVERY LOGIC INITIATED BY DEVICE
alErrlogClass: hard
objectclass : alixdErrlogEntry

dn: alixdErrlogEntryID=soft.A6949CEB,version=1.0,configName=essai,o=RedEagle Corp.,c=be
alixdErrlogEntryID: soft.A6949CEB
alErrlogType: perm
alErrlogLabel: SIODD_PORT_UCFG_MET
alErrlogIdentifier : A6949CEB
alErrlogDescription: A6949CEB SIODD_UCFG_MET PERM S CONFIGURATION OR CUSTOMIZATION ERROR
alErrlogClass: soft
objectclass : alixdErrlogEntry

F.5 Exemple de fichier LDIF pour l'agent d'extension

dn: configName=essai1,o=RedEagle Corp.,c=be
configName: essai1
objectClass: OAXconfig

dn: version=1.0,configName=essai1,o=RedEagle Corp.,c=be
generixTraceLevel: none
version : 1.0
generixPath: /icmagent/etc/agix:/icmagent/etc/generix:/icmagent/etc:/icmagent/install/install_script:/etc:/bin:/usr/bin:/usr/local/bin:/usr/sbin:/usr/ucb:/usr/dt/bin:/usr/bin/X11:/sbin:/home/alain/bin:.:/icmagent/install/instAgent:/icmagent/perl/bin:/icmagent/install_tmp/bin:/icmagent:/icmagent/bin:/usr/ccs/bin:/home/ismdev/bin:/icmagent/etc/generix/internet
generixSaveConfigFlag: false
generixSaveConfigWhenExitFlag: true
objectClass: generix

dn: generixPackageEntryID=edns,version=1.0,configName=essai1,o=RedEagle Corp.,c=be
generixPackageStatus: ok
generixPackageEntryID: edns

```
generixPackageName: edns
objectClass: generixPackageEntry

dn: generixPackageEntryID=einet,version=1.0,configName=essai1,o=RedEagle Corp.,c=
    be
generixPackageStatus: ok
generixPackageEntryID: einet
generixPackageName: einet
objectClass: generixPackageEntry

dn: generixPackageEntryID=generix,version=1.0,configName=essai1,o=RedEagle Corp.,
    c=be
generixPackageStatus: ok
generixPackageEntryID: generix
generixPackageName: generix
objectClass: generixPackageEntry
```

F.6 Exemple de fichier LDIF pour l'agent COACH

```
dn: configName=essai2,o=RedEagle Corp.,c=be
configName: essai2
objectclass : COACHconfig

dn: version=1.0,configName=essai2,o=RedEagle Corp.,c=be
version : 1.0
objectclass : aic

dn: cfgFilterId =3,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgFilterSpecific : 33
cfgFilterPeriodValid: 30
cfgFilterLabel : alix-uxLoginSession-setFailed
cfgFilterId : 3
cfgFilterGeneric: 6
cfgFilterEnterprise : bull.118
cfgFilterDomain: 2
cfgFilterCptMax: 3
objectclass : cfgFilterEntry

dn: cfgFilterId =1,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgFilterSpecific : 4
cfgFilterPeriodValid: 60
cfgFilterLabel : alix-fs-nfull
cfgFilterId : 1
cfgFilterGeneric: 6
```

cfgFilterEnterprise : bull.118
cfgFilterDomain: 2
cfgFilterCptMax: 1
objectclass : cfgFilterEntry

dn: cfgFilterId =2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgFilterSpecific : 5
cfgFilterPeriodValid : 6
cfgFilterLabel : alix−fs−error
cfgFilterId : 2
cfgFilterGeneric : 6
cfgFilterEnterprise : 1.3.6.1.4.1.107.114
cfgFilterDomain: 2
cfgFilterCptMax: 0
objectclass : cfgFilterEntry

dn: cfgIndicatorId=11431,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicAutoTrapWarning
cfgIndicatorThreshold: −1
cfgIndicatorPeriodValid: 1200
cfgIndicatorPeriodPolling: 120
cfgIndicatorLogOn: on
cfgIndicatorLabel: EdnsAvailId
cfgIndicatorId: 11431
cfgIndicatorEquation: (ednsAvailId.?s0)
cfgIndicatorDomain: 16
cfgIndicatorCptMax: 0
cfgIndicatorComparaison: less
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=11450,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicAutoTrapCritical
cfgIndicatorThreshold: 0
cfgIndicatorPeriodValid: 1200
cfgIndicatorPeriodPolling: 120
cfgIndicatorLogOn: off
cfgIndicatorLabel: EdnsConnectAvail1
cfgIndicatorId: 11450
cfgIndicatorEquation: (ednsStatusConnect.?s0)
cfgIndicatorDomain: 16
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: different
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 1
cfgIndicatorPeriodValid: 120
cfgIndicatorPeriodPolling: 120
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifDiscards
cfgIndicatorId : 3
cfgIndicatorEquation: ($-(ifInDiscards.1+ifOutDiscards.1))
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=19,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 1
cfgIndicatorPeriodValid: 300
cfgIndicatorPeriodPolling: 100
cfgIndicatorLogOn: on
cfgIndicatorLabel: NoDisponibility
cfgIndicatorId : 19
cfgIndicatorEquation: ($t/($-(sysUpTime.0)))
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=9,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 2
cfgIndicatorPeriodValid: 620
cfgIndicatorPeriodPolling: 620
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrorsSUM
cfgIndicatorId : 9
cfgIndicatorEquation : (!SUM(ifErrors))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 3
cfgIndicatorPeriodValid: 320
cfgIndicatorPeriodPolling: 320
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifDiscardsAll
cfgIndicatorId : 4
cfgIndicatorEquation: (!SUM(ifDiscards))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=8,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 290
cfgIndicatorPeriodPolling: 290
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrors
cfgIndicatorId : 8
cfgIndicatorEquation: ($-(ifInErrors.1+ifOutErrors.1)/$t)
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=7,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 340
cfgIndicatorPeriodPolling: 340
cfgIndicatorLogOn: on
cfgIndicatorLabel: coachtcpRetransSegs
cfgIndicatorId : 7
cfgIndicatorEquation: (tcpRetransSegs.0)
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=10,version=1.0,configName=essai2,o=RedEagle Corp.,c=be

cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 630
cfgIndicatorPeriodPolling: 630
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrorsMOY
cfgIndicatorId: 10
cfgIndicatorEquation: (!MOY(ifErrors)*100)
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 280
cfgIndicatorPeriodPolling: 280
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifInPackets
cfgIndicatorId: 11
cfgIndicatorEquation: (ifInUcastPkts.1+ifInNUcastPkts.1)/$t
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=12,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 280
cfgIndicatorPeriodPolling: 280
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifOutPackets
cfgIndicatorId: 12
cfgIndicatorEquation: (ifOutUcastPkts.1+ifOutNUcastPkts.1)/$t
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical

cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 330
cfgIndicatorPeriodPolling: 330
cfgIndicatorLogOn: on
cfgIndicatorLabel: coachifOutQlen
cfgIndicatorId: 5
cfgIndicatorEquation: (ifOutQlen.1)
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=13,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 570
cfgIndicatorPeriodPolling: 570
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrorsRatio
cfgIndicatorId: 13
cfgIndicatorEquation: (&ifErrors/(&ifInPackets+&ifOutPackets))
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=18,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 300
cfgIndicatorPeriodPolling: 300
cfgIndicatorLogOn: on
cfgIndicatorLabel: ipInputErrorsPercentOnLink
cfgIndicatorId: 18
cfgIndicatorEquation: (!SUM(ipInputErrorsPercent))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass: cfgIndicatorEntry

dn: cfgIndicatorId=14,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5

cfgIndicatorPeriodValid: 1220
cfgIndicatorPeriodPolling: 1220
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrorsRatioLinkMOY
cfgIndicatorId: 14
cfgIndicatorEquation: (!MOY(ifErrorsRatio))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=17,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 650
cfgIndicatorPeriodPolling: 650
cfgIndicatorLogOn: on
cfgIndicatorLabel: ipInputErrorsPercent
cfgIndicatorId: 17
cfgIndicatorEquation: (&ipInputErrors/(&ipInputErrors/($−(ipIndelivers.0)))
*100
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 5
cfgIndicatorPeriodValid: 650
cfgIndicatorPeriodPolling: 650
cfgIndicatorLogOn: on
cfgIndicatorLabel: ipInputErrors
cfgIndicatorId: 16
cfgIndicatorEquation: ($−(ipInHdrErrors.0+ipInAddrErrors.0))
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 10

cfgIndicatorPeriodValid: 3600
cfgIndicatorPeriodPolling: 1210
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifUtilizationBandWithall
cfgIndicatorId : 2
cfgIndicatorEquation : (!SUM(ifUtilizationBandWith))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=1,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 10
cfgIndicatorPeriodValid: 1200
cfgIndicatorPeriodPolling: 600
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifUtilizationBandWith
cfgIndicatorId : 1
cfgIndicatorEquation: (8*$-(ifInOctets.1+ifOutOctets.1)/$t)/ifSpeed.1
cfgIndicatorDomain: 2
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=15,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 20
cfgIndicatorPeriodValid: 1220
cfgIndicatorPeriodPolling: 1220
cfgIndicatorLogOn: on
cfgIndicatorLabel: ifErrorsRatioLinkSUM
cfgIndicatorId : 15
cfgIndicatorEquation : (!SUM(ifErrorsRatio))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=6,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 50
cfgIndicatorPeriodValid: 670

cfgIndicatorPeriodPolling: 670
cfgIndicatorLogOn: on
cfgIndicatorLabel: coachIfOutQlenAll
cfgIndicatorId : 6
cfgIndicatorEquation : (! SUM(coachIfOutQlen))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=20,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicTrapCritical
cfgIndicatorThreshold: 100
cfgIndicatorPeriodValid: 300
cfgIndicatorPeriodPolling: 150
cfgIndicatorLogOn: on
cfgIndicatorLabel: NoDisponibilityOnLink
cfgIndicatorId : 20
cfgIndicatorEquation : (! SUM(NoDisponibility))
cfgIndicatorDomain: 3
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=11422,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicMibTrapWarning
cfgIndicatorThreshold: 900
cfgIndicatorPeriodValid: 1200
cfgIndicatorPeriodPolling: 120
cfgIndicatorLogOn: off
cfgIndicatorLabel: EdnsTotalUsageMem
cfgIndicatorId: 11422
cfgIndicatorEquation: (ednsUsageMem.?s.0(="total_dns_Usage"))
cfgIndicatorDomain: 16
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgIndicatorId=11421,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgIndicatorTrap: aicMibTrapWarning
cfgIndicatorThreshold: 900
cfgIndicatorPeriodValid: 1200
cfgIndicatorPeriodPolling: 120

cfgIndicatorLogOn: off
cfgIndicatorLabel: EdnsTotalUsageCpu
cfgIndicatorId: 11421
cfgIndicatorEquation: (ednsUsageCpu.?s0(="total_dns_Usage"))
cfgIndicatorDomain: 16
cfgIndicatorCptMax: 1
cfgIndicatorComparaison: greater
objectclass : cfgIndicatorEntry

dn: cfgDiscoverId=0,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDiscoverTimeOut: 80
cfgDiscoverSubnetMask: 255.255.255.255
cfgDiscoverRedoPeriod: 3
cfgDiscoverPeriod: 50
cfgDiscoverId: 0
cfgDiscoverDomainAtNextTime: no
objectclass : cfgDiscoveryEntry

dn: cfgDiscoverIpSetIndex=1,cfgDiscoverId=0,version=1.0,configName=essai2,o=
RedEagle Corp.,c=be
cfgDiscoverIpUpper: *.*.*.254
cfgDiscoverIpSetIndex: 1
cfgDiscoverIpLower: *.*.*.1
objectclass : cfgDiscoverIpSetEntry

dn: cfgSystemId=0,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgSystemId: 0
systemSnmpCommunity: public
systemSaveConfiguration: off
logOnTraps: on
logOnIndicators: on
indicatorPollingUnitPeriod: 10
indicatorMaxTimePolling: 15
detachTrapLogs: off
detachIndicatorLogs: off
objectclass : cfgSystemEntry

dn: cfgDomainId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 3
cfgDomainLabel: coach
objectclass : cfgDomainEntry

dn: cfgAtt1 =1.3.6.1.4.1.107.146.1.3.4.3.0, cfgDomainId=3,version=1.0,configName=

```
    essai2,o=RedEagle Corp.,c=be
cfgAtt1Value: value
cfgAtt1Comparaison: exist
cfgAtt1  : 1.3.6.1.4.1.107.146.1.3.4.3.0
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=
    be
cfgAtt2Value: value
cfgAtt2Comparaison: exist
cfgAtt2: 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=
    be
cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3: 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=
    be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4: 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=3,version=1.0,configName=essai2,o=RedEagle Corp.,c=
    be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5: 0.0
objectclass : cfgAtt5Entry

dn: cfgDomainId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 2
cfgDomainLabel: mib 2
objectclass : cfgDomainEntry

dn: cfgAtt1 =1.3.6.1.2.1.1.3.0, cfgDomainId=2,version=1.0,configName=essai2,o=
    RedEagle Corp.,c=be
cfgAtt1Value: value
cfgAtt1Comparaison: exist
```

cfgAtt1 : 1.3.6.1.2.1.1.3.0
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt2Value: value
cfgAtt2Comparaison: exist
cfgAtt2: 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3: 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4: 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=2,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5: 0.0
objectclass : cfgAtt5Entry

dn: cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 4
cfgDomainLabel: UNIX System
objectclass : cfgDomainEntry

dn: cfgAtt1 =1.3.6.1.4.1.107.118.2.0, cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt1Value: "UNIX"
cfgAtt1Comparaison: exist
cfgAtt1 : 1.3.6.1.4.1.107.118.2.0
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=
be
cfgAtt2Value: value
cfgAtt2Comparaison: exist
cfgAtt2: 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=
be
cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3: 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=
be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4: 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=4,version=1.0,configName=essai2,o=RedEagle Corp.,c=
be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5: 0.0
objectclass : cfgAtt5Entry

dn: cfgDomainId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 5
cfgDomainLabel: NT System
objectclass : cfgDomainEntry

dn: cfgAtt1 =1.3.6.1.4.1.107.136.1.1.0, cfgDomainId=5,version=1.0,configName=essai2,o
=RedEagle Corp.,c=be
cfgAtt1Value: value
cfgAtt1Comparaison: exist
cfgAtt1 : 1.3.6.1.4.1.107.136.1.1.0
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=
be
cfgAtt2Value: value

cfgAtt2Comparaison: exist
cfgAtt2 : 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3 : 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4 : 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=5,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5 : 0.0
objectclass : cfgAtt5Entry

dn: cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 11
cfgDomainLabel: inconnu
objectclass : cfgDomainEntry

dn: cfgAtt1=0.0,cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt1Value: value
cfgAtt1Comparaison: exist
cfgAtt1 : 0.0
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt2Value: value
cfgAtt2Comparaison: exist
cfgAtt2 : 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3: 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4: 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=11,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5: 0.0
objectclass : cfgAtt5Entry

dn: cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgDomainId: 16
cfgDomainLabel: einet
objectclass : cfgDomainEntry

dn: cfgAtt1 =1.3.6.1.4.1.107.195.5.1.1.1, cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt1Value: value
cfgAtt1Comparaison: node
cfgAtt1 : 1.3.6.1.4.1.107.195.5.1.1.1
objectclass : cfgAtt1Entry

dn: cfgAtt2=0.0,cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt2Value: value
cfgAtt2Comparaison: exist
cfgAtt2: 0.0
objectclass : cfgAtt2Entry

dn: cfgAtt3=0.0,cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be

cfgAtt3Value: value
cfgAtt3Comparaison: exist
cfgAtt3 : 0.0
objectclass : cfgAtt3Entry

dn: cfgAtt4=0.0,cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt4Value: value
cfgAtt4Comparaison: exist
cfgAtt4 : 0.0
objectclass : cfgAtt4Entry

dn: cfgAtt5=0.0,cfgDomainId=16,version=1.0,configName=essai2,o=RedEagle Corp.,c=be
cfgAtt5Value: value
cfgAtt5Comparaison: exist
cfgAtt5 : 0.0
objectclass : cfgAtt5Entry

F.7 Exemple d'un programme Java pour l'importation de données

REM : Une importation de données se fait d'un annuaire LDAP vers une feuille XML

```
import java.util.Enumeration;
import javax.naming.*;
import java.io.*;
import javax.naming.directory.*;
import java.util.Hashtable;
import java.util.StringTokenizer;
import java.util.NoSuchElementException;
import java.util.Vector;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import org.w3c.dom.*;

public class ldapImport {

    private static String configName;
    private static String version;
    private static PrintWriter out;

    private static Hashtable entite_predefinies = new Hashtable();

    private static void initEntitePredefinies ()
```

```
    {
     entite_predefinies .put("\"","&#34;");
     entite_predefinies .put("&","&#38;");
     entite_predefinies .put("'","&#39;");
     entite_predefinies .put("<","&#60;");
     entite_predefinies .put(">","&#62;");
    }

private static Vector init(String agentType)
    {
    Vector v = new Vector();
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    dbf.setNamespaceAware(true);
    try {
        DocumentBuilder db = dbf.newDocumentBuilder();
        db.setErrorHandler(new MyErrorHandler(System.err));
        Document doc = db.parse(agentType+".xsd");
        NodeList n = doc.getElementsByTagName("xsd:appinfo");
        String index="";
        for(int i=0;i<n.getLength();i++)
            {
            Node valeur = n.item(i).getParentNode().getParentNode().getAttributes
                ().getNamedItem("name");
            if(valeur!=null && valeur.getNodeValue().equals(agentType))
                {index=n.item(i).getFirstChild().getNodeValue();}
            }
        StringTokenizer tokenizer = new StringTokenizer(index,"|");
        while(tokenizer.hasMoreTokens())
            {v.addElement(tokenizer.nextToken());}
        }
    catch(SAXException e){System.out.println(e);}
    catch(IOException e){System.out.println(e);}
    catch(ParserConfigurationException e){System.out.println(e);}
    return v;
    }

private static void importation(DirContext ctx,Vector v,String entete,String
    agentType)
    {
    try
       {
       SearchControls constraints = new SearchControls();
       constraints.setSearchScope(SearchControls.SUBTREE_SCOPE);
```

```
String token = new String("");
out.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
out.println("<"+agentType+" xmlns:xsi=\"http://www.w3.org/2001/
    XMLSchema-instance\" xsi:noNamespaceSchemaLocation=\""+
    agentType+".xsd\">");
for(int i=0;i<v.size();i++)
    {
    String param = "objectclass="+v.elementAt(i);
    NamingEnumeration Instanceclass = ctx.search("version="+version+",
        configName="+configName,param, constraints);
    if(Instanceclass != null && Instanceclass.hasMore())
        {
        out.println("<"+v.elementAt(i)+">");
        while (Instanceclass.hasMore())
            {
            SearchResult tableau = (SearchResult)Instanceclass.next();
            String nom = tableau.getName();
            if (!nom.equals("") && nom.charAt(0)=='')
                {nom=nom.substring(1,nom.length()-1);}
            StringTokenizer tokenizer = new StringTokenizer(nom,"=");
            if(tokenizer.countTokens()!=0)
                {token = tokenizer.nextToken();}
                javax.naming.directory.Attributes attrs = tableau.
                    getAttributes();
                if (attrs == null)
                    {System.out.println("\tNo attributes");}
                else
                    {
                    String header = "Instance"+((String)v.elementAt(i)).
                        substring(entete.length());
                    if (!v.elementAt(i).equals(entete))
                        {out.println("\t<"+header+">");}
                    for (NamingEnumeration instance = attrs.getAll();
                        instance.hasMoreElements();)
                        {
                        javax.naming.directory.Attribute attr = (Attribute
                            )instance.next();
                        String attrId = attr.getID();
                        for (Enumeration valeurs = attr.getAll();valeurs.
                            hasMoreElements();)
                            {
                            if (!attrId.equals(token) && !attrId.equals("
                                objectClass") && !attrId.equals("version"))
```

```
                                        {out.println("\t\t<"+attrId+">"+(
                                            conversion((String)valeurs.nextElement
                                            ()).trim())+"</"+attrId+">");}
                                    else
                                        {valeurs.nextElement();}
                                    }
                                }
                            if (!v.elementAt(i).equals(entete))
                                {out.println("\t</"+header+">\n");}
                            }
                        }
                    out.println("</"+v.elementAt(i)+">");
                    }
                }
            out.println("</"+agentType+">");
            }
        catch (NamingException e) {e.printStackTrace();}
        catch (NoSuchElementException e) {e.printStackTrace();}
        }

    private static String conversion(String chaine)
        {
        String result="";
        String c;
        for(int i=0;i<chaine.length();i++)
            {
            c=chaine.substring(i,i+1);
            if( entite_predefinies .get(c)!=null)
                {result=result+entite_predefinies.get(c);}
            else
                {result=result+c;}
            }
        return result;
        }

    private static void usage()
        {
        System.out.println("Description : allow the importation of an Open agent
            configuration from a LDAP directory service");
        System.out.println("Usage : \"java    ... \" ldapExport <configuration_name
            > <version>");
        }
```

```
private static void XML_attributs(String element,NamingEnumeration
    Instanceclass,String ecart)
    {
    try{
        if(Instanceclass != null && Instanceclass.hasMore())
            {
            while (Instanceclass.hasMore())
                {
                out.println(ecart+"<"+element+">");
                SearchResult tableau = (SearchResult)Instanceclass.next();
                String nom = tableau.getName();
                javax.naming.directory.Attributes attrs = tableau.getAttributes();
                for (NamingEnumeration instance = attrs.getAll();instance.
                    hasMoreElements();)
                    {
                    javax.naming.directory.Attribute attr = (Attribute)instance.
                        next();
                    String attrId = attr.getID();
                    for (Enumeration valeurs = attr.getAll();valeurs.
                        hasMoreElements();)
                        {
                        if (!attrId.equals("objectClass") && !attrId.equals("
                            cfgSystemId"))
                            {out.println(ecart+"    <"+attrId+">"+(conversion((
                                String)valeurs.nextElement()).trim())+"</"+attrId
                                +">");}
                        else
                            {valeurs.nextElement();}
                        }
                    }
                out.println(ecart+"</"+element+">\n");
                }
            }
        }
    catch (NamingException e) {e.printStackTrace();}
    }

private static void confmodImportation(DirContext ctx)
    {
    try {
        SearchControls constraints = new SearchControls();
        constraints.setSearchScope(SearchControls.SUBTREE_SCOPE);
        out.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
```

```
        out.println("<COACHconfig xmlns:xsi=\"http://www.w3.org/2001/
            XMLSchema-instance\" xsi:noNamespaceSchemaLocation=\"
            COACHconfig.xsd\">\n");
        out.println("    <confmod.ini>");
        XML_attributs("filter",ctx.search("version="+version+",configName="+
            configName,"objectclass=cfgFilterEntry", constraints),"\t");
        XML_attributs("indicator",ctx.search("version="+version+",configName
            ="+configName,"objectclass=cfgIndicatorEntry", constraints),"\t");
        out.println("    </confmod.ini>\n");
        }
    catch (NamingException e) {e.printStackTrace();}
    }

private static void aicImportation(DirContext ctx)
    {
    try {
        SearchControls constraints = new SearchControls();
        constraints.setSearchScope(SearchControls.SUBTREE_SCOPE);
        out.println("    <aic.inst>\n");
        NamingEnumeration Instanceclass = ctx.search("version="+version+",
            configName="+configName,"objectclass=cfgDiscoveryEntry",
            constraints);
        try {
            String chaine="<cfgDiscovery ";
            SearchResult tableau = (SearchResult)Instanceclass.next();
            String nom = tableau.getName();
            javax.naming.directory.Attributes attrs = tableau.getAttributes();
            for (NamingEnumeration instance = attrs.getAll();instance.
                hasMoreElements();)
                {
                javax.naming.directory.Attribute attr = (Attribute)instance.next();
                String attrId = attr.getID();
                for (Enumeration valeurs = attr.getAll();valeurs.hasMoreElements
                    ();)
                    {
                    if (!attrId.equals("objectClass") && !attrId.equals("
                        cfgDiscoverId"))
                        {chaine+=(attrId+"=\""+conversion((String)valeurs.
                            nextElement()).trim())+"\" ";}
                    else
                        {valeurs.nextElement();}
                    }
                }
```

```
out.println("\t"+chaine.trim()+">\n");
XML_attributs("cfgDiscoverIpEntry",ctx.search("version="+version+",
    configName="+configName,"objectclass=cfgDiscoverIpEntry",
    constraints),"\t ");
XML_attributs("cfgDiscoverIpSetEntry",ctx.search("version="+version
    +",configName="+configName,"objectclass=cfgDiscoverIpsetEntry
    ", constraints),"\t ");
out.println("\t</cfgDiscovery>\n");
XML_attributs("cfgSystem",ctx.search("version="+version+",
    configName="+configName,"objectclass=cfgSystemEntry",
    constraints),"\t");
String ID="";
Instanceclass = ctx.search("version="+version+",configName="+
    configName,"objectclass=cfgDomainEntry", constraints);
if(Instanceclass != null && Instanceclass.hasMore())
    {
    while (Instanceclass.hasMore())
        {
        out.println("\t<cfgDomain>");
        SearchResult tableauDomain = (SearchResult)Instanceclass.
            next();
        String nomDomain = tableauDomain.getName();
        if(nomDomain.startsWith("cfgDomainId"))
            {ID=nomDomain;}
        javax.naming.directory.Attributes attrsDomain =
            tableauDomain.getAttributes();
        for (NamingEnumeration instance = attrsDomain.getAll();
            instance.hasMoreElements();)
            {
            javax.naming.directory.Attribute attr = (Attribute)instance
                .next();
            String attrId = attr.getID();
            for (Enumeration valeurs = attr.getAll();valeurs.
                hasMoreElements();)
                {
                if (!attrId.equals("objectClass"))
                    {out.println("\t     <"+attrId+">"+(conversion((
                        String)valeurs.nextElement()).trim())+"</"+
                        attrId+">");}
                else
                    {valeurs.nextElement();}
                }
        }
```

```
                    NamingEnumeration InstancecfgAtt = ctx.search(ID+",version
                        ="+version+",configName="+configName,"objectclass
                        =*", constraints);
                    while(InstancecfgAtt != null && InstancecfgAtt.hasMore())
                        {
                        SearchResult tableaucfgAtt = (SearchResult)InstancecfgAtt.
                            next();
                        String nomcfgAtt = tableaucfgAtt.getName();
                        if(nomcfgAtt.startsWith("cfgAtt"))
                            {
                            String chainecfgAtt="<"+nomcfgAtt.substring(0,7)
                                +" ";
                            javax.naming.directory.Attributes attrscfgAtt =
                                tableaucfgAtt.getAttributes();
                            for (NamingEnumeration instance = attrscfgAtt.getAll
                                ();instance.hasMoreElements();)
                                {
                                javax.naming.directory.Attribute attr = (Attribute
                                    )instance.next();
                                String attrId = attr.getID();
                                for (Enumeration valeurs = attr.getAll();valeurs.
                                    hasMoreElements();)
                                    {
                                    if (!attrId.equals("objectClass"))
                                        {chainecfgAtt+=(attrId+"=\""+
                                            conversion((String)valeurs.nextElement
                                            ()).trim())+"\" ";}
                                    else
                                        {valeurs.nextElement();}
                                    }
                                }
                            out.println("\t     "+chainecfgAtt.trim()+"/>");
                            }
                        }
                    out.println("\t</cfgDomain>\n");
                    }
                }
            }
        catch (NamingException e) {e.printStackTrace();}
            out.println("     </aic.inst>\n");
            out.println("</COACHconfig>");
            }
        catch (NamingException e) {e.printStackTrace();}
```

```
    }

public static void main (String[] argv)
    {
    try {
        configName=argv[0];
        version=argv[1];
        Vector v = new Vector();
        initEntitePredefinies ();
        Hashtable env = new Hashtable(5, 0.75f);
        env.put(Context.INITIAL_CONTEXT_FACTORY,"com.sun.jndi.ldap.
            LdapCtxFactory");
        env.put(Context.PROVIDER_URL, "ldap://redscorpion:389/o=RedEagle
            Corp.,c=be");
        DirContext ctx = new InitialDirContext(env);
        javax.naming.directory.Attributes attrs = ctx.getAttributes("configName
            ="+configName);
        if (attrs.get(" objectclass").get().equals("OACconfig"))
            {
            System.out.println("Importation of a control open agent configuration
                ;-)");
            out = new PrintWriter(new FileWriter("OACconfig_"+configName+"_
                "+version+".xml"));
            v=init("OACconfig");
            importation(ctx,v," alixd","OACconfig");
            System.out.println(" File OACconfig_"+configName+"_"+version+".xml
                generated ;-)");
            out.close();
            }
        else if (attrs.get(" objectclass").get().equals("OAXconfig"))
            {
            System.out.println("Importation of an Extension open agent
                configuration ;-)");
            out = new PrintWriter(new FileWriter("OAXconfig_"+configName+"_
                "+version+".xml"));
            v=init("OAXconfig");
            importation(ctx,v,"generix","OAXconfig");
            System.out.println(" File OAXconfig_"+configName+"_"+version+".xml
                generated ;-)");
            out.close();
            }
        else if (attrs.get(" objectclass").get().equals("COACHconfig"))
            {
```

```
            System.out.println("Importation of a COACH open agent configuration
                ;-)");
            out = new PrintWriter(new FileWriter("COACHconfig_"+configName
                +"_"+version+".xml"));
            confmodImportation(ctx);
            aicImportation(ctx);
            System.out.println("File COACHconfig_"+configName+"_"+version+".
                xml generated ;-)");
            out.close();
            }
        else
            {System.out.println("Unsupported agent type :-(");}
        }
    catch (NamingException e) {e.printStackTrace();}
    catch (NoSuchElementException e) {e.printStackTrace();}
    catch(IOException e){System.out.println(e);}
    catch(ArrayIndexOutOfBoundsException e){usage();}
    }
private static class MyErrorHandler implements ErrorHandler {

    private PrintStream out;

    MyErrorHandler(PrintStream out) {
        this.out = out;
    }

    private String getParseExceptionInfo(SAXParseException spe) {
        String systemId = spe.getSystemId();
        if (systemId == null) {
            systemId = "null";
        }
        String info = "URI=" + systemId +
            " Line=" + spe.getLineNumber() +
            ": " + spe.getMessage();
        return info;
    }

    public void warning(SAXParseException spe) throws SAXException {
        out.println("Warning: " + getParseExceptionInfo(spe));
    }

    public void error(SAXParseException spe) throws SAXException {
        String message = "Error: " + getParseExceptionInfo(spe);
```

```
            throw new SAXException(message);
        }

        public void fatalError(SAXParseException spe) throws SAXException {
            String message = "Fatal Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }
    }
}
```

F.8 Exemple d'un programme Java pour l'Exportation de données

REM : Une Exportation de données se fait d'une feuille XML vers un annuaire LDAP

```
import javax.naming.*;
import javax.naming.directory.*;
import java.util.Hashtable;
import java.util.Vector;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import org.w3c.dom.*;
import java.io.*;

class ldapExport {

    private static String configName;
    private static String version;
    private static String base_gen;
    private static Vector dn_vecteur=new Vector();

    private static Document init_Parsing(String file)
        {
        Document doc=null;
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        dbf.setNamespaceAware(true);
        dbf.setValidating(true);
        dbf.setAttribute("http://java.sun.com/xml/jaxp/properties/schemaLanguage
            ","http://www.w3.org/2001/XMLSchema");
        try {
            DocumentBuilder db = dbf.newDocumentBuilder();
            db.setErrorHandler(new MyErrorHandler(System.err));
            doc = db.parse(file);
            }
```

```
    catch(SAXException e){System.out.println(e);}
    catch(IOException e){System.out.println(e);}
    catch(ParserConfigurationException e){System.out.println(e);}
    return doc;
    }

private static void init_ajout (Document doc,DirContext ctx,String agentType,
    String entete)
    {
    try {
        javax.naming.directory.Attribute objClasses_agentType = new
            BasicAttribute("objectclass");
        javax.naming.directory.Attributes agent = new BasicAttributes();
        objClasses_agentType.add(agentType);
        agent.put(objClasses_agentType);
        agent.put(new BasicAttribute("configName",configName));
        ctx.createSubcontext(base_gen,agent);
        System.out.println( "Added entry " + base_gen + ".");
        dn_vecteur.addElement(base_gen);
        ajout(ctx,base_gen.substring(1,base_gen.length()-1),doc.
            getElementsByTagName(entete).item(0),"version=");
        doc.getElementsByTagName(agentType).item(0).removeChild(doc.
            getElementsByTagName(entete).item(0));
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + base_gen + " already exists, no need to add
            ");
        annulation_insertion(ctx);
        System.exit(0);
        }
    catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static void ajout(DirContext ctx,String base,Node instances,String ID)
    {
    NodeList Childs = instances.getChildNodes();
    String result =recherche_cle(Childs);
```

```
String dn="";
try {
    javax.naming.directory.Attribute objClasses = new BasicAttribute("
        objectclass");
    javax.naming.directory.Attributes attr = new BasicAttributes();
    attr.put(objClasses);
    if(result!=null)
        {
        dn="\""+ID+"="+result+","+base+"\"";
        attr.put(new BasicAttribute(ID,result));
        objClasses.add(instances.getParentNode().getNodeName());
        }
    else
        {
        dn="\""+ID+version+","+base+"\"";
        attr.put(new BasicAttribute("version",version));
        objClasses.add(instances.getNodeName());
        base_gen=dn.substring(1,dn.length()-1);
        }
    for(int j=1;j<Childs.item(0).getParentNode().getChildNodes().getLength()
        -1;j++)
        {
        Node elt = Childs.item(0).getParentNode().getChildNodes().item(j);
        if (! elt.getNodeName().equals("#text"))
            {attr.put(new BasicAttribute(elt.getNodeName(),elt.getChildNodes
                ().item(0).getNodeValue()));}
        }
    ctx.createSubcontext(dn,attr);
    dn_vecteur.addElement(dn);
    System.out.println( "Added entry " + dn + ".");
    }
catch (NameAlreadyBoundException e)
    {
    System.out.println("Entry " + dn + " already exists, no need to add");
    annulation_insertion(ctx);
    System.exit(0);
    }
catch (NamingException e)
    {
    System.err.println(e);
    annulation_insertion(ctx);
    System.exit(0);
    }
```

```
    }

private static String recherche_cle(NodeList element)
    {
    Hashtable t = new Hashtable();
    String cle="";
    try {
        for(int i=1;i<element.getLength();i++)
            {
            if(element.item(i).hasAttributes())
                {t.put(element.item(i).getAttributes().getNamedItem("index").
                    getNodeValue(),element.item(i).getFirstChild().getNodeValue())
                    ;}
            }
        for(int j=1;j<=t.size();j++)
            {cle=cle+"."+t.get(Integer.toString(j));}
        return cle.substring(1);
        }
    catch(NullPointerException e){return null;}
    catch(StringIndexOutOfBoundsException e){return null;}
    }

private static void insertion(Hashtable env,String type,String entete,Document
    doc,String agentType)
    {
     System.out.println("Exportation of a "+type+" configuration to the LDAP
         server : configName="+configName+",version="+version);
     base_gen = "\"configName="+configName+"\"";
     DirContext ctx = null;
     try {
         ctx = new InitialDirContext(env);
         init_ajout(doc,ctx,agentType,entete);
         NodeList Childs = doc.getElementsByTagName(agentType).item(0).
             getChildNodes();
         for(int i=1;i<Childs.getLength()-1;i++)
             {
             Node tableaux = Childs.item(i);
             for(int j=1;j<tableaux.getChildNodes().getLength();j++)
                 {
                 Node instances = tableaux.getChildNodes().item(j);
                 if(!instances.getNodeName().equals("#text"))
                     {ajout(ctx,base_gen,instances,instances.getParentNode().
                         getNodeName()+"ID");}
```

```
                }
            }
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + base_gen + " already exists, no need to add
            ");
        annulation_insertion(ctx);
        System.exit(0);
        }
    catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static void COACHinsertion(Hashtable env,Document doc)
    {
    System.out.println("Exportation of a COACH configuration to the LDAP server
        : configName="+configName+",version="+version);
    base_gen="\"configName="+configName+"\"";
    String init_dn="";
    DirContext ctx = null;
    try {
        ctx = new InitialDirContext(env);
        COACHinit_ajout(ctx);
        init_dn=base_gen;
        NodeList Childs = doc.getElementsByTagName("confmod.ini").item(0).
            getChildNodes();
        for(int i=1;i<Childs.getLength()-1;i++)
            {
            if (!Childs.item(i).getNodeName().equals("#text"))
                {confmodAjout(ctx,Childs.item(i));}
            }
        Node discovery = doc.getElementsByTagName("cfgDiscovery").item(0);
        NamedNodeMap attributs = discovery.getAttributes();
        base_gen="\""+attributs.getNamedItem("cfgDiscoverId").getNodeName()
            +"="+attributs.getNamedItem("cfgDiscoverId").getNodeValue()+","+
            base_gen.substring(1);
        javax.naming.directory.Attribute objClasse_discovery = new BasicAttribute
            ("objectclass");
```

```
javax.naming.directory.Attributes attr_discovery = new BasicAttributes();
objClasse_discovery.add("cfgDiscoveryEntry");
attr_discovery.put(objClasse_discovery);
for(int i=0;i<attributs.getLength();i++)
    {
    Node elt = attributs.item(i);
    if (! elt.getNodeName().equals("#text"))
        {attr_discovery.put(new BasicAttribute(elt.getNodeName(),elt.
            getFirstChild().getNodeValue()));}
    }
ctx.createSubcontext(base_gen,attr_discovery);
dn_vecteur.addElement(base_gen);
System.out.println( "Added entry " + base_gen + ".");
NodeList discoveryChilds = discovery.getChildNodes();
for(int i=1;i<discoveryChilds.getLength()-1;i++)
    {
    if (!discoveryChilds.item(i).getNodeName().equals("#text"))
        {aicAjout(ctx,discoveryChilds.item(i));}
    }
base_gen=init_dn;
Node system = doc.getElementsByTagName("cfgSystem").item(0);
if(system!=null)
    {
    attributs = system.getAttributes();
    base_gen="\""+attributs.getNamedItem("cfgSystemId").getNodeName
        ()+"="+attributs.getNamedItem("cfgSystemId").getNodeValue()
        +","+base_gen.substring(1);
    javax.naming.directory.Attribute objClasse_cfgSystem = new
        BasicAttribute("objectclass");
    javax.naming.directory.Attributes attr_cfgSystem = new
        BasicAttributes();
    objClasse_cfgSystem.add("cfgSystemEntry");
    attr_cfgSystem.put(objClasse_cfgSystem);
    attr_cfgSystem.put(attributs.getNamedItem("cfgSystemId").
        getNodeName(),attributs.getNamedItem("cfgSystemId").
        getNodeValue());
    NodeList systemChilds = system.getChildNodes();
    for(int i=1;i<systemChilds.getLength()-1;i++)
        {
        Node elt = systemChilds.item(i);
        if (! elt.getNodeName().equals("#text"))
            {attr_cfgSystem.put(new BasicAttribute(elt.getNodeName(),elt.
                getFirstChild().getNodeValue()));}
```

```
                }
            ctx.createSubcontext(base_gen,attr_cfgSystem);
            dn_vecteur.addElement(base_gen);
            System.out.println( "Added entry " + base_gen + ".");
            }
        base_gen=init_dn;
        NodeList cfgDomain = doc.getElementsByTagName("cfgDomain");
        DomainAjout(ctx,cfgDomain);
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + base_gen + " already exists, no need to add
            ");
        annulation_insertion(ctx);
        System.exit(0);
        }
    catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static void COACHinit_ajout(DirContext ctx)
    {
    try {
        javax.naming.directory.Attribute objClasses_agentType = new
            BasicAttribute("objectclass");
        javax.naming.directory.Attributes agent = new BasicAttributes();
        objClasses_agentType.add("COACHconfig");
        agent.put(objClasses_agentType);
        agent.put(new BasicAttribute("configName",configName));
        ctx.createSubcontext(base_gen,agent);
        System.out.println( "Added entry " + base_gen + ".");
        dn_vecteur.addElement(base_gen);
        javax.naming.directory.Attribute objClasses_aic = new BasicAttribute("
            objectclass");
        javax.naming.directory.Attributes aic = new BasicAttributes();
        objClasses_aic.add("aic");
        aic.put(objClasses_aic);
        aic.put(new BasicAttribute("version",version));
        base_gen="\"version="+version+","+base_gen.substring(1);
```

```
            ctx.createSubcontext(base_gen,aic);
            System.out.println( "Added entry " + base_gen + ".");
            dn_vecteur.addElement(base_gen);
            }
        catch (NameAlreadyBoundException e)
            {
            System.out.println("Entry " + base_gen + " already exists, no need to add
                ");
            annulation_insertion(ctx);
            System.exit(0);
            }
        catch (NamingException e)
            {
            System.err.println(e);
            annulation_insertion(ctx);
            System.exit(0);
            }
        }

    private static void confmodAjout(DirContext ctx,Node Child)
        {
        NodeList attributs = Child.getChildNodes();
        String dn="";
        try {
            javax.naming.directory.Attribute objClasses = new BasicAttribute("
                objectclass");
            javax.naming.directory.Attributes attr = new BasicAttributes();
            String tmp = attributs.item(0).getParentNode().getNodeName().
                toUpperCase().substring(0,1);
            tmp+=attributs.item(0).getParentNode().getNodeName().substring(1);
            objClasses.add("cfg"+tmp+"Entry");
            attr.put(objClasses);
            for(int j=1;j<attributs.getLength()-1;j++)
                {
                Node elt = attributs.item(j);
                if (!elt.getNodeName().equals("#text"))
                    {
                    if(elt.getNodeName().endsWith("Id"))
                        {dn="\""+elt.getNodeName()+"="+elt.getFirstChild().
                            getNodeValue()+","+base_gen.substring(1);}
                    attr.put(new BasicAttribute(elt.getNodeName(),elt.getFirstChild().
                        getNodeValue()));
                    }
```

```
            }
        ctx.createSubcontext(dn,attr);
        dn_vecteur.addElement(dn);
        System.out.println( "Added entry " + dn + ".");
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + dn + " already exists, no need to add");
        annulation_insertion(ctx);
        System.exit(0);
        }
    catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static void aicAjout(DirContext ctx,Node Child)
    {
    NodeList attributs = Child.getChildNodes();
    String dn="";
    try {
        javax.naming.directory.Attribute objClasses = new BasicAttribute("
            objectclass");
        javax.naming.directory.Attributes attr = new BasicAttributes();
        String tmp = attributs.item(0).getParentNode().getNodeName().
            toUpperCase().substring(0,1);
        tmp+=attributs.item(0).getParentNode().getNodeName().substring(1);
        objClasses.add(tmp);
        attr.put(objClasses);
        for(int j=1;j<attributs.getLength()-1;j++)
            {
            Node elt = attributs.item(j);
            if (! elt.getNodeName().equals("#text"))
                {
                if(elt.getNodeName().endsWith("Index"))
                    {dn="\""+elt.getNodeName()+"="+elt.getFirstChild().
                        getNodeValue()+","+base_gen.substring(1);}
                attr.put(new BasicAttribute(elt.getNodeName(),elt.getFirstChild().
                    getNodeValue()));
                }
```

```
            }
        ctx.createSubcontext(dn,attr);
        dn_vecteur.addElement(dn);
        System.out.println( "Added entry " + dn + ".");
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + dn + " already exists, no need to add");
        annulation_insertion(ctx);
        System.exit(0);
        }
    catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static void DomainAjout(DirContext ctx,NodeList cfgDomain)
    {
    String tmp="";
    try {
        for(int i=0;i<cfgDomain.getLength();i++)
            {
            Vector cfgAtt = new Vector();
            Vector cfgAtt_dn = new Vector();
            NodeList childs = cfgDomain.item(i).getChildNodes();
            javax.naming.directory.Attribute objClasse_domain = new
                BasicAttribute("objectclass");
            javax.naming.directory.Attributes attr = new BasicAttributes();
            objClasse_domain.add("cfgDomainEntry");
            attr.put(objClasse_domain);
            tmp = recherche_domainId(childs);
            for (int j=1;j<childs.getLength()-1;j++)
                {
                if (!childs.item(j).getNodeName().equals("#text"))
                    {
                    if (!childs.item(j).getNodeName().startsWith("cfgAtt"))
                        {attr.put(new BasicAttribute(childs.item(j).getNodeName
                            (),childs.item(j).getFirstChild().getNodeValue()));}
                    else
                        {
```

```
                        cfgAtt_dn.addElement(tmp+","+base_gen.substring(1));
                        cfgAtt.addElement(childs.item(j));
                        }
                    }
                }
            ctx.createSubcontext(tmp+","+base_gen.substring(1),attr);
            dn_vecteur.addElement(tmp+","+base_gen.substring(1));
            System.out.println( "Added entry " + tmp+","+base_gen.substring(1)
                + ".");
            for (int j=0;j<cfgAtt.size();j++)
                {cfgAttAjout(ctx,(String)cfgAtt_dn.elementAt(j),(Node)cfgAtt.
                    elementAt(j));}
            }
        }
    catch (NameAlreadyBoundException e)
        {
        System.out.println("Entry " + tmp+","+base_gen.substring(1) + " already
            exists, no need to add");
        annulation_insertion(ctx);
        System.exit(0);
        }
  catch (NamingException e)
        {
        System.err.println(e);
        annulation_insertion(ctx);
        System.exit(0);
        }
    }

private static String recherche_domainId(NodeList childs)
    {
    String ID="";
    for (int i=1;i<childs.getLength()-1;i++)
        if(childs.item(i).getNodeName().endsWith("Id"))
            {ID="\""+childs.item(i).getNodeName()+"="+childs.item(i).
                getFirstChild().getNodeValue();}
    return ID;
    }

private static void cfgAttAjout(DirContext ctx,String dn,Node cfgAtt)
    {
    String base="";
    try {
```

```
            NamedNodeMap attributs = cfgAtt.getAttributes();
            base="\""+attributs.getNamedItem(cfgAtt.getNodeName()).getNodeName
                ()+"="+attributs.getNamedItem(cfgAtt.getNodeName()).getNodeValue
                ()+","+dn.substring(1);
            javax.naming.directory.Attribute objClass = new BasicAttribute("
                objectclass");
            javax.naming.directory.Attributes attr = new BasicAttributes();
            objClass.add(cfgAtt.getNodeName()+"Entry");
            attr.put(objClass);
            for(int i=0;i<attributs.getLength();i++)
                {
                Node elt = attributs.item(i);
                if (! elt.getNodeName().equals("#text"))
                    {attr.put(new BasicAttribute(elt.getNodeName(),elt.getFirstChild()
                        .getNodeValue()));}
                }
            ctx.createSubcontext(base,attr);
            dn_vecteur.addElement(base);
            System.out.println( "Added entry " + base + ".");
            }
        catch (NameAlreadyBoundException e)
            {
            System.out.println("Entry " + base + " already exists, no need to add");
            annulation_insertion(ctx);
            System.exit(0);
            }
        catch (NamingException e)
            {
            System.err.println(e);
            annulation_insertion(ctx);
            System.exit(0);
            }
        }

    private static void annulation_insertion(DirContext ctx)
        {
        System.out.println("\nInsertions canceled ...\n");
        for(int i=dn_vecteur.size()-1;i>=0;i--)
            {
            try {
                System.out.println("Entry : "+dn_vecteur.elementAt(i)+" destroyed.");
                ctx.destroySubcontext((String)dn_vecteur.elementAt(i));
                }
```

```
        catch(NamingException e) {System.err.println(e);}
        }
    }

private static void usage()
    {
    System.out.println("Description : allow the exportation of an Open agent
        configuration to a LDAP directory service");
    System.out.println("Usage : \"java  ... \"  ldapExport <XMLfile_name> <
        configuration_name> <version>");
    }

public static void main(String[] argv)
    {
    try {
        Document doc = init_Parsing(argv[0]);
        configName=argv[1];
        version=argv[2];
        if (doc==null){System.out.println(argv[0]+" seems to be not well-formed,
            loading failed...");}
        else {
            Hashtable env = new Hashtable(5, 0.75f);
            env.put(Context.INITIAL_CONTEXT_FACTORY,"com.sun.jndi.ldap.
                LdapCtxFactory");
            env.put(Context.SECURITY_AUTHENTICATION,"simple");
            env.put(Context.SECURITY_PRINCIPAL,"cn=ldap_admin,o=
                RedEagle Corp.,c=be");
            env.put(Context.SECURITY_CREDENTIALS,"secret");
            env.put(Context.PROVIDER_URL,"ldap://redscorpion:389/o=
                RedEagle Corp.,c=be");
            String agentType=doc.getFirstChild().getNodeName();
            String schemaFile=doc.getFirstChild().getAttributes().getNamedItem
                ("xsi:noNamespaceSchemaLocation").getNodeValue();
            if (agentType.equals("OACconfig"))
                {
                System.out.println("Exportation of a Control open agent
                    configuration ;-)");
                insertion (env,"OAC","alixd",doc,agentType);
                System.out.println("LDAP directory service updated ;-)");
                }
            else  if (agentType.equals("OAXconfig"))
                {
```

```
                System.out.println("Exportation of an Extension open agent
                    configuration ;-)");
                insertion (env,"OAX","generix",doc,agentType);
                System.out.println("LDAP directory service updated ;-)");
                }
            else  if (agentType.equals("COACHconfig"))
                {
                System.out.println("Exportation of a COACH open agent
                    configuration ;-)");
                COACHinsertion(env,doc);
                System.out.println("LDAP directory service updated ;-)");
                }
            else
                {
                System.out.println("Unknown agent type...");
                }
            }
        }
    catch(ArrayIndexOutOfBoundsException e){usage();}
    }

private static class MyErrorHandler implements ErrorHandler {

    private PrintStream out;

    MyErrorHandler(PrintStream out) {
        this.out = out;
    }

    private String getParseExceptionInfo(SAXParseException spe) {
        String systemId = spe.getSystemId();
        if (systemId == null) {
            systemId = "null";
        }
        String info = "URI=" + systemId +
            " Line=" + spe.getLineNumber() +
            ": " + spe.getMessage();
        return info;
    }

    public void warning(SAXParseException spe) throws SAXException {
        out.println("Warning: " + getParseExceptionInfo(spe));
    }
```

```
        public void error(SAXParseException spe) throws SAXException {
            String message = "Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }

        public void fatalError(SAXParseException spe) throws SAXException {
            String message = "Fatal Error: " + getParseExceptionInfo(spe);
            throw new SAXException(message);
        }
    }

}
```

G Illustration d'une connexion entre une application Zope et un annuaire LDAP

Afin de démontrer les possiblités de connexion entre Zope et LDAP, voici un exemple élémentaire de projet Zope permettant d'afficher des configurations d'un agent stockées sur un annuaire. Il a été réalisé à partir de la librairie livrée en standard avec le package Zope-ldap.

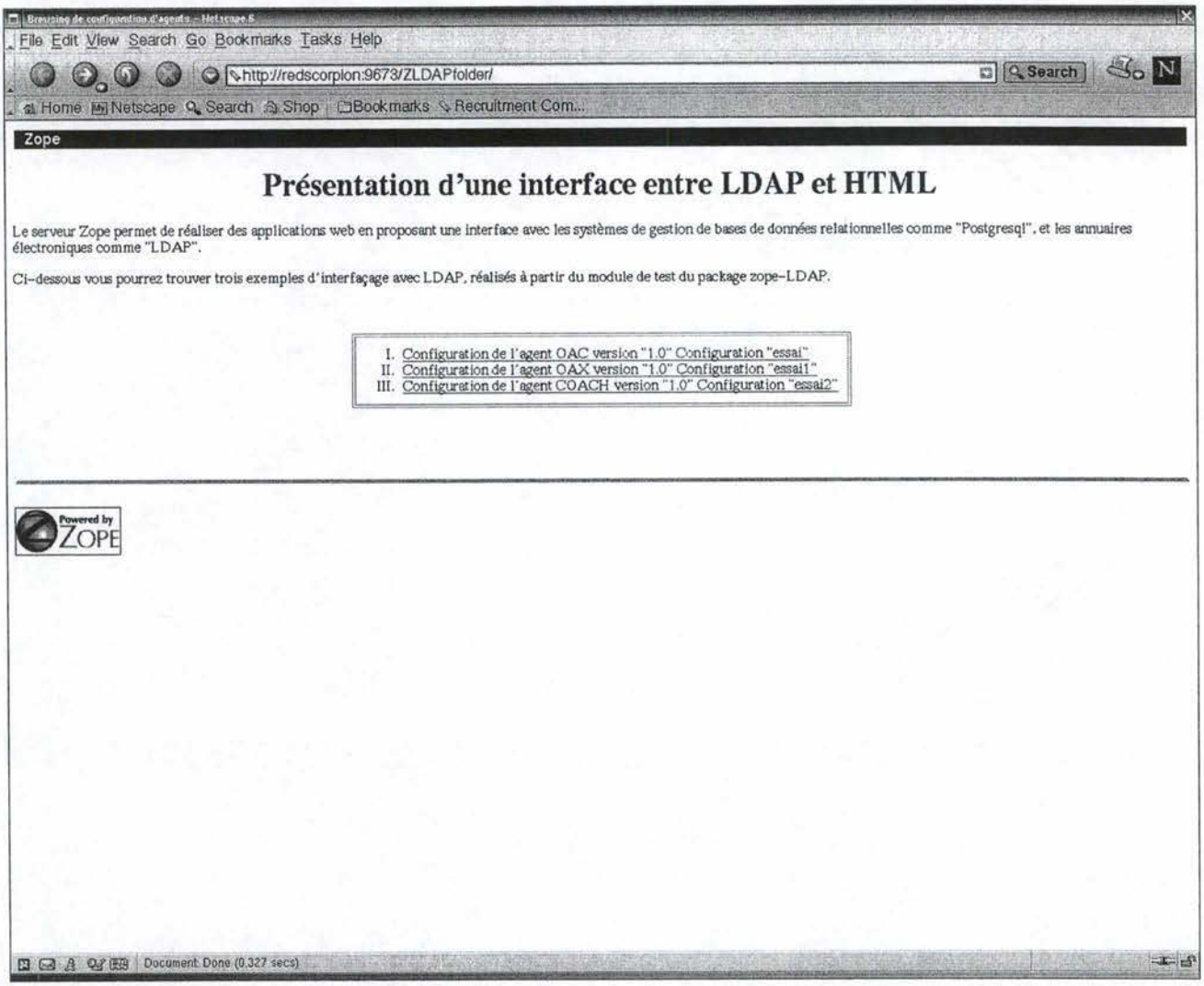

FIG. 22 – Exemple de page d'acceuil

Comme vous pouvez le voir, il suffit d'entrer une URL pour accéder à la page d'acceuil. A partir de cette page, vous avez la possibilité d'offrir tout une série de services. Ici, vous avez le choix d'afficher une configuration particulière de chacun des trois agents.

Si nous choisissons d'afficher la configuration de l'agent COACH en cliquant sur le dernier lien, nous obtenons la page suivante à l'écran :

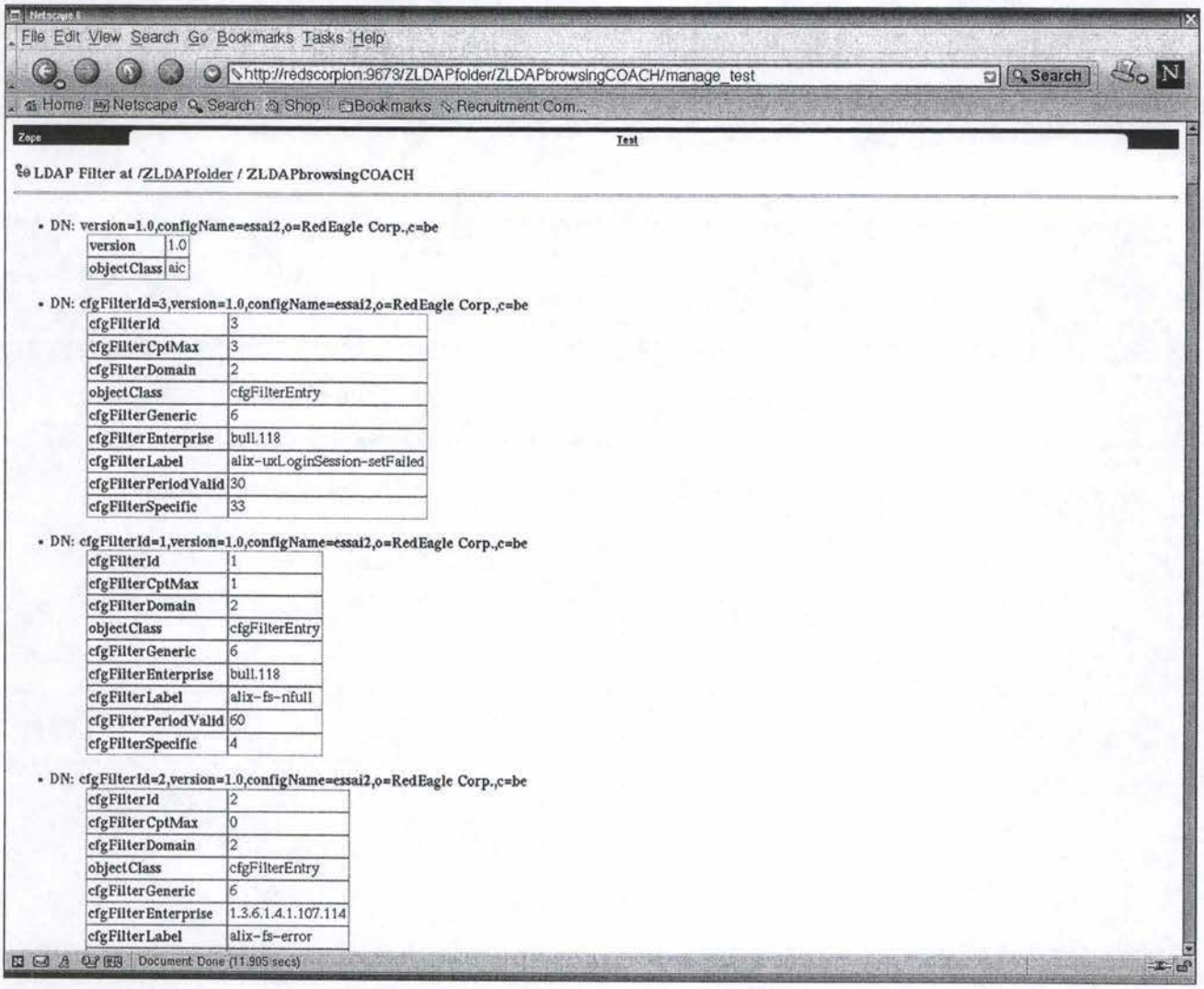

FIG. 23 – Exemple de résultat d'une requête

La configuration est alors renvoyée au navigateur et peut être consultée à partir de celui-ci.

G.1 Construction pas-à-pas de cet exemple

La première chose pour construire un petit exemple tel que celui qui vient d'être présenté consiste à créer un nouveau folder, dans lequel tous les fichiers relatifs au projet vont être stockés. Remarquons que le nom de ce folder va servir d'URL pour accéder à la page d'acceuil. Dans notre exemple le folder s'appelle "ZLDAPfolder" et l'URL est "http ://redscorpion :9673/ZLDAPfolder" ou redscorpion :9673 représente le nom du serveur où le service Zope a été activé, suivi du port TCP qui lui a été attribué.

La figure ci-dessous nous montre l'interface graphique de gestion dont nous avons parlé ci-dessus. La page d'acceuil est représentée par l'objet "index_html(Page d'acceuil)". Le

code de cette page n'est rien d'autre que du HTML classique. L'objet "acl_users(User Folder)" permet de définir les différents niveaux d'accès des utilisateurs aux éléments du projet(Owner ou Manager).

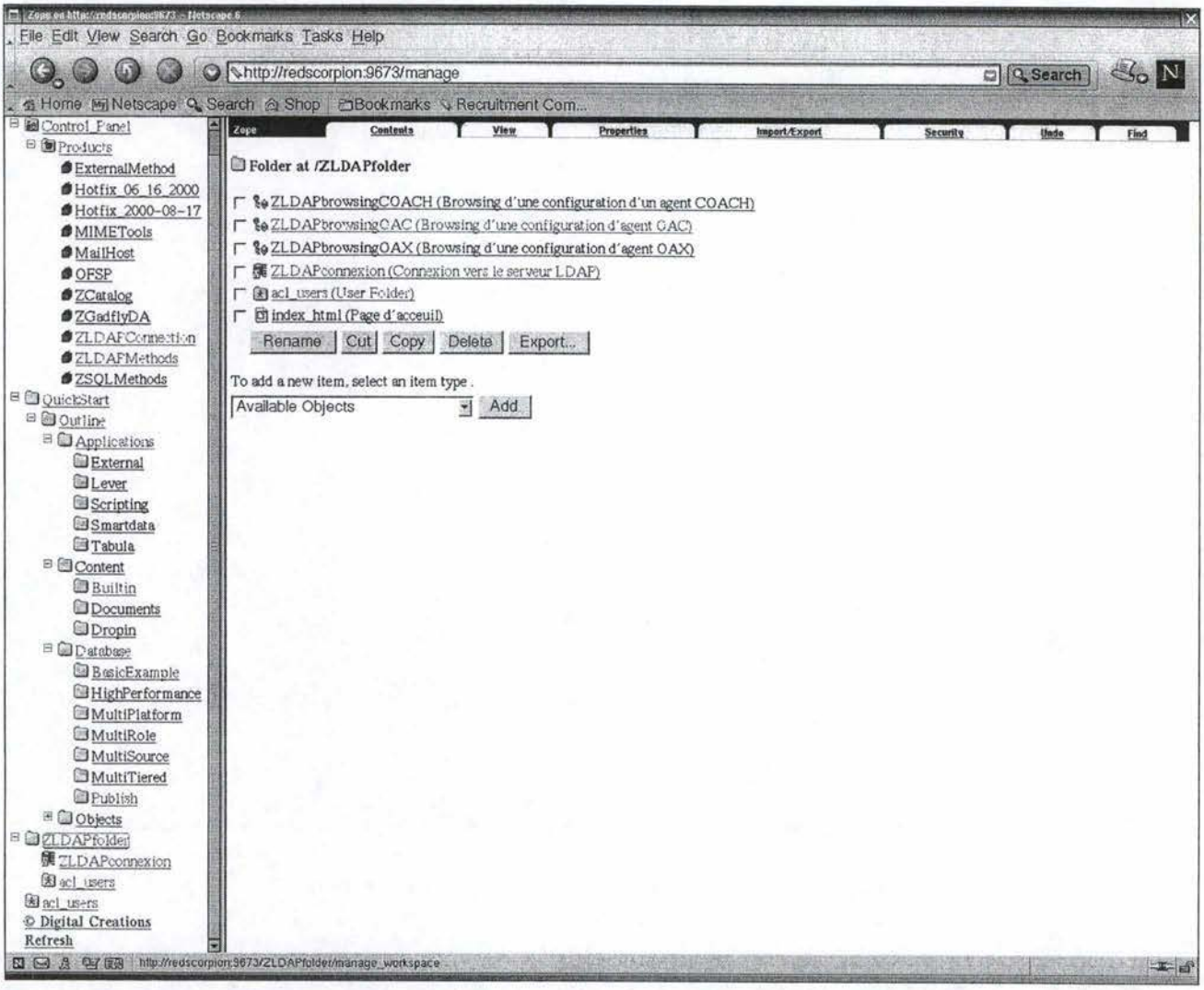

FIG. 24 – Interface de gestion d'un projet Zope

Avant toute utilisation d'une base de données, il faut créer un objet de connexion vers le serveur. Ici cette objet est représenté par "ZLDAPconnexion(Connexion vers le serveur LDAP)". Pour ajouter une connexion il suffit d'aller dans la combo "To add a new item, select an item type" et de selectionner "LDAP connection". Ensuite l'écran suivant apparaît et il suffit de complèter...

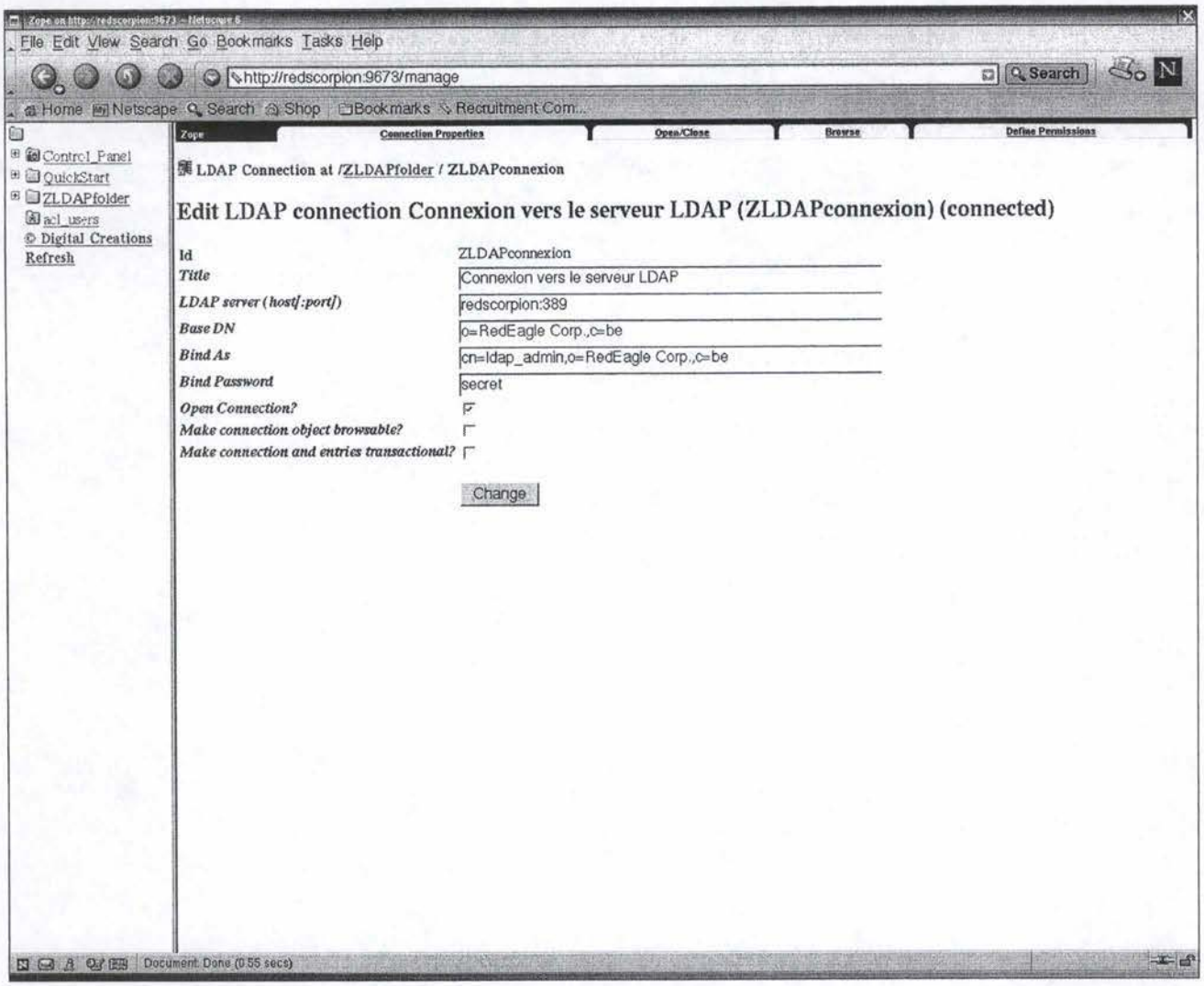

FIG. 25 – Interface de création d'une connexion

Une fois la connexion disponible, il vous est alors possible d'y ajouter des objets "LDAP Filter" afin de pouvoir définir vos critères de recherche. La figure suivante donne un exemple de filtre. Bien sûr, il vous est loisible d'y ajouter vos propres méthodes permettant de personnaliser vos traitements. Grâce à zope, vous pourrez ajouter, modifier ou supprimer des objets LDAP sans difficulté en vous servant d'un navigateur Internet comme support graphique. Ce qui signifie que vous pourrez faire de l'administration à distance, tenant compte du fait que la sécurité d'accès devra dès lors être minutieusement renforcée.

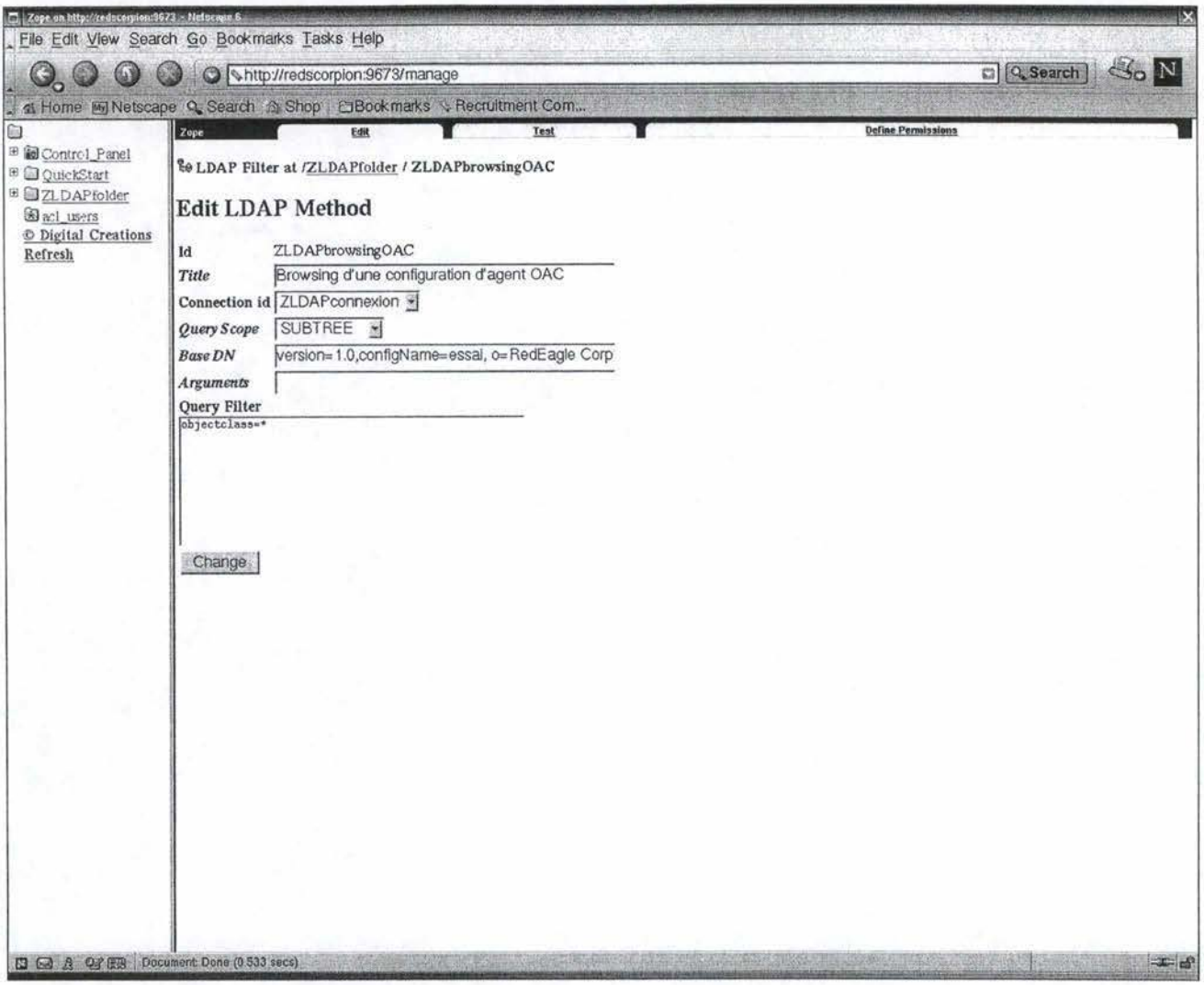

FIG. 26 – Interface de création d'une requête LDAP

Nous disposons maintenant d'une connexion vers un annuaire LDAP qui nous permet d'afficher trois configurations distinctes représentant chacune une configuration particulière pour l'agent de contrôle, l'agent d'extension et COACH.