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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : SYSTEME D'INFORMATION

Sujet :

Conception et mise en œuvre d'un format standard pour les fichiers logs

Présenté par : BARBARA Malek

AMARI Mohamed

Encadreur : Mohand Oussaid. L

Présidente des Jurys : M^me^ Mokhtari.
Membres des Jurys : M^me^ Sellali.
M^r^ Menacer.

Organisme d'accueil : CERIST

Soutenue le: 01^er^ Décembre 2005

- Promotion 2005-

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

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.

Mémoire pour l'obtention
d'un diplôme d'ingénieur d'état en informatique.
Option : SYSTEME D'INFORMATION

Sujet :

Conception et mise en œuvre d'un format standard pour les fichiers logs

Présenté par : BARBARA Malek

AMARI Mohamed

Encadreur : Mohand Oussaid. L

Président des jurys : M^me^ Mokhtari.
Membres des jurys : M^me^ Sellali.
M^r^ Menacer.

Organisme d'accueil : CERIST

Soutenue le: 01^er^ Décembre 2005

- Promotion 2005-

Remerciements

En premier lieu nous remercions ALLAH le tout puissant.

Nous remercions vivement notre encadreur M^lle^ MOHAND OUSSAID pour nous avoir proposé ce sujet, de nous avoir encadré et guidé pendant toute la période du stage.

Nous remercions chaleureusement notre promoteur M^r^ BOUKHELEF pour nous avoir guidé pendant toute la période du stage.

Nous adressons nos sincères remerciements aux membres de jury pour honorer notre travail en acceptant de le juger.

A tous les enseignants ayant contribué à notre formation, notre profonde gratitude.

Nous tenons également à adresser nos sincères remerciements à nos amis de la promotion.

Que toute personne ayant contribué de près ou de loin à l'aboutissement de ce travail trouve ici notre sincère reconnaissance.

TABLE DES MATIERES

TABLE DES MATIERES

Introduction Générale

Chapitre I : La Sécurité Informatique

Chapitre II : Les Fichiers Logs

Partie I : Les fichiers logs des Firewalls

Partie II : Les fichiers logs des Sites Web

Partie III : Les fichiers logs des SGBD

Chapitre III : Les Approches Possibles

Chapitre IV : La Norme XML

Chapitre V : La Conception

Chapitre VI : Implémentation et Mise en Oeuvre

Conclusion Générale et Perspective

LISTE DES FIGURES

Liste des Figures

LISTE DES TABLEAUX

Liste des Tableaux

INTRODUCTION GENERALE

Introduction Générale

Les informations contenues dans les fichiers logs des : systèmes d'exploitation, serveurs de bases de données, serveurs Web et firewalls sont précieuses. Mais leur format souvent peu pratique et non standard ne permet pas une exploitation facile. Tout au plus, on peut chercher à déboguer en direct pour diagnostiquer un problème.

En dehors du déboguage, l'administrateur consciencieux a besoin de statistiques sur les différents services qu'il exploite. En extrapolant ces données, il devient possible d'anticiper proactivement.

Les outils utilisés dans ce cadre sont plutôt axés sur le reporting de trafic. En outre, il existe d'autres outils d'analyse des fichiers logs qui servent à réagir en temps réel à des événements. En général, il s'agit de surveiller les messages du firewall installé et de réagir en un temps le plus bref possible.

Les fichiers logs étant des fichiers volumineux et ne répondant à aucun standard, ceci rend leur exploitation à l'état brut quasi-impossible.

Actuellement les analyseurs de fichiers logs sont dédiés à des produits spécifiques (serveur web, firewall,...).

L'objectif de notre travail est de proposer pour les fichiers logs un format standard qui fournit une présentation structurée des données et qui permet une exploitation de ces dernières pour des fins d'analyses et de prise de décisions.

Nous allons tout d'abord étudié les principaux types de fichiers logs générés par les systèmes les plus courants : Firewalls, Sites Web, SGBD et ce du point de vue : format, outils disponibles afin de pouvoir évaluer l'opportunité de les standardiser à travers le format que nous allons proposer.

Etant donné que les fichiers logs n'obéissent pas à un format permettant des manipulations intéressantes de leurs données, nous allons étudier la possibilité d'utiliser XML pour restructurer et faciliter l'exploitation des données contenues dans ces derniers. En effet, le format XML permet l'échange de données des systèmes d'information à une échelle globale, il renforce la communication du document électronique en séparant : contenu,

structure et présentation et offre de nouvelles voies pour décrire, structurer et présenter l'information contenue dans les documents de formats natifs à traiter.

Ce document décrit la démarche que nous avons adoptée pour l'étude de ces différents fichiers ainsi que l'architecture détaillée du système que nous proposons. Il s'articule sur les six chapitres suivants :

- **Chapitre I :** Dans ce chapitre nous présenterons la sécurité informatique de façon globale, on exposera quelques chiffres sur la sécurité, les principales vulnérabilités, les mesures de protection et leurs limites.

- **Chapitre II :** Dans ce chapitre nous présenterons les différents fichiers logs à étudier selon les trois catégories retenues. Nous en exposerons : la structure, l'importance, les outils d'analyse et leurs limites.

- **Chapitre III :** Dans ce chapitre nous étudierons les approches possibles pour la conception de notre système à savoir : les bases de données classiques et la norme XML (eXtensible Markup Language) avec les avantages et inconvénients qu'elles offrent afin de nous permettre de faire notre choix.

- **Chapitre IV :** Dans ce chapitre nous présenterons en détail la norme XML (eXtensible Markup Language), ainsi que l'outil Xquery qui permet l'interrogation des documents XML.

- **Chapitre V :** Dans ce chapitre nous présenterons l'approche de conception que nous proposons et qui est basée sur la norme XML.

- **Chapitre VI :** Ce chapitre est consacré à la description de l'implémentation du système.

CHAPITRE I

LA SECURITE INFORMATIQUE

Introduction

Le savoir est synonyme de pouvoir parce qu'il nous permet de prendre des décisions en toute connaissance de cause et non en fonction de ce que nous supposons être vrai.

L'informatique, étant une source perpétuelle de nouvelles technologies, elle engendre de nouveaux problèmes de sécurité. Les problèmes de sécurité concernent tout aussi bien les ordinateurs, les réseaux, la téléphonie. L'objectif de la sécurité des systèmes est de garantir qu'aucun préjudice ne puisse mettre en péril sa pérennité. Cela nous pousse à diminuer la probabilité de voir des menaces se concrétiser, à en limiter les atteintes ou dysfonctionnements induits, et autoriser le retour à un fonctionnement normal à des coûts et des délais acceptables en cas de sinistre. D'où l'importance d'étudier la sécurité d'un système en montrant quelles sont ses vulnérabilités (failles), les menaces, les attaques, et les mesures à prendre.

Dans ce qui va suivre nous aborderons les principaux concepts qui se rapportent à la sécurité des systèmes en mettant l'accent sur les vulnérabilités qu'il est possible d'exploiter et les mesures à mettre en place pour les contourner.

1-Qu'est ce que la sécurité ?

La sécurité sur un système consiste en général à s'assurer que celui qui modifie ou consulte des données du système en a l'autorisation et qu'il peut le faire correctement car le service est disponible. **[DES 00]**

1.1-Principales notions de la sécurité

- **La confidentialité :** caractère réservé d'une information dont l'accès est limité aux seules personnes admises à la connaître pour les besoins du service.
- **L'intégrité :** le système et l'information traitée ne sont modifiés que par une action volontaire et légitime. Lorsque l'information est échangée, l'intégrité s'étend à l'authentification du message, c'est-à-dire la garantie de son origine et de sa destination.
- **La disponibilité :** aptitude du système à remplir une fonction dans les conditions définies d'horaires, de délais, de performances. **[ARM 01]**

1.2-Quelques chiffres

Après un test effectué en l'an 2000 sur 12 000 hôtes du Département de la défense américaine, on retient que 1 à 3% des hôtes ont des ouvertures exploitables et que 88%

peuvent être pénétrés par les relations de confiance. Notons que seules 4% de ces attaques sont détectées et que 5% de ces 4% sont rapportées. Enfin, notons que le nombre de voleurs d'informations a augmenté de 250% en 5 ans, que 99% des grandes entreprises rapportent au moins un incident majeur et que les fraudes informatiques et de télécommunication ont totalisés 10 milliards de dollars pour seuls les Etats-Unis. Enfin, 1290 des plus grandes entreprises rapportent une intrusion dans leur réseau interne et 2/3 d'entre elles à cause de virus (cf. 3.5.2.2.k.). **[DES 00]**

2-La vulnérabilité

Une vulnérabilité représente une faiblesse ou une faille dans les procédures de sécurité, les contrôles administratifs ou les contrôles internes d'un système. **[ARM 01]**

2.1-Pourquoi les systèmes sont vulnérables ?

- La sécurité est coûteuse et difficile. Les organisations n'ont pas le budget nécessaire.
- La sécurité ne peut être sûre à 100%, elle est même souvent inefficace.
- La politique de sécurité est complexe et basée sur des jugements humains.
- Les organisations acceptent de courir le risque, la sécurité n'est pas une priorité.
- De nouvelles technologies (et donc vulnérabilités) émergent en permanence.
- Les systèmes de sécurité sont faits, gérés et configurés par des hommes. **[DES 00]**

La vulnérabilité peut être liée :

- à l'organisation :
- Absence de méthodologie.
- Absence de plan de secours.
- Absence de formation.
- Méconnaissance des règlements.
- aux systèmes d'information.
- Spécifications mal adaptées ou mal définies:ce qui est demandé ne correspond pas à ce que l'on souhaite.
- Difficulté de vérification de la complétude : le manque de lisibilité de l'application ne permet pas de s'assurer que tout ce qui a été demandé a été programmé.
- Définition de sécurité à posteriori (pas de prise en compte dans les spécifications).
- Erreurs de conception, divergence par rapport aux spécifications.

- insuffisance des tests qui ne permettront pas de s'assurer que le système fait essentiellement ce qu'il doit faire.
- Faiblesse de l'intégration, passage des étapes d'assemblage sans grande rigueur.
- Complexité mal maîtrisée : la lourdeur des programmes conjuguée à une approche mal cernée du besoin.
- Évolution du système, programmation mal structurée, non documentée engendre des difficultés de maintenance et d'évolutivité.

3-La menace et les attaques sur les systèmes d'information

3.1-La menace

La menace est une violation potentielle de la sécurité (accident, erreur, malveillance). D'une manière générale deux types de menace peuvent être distingués : la menace accidentelle et la menace intentionnelle. **[ARM 01]**

3.1.1– La menace accidentelle

Elle peut découler d'une catastrophe naturelle (incendie, inondation, tremblement de terre...), d'une erreur dans l'exploitation du système d'information (manipulation, saisie...) ou de pannes.

3.1.2– La menace intentionnelle

Elle est le fait d'un acte volontaire. Cette menace peut être interne (représente 70 à 80% des cas connus) ou externe. La **menace interne** est le fait d'utilisateurs ou d'administrateurs autorisés qui abusent de leurs privilèges ou les accroissent. La **menace externe** est le fait d'individus qui ne possèdent aucun accès légitime au système d'information et tentent d'en briser les barrières de sécurité.

3.2- les types de la menace

Suivant les informations traitées et les missions de l'organisme qui les traite, les menaces peuvent revêtir des aspects différents :

3.2.1– la menace ludique : les nouvelles techniques de traitement de l'information ont créé cette menace qui dans l'esprit de ceux qui en sont les auteurs relève plus du jeu et des loisirs que d'un forfait. Motivés par la prouesse technique, les auteurs cherchent davantage à démontrer la fragilité des systèmes plutôt que la nuisance ; les moyens sont faibles et les actes isolés.

3.2.2- la menace cupide : elle consiste en la recherche d'un gain financier important et rapide par des individus ou des groupes sans considération morale ; ses victimes se choisissent parmi ceux qui détiennent l'argent (banques, compagnies d'assurance).

3.2.3- la menace terroriste : un groupuscule, un groupe, voire un Etat, veulent frapper l'opinion par une action la plus spectaculaire possible, amplifiée par les médias, afin de déclencher une psychose de peur. Ce moyen d'action peut viser par exemple le sabotage de systèmes vitaux.

3.2.4- la menace stratégique : un Etat peut prendre connaissance d'informations classifiées de défense en accédant frauduleusement à des banques de données classifiées. Au-delà, on peut aussi envisager l'attaque massive de tous les systèmes vitaux d'un pays pour le neutraliser, le paralyser et le forcer à négocier.

Les menaces intentionnelles se concrétisent le plus souvent par des actions d'espionnage, de perturbation, de vol, de fraude physique, de chantage, de sabotage, d'accès illégitime, d'altération.

3.3- Le risque

Le risque peut se caractériser comme étant la probabilité qu'une menace particulière puisse exploiter une vulnérabilité donnée du système.

3.4. La typologie de l'agresseur

Il n'y a pas de portrait robot établi, toutefois quelques études montrent qu'il s'agit dans la grande majorité, d'hommes qui n'ont pas un travail gratifiant mais qui détiennent d'importantes responsabilités et l'accès à des informations sensibles. L'avidité, l'appât du gain, les problèmes personnels influent sur le comportement social.

3.4.1- le pirate :

- *Hacker* : individu curieux qui cherche à se faire plaisir, possède souvent un code d'honneur et de conduite, jeune, compétent il est patient et tenace.
- *Cracker* : plus dangereux, il cherche à nuire. Mal dans sa peau, il veut se venger de la société ou d'individus ; veut prouver sa supériorité et fait parti de clubs où il peut échanger des informations.

3.4.2- le fraudeur :

- *Interne* : techniquement compétent, informaticien, casier judiciaire vierge, il pense qu'il est sous-estimé, veut se venger de son employeur ; ses moyens sont ceux de l'entreprise.

• *Externe* : bénéficie de complicité volontaire ou non. Son but est de gagner de l'argent, son profil est proche de celui du malfaiteur ; il est parfois lié au grand banditisme.

3.4.3- l'espion :

• *D'Etat* : dispose de moyens importants ; patient, motivé, il exploite les vulnérabilités, difficiles à déceler, ne fait pas état de sa réussite.

• *Privé* : dispose de moins de moyens, mais il est doté des mêmes qualités ; il s'agit souvent d'un espion d'état reconverti.

3.5- L'attaque

Une attaque est une action malveillante consistant à tenter de contourner les mesures de sécurité d'un système d'information. **[ARM 01]**

3.5.1-Méthodes utilisées pour les attaques :

• Utiliser la négligence interne des utilisateurs vis à vis des droits et autorisations d'accès.

• Se faire passer pour un ingénieur pour obtenir des infos comme le mot de passe.

• Retrouver le mot de passe en utilisant des algorithmes systématiques.

• Casser les clefs de cryptographie lorsqu'elles sont trop courtes.

• se mettre à l'écoute sur le réseau et obtenir des informations.

• changer son adresse IP et passer pour quelqu'un de confiance. (IP spoofing)

• Injecter du code dans la cible comme des virus (cf. 3.5.2.2.k) ou un cheval de Troie (cf. 3.5.2.2.i)

• Exploiter les faiblesses des systèmes d'exploitation, des protocoles ou des applications. **[DES 00]**

3.5.2-Types d'attaques :

Les attaques peuvent porter sur les communications, les machines, les traitements, les personnels et l'environnement. Elles peuvent revêtir plusieurs formats sachant qu'elles sont à classer en deux catégories attaques physiques et logiques. **[ARM 01]** (Voir la fig-01)

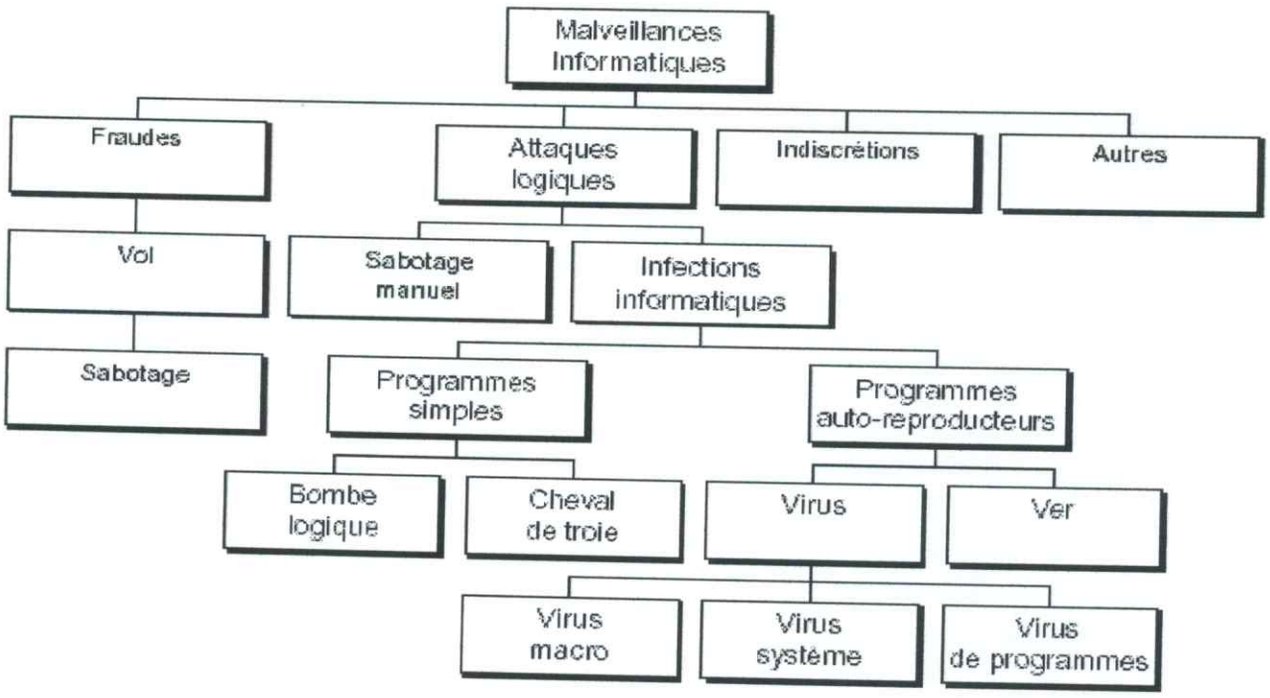


Fig-01 : Les différentes formes de malveillances informatiques [ARM 01]

3.5.2.1. Les attaques physiques :

a. Interception : récupération d'un signal électromagnétique. Après interprétation, il est possible d'en déduire des informations.

b. Brouillage : utilisé en télécommunication, il rend inopérant le système d'information.

c. Ecoute : technique traditionnelle qui consiste à se placer sur un réseau, à analyser et sauvegarder les informations qui y transitent.

d. Balayage : envoi, au système d'information, d'un panel d'informations pour déterminer celles qui suscitent une réponse.

e. Piégeage : à la conception, à la maintenance, consiste à introduire des instruction dans le but d'attaquer le système. **[ARM 01]**

3.5.2.2. Les attaques logiques :

a. Fouille : étude des fichiers et des variables du système d'information pour en extraire les informations importantes.

b. Canal caché : de très haut niveau cette attaque permet la fuite d'information en violant la politique de sécurité.

c. Déguisement : accès aux privilèges et aux droits d'un utilisateur par usurpation d'identité.

d. Mystification : l'attaquant va simuler le comportement d'une machine pour tromper un utilisateur.

e. Rejeu : envoi d'une séquence de connexion qui a été enregistrée à l'insu d'un utilisateur légitime.

f. Saturation : atteint la disponibilité par remplissage de zones de stockage ou par encombrement d'un canal de communication.

g. Trappe : point d'entrée dans une application à l'usage du programmeur, elle n'est pas retirée lors de la commercialisation.

h. La bombe logique : Une bombe logique est un programme contenant une fonction malveillante donc illicite associée à un déclenchement différé. Elle se caractérise par l'unicité de la cible et l'absence d'autoreproductions.

i. Le cheval de Troie (ou troyen) : Un cheval de Troie est un programme en apparence inoffensif contenant une fonction illicite cachée. Bien qu'il ne s'auto reproduise pas, il peut toucher un grand nombre de systèmes par les circuits de diffusion des logiciels. Un cheval de Troie est utilisé pour pénétrer par effraction dans l'ordinateur afin de modifier détruire ou consulter des informations.

j. Le ver : Un ver est une procédure parasite qui consomme les ressources du système (mémoire, réseau...). Doté de la faculté de se déplacer et de se reproduire au sein des mémoires et à travers les réseaux, il peut déclencher des actions malveillantes. Il est auto-reproductible.

k. Le virus : Un virus est un programme qui à la faculté de se dupliquer au sein d'autres programme ou sur des zones systèmes. Il est auto-reproductible.

Actuellement la plupart des virus possèdent une fonction de déclenchement différé.

Par ailleurs, certains virus marquent d'une empreinte les programmes qu'ils ont contaminés, afin de ne pas infecter plusieurs fois les mêmes cibles.

l. Le virus macro (virus de macro ou macro-virus): Il s'agit hélas d'un type d'infection très courant diffusé essentiellement par les applications bureautiques. Il se répand lorsque les utilisateurs se transmettent des documents infectés sans le savoir, soit par disquette ou autre support magnétique (cd, disque magnéto-optique,...), soit via un réseau local interne (intranet) (souvent sous la forme de pièces jointes à un courrier électronique) ou encore via un réseau de courrier électronique externe (Internet). Il est auto-reproductible. **[ARM 01]**

4-Infections informatiques

Les infections informatiques **[ARM 01]** se distinguent de la plupart des autres risques pour la sécurité des systèmes d'informations du fait que les formes d'attaques sont multiples, changent de forme en permanence en fonction :

- De l'évolution des systèmes d'exploitation.
- De la progression des réseaux (réseaux locaux, intranet, extranet et Internet).
- De l'évolution des habitudes des utilisateurs (confiance accrue dans les matériels et les logiciels).
- Du fait que les sauvegardes régulières n'offrent pas toujours les précautions suffisantes.

Qu'est-ce qu'une infection informatique ? Exprimé de façon simple, il s'agit d'un programme qui a le potentiel de se reproduire tout seul ou non. On parle alors de programme auto-reproducteur ou programme simple. Dans le cas d'un programme auto-reproducteur et lorsqu'il est exécuté, il effectue tout simplement une ou plusieurs copies de lui-même. Ces copies pourront être exécutées ultérieurement pour, elles-mêmes, donner lieu à de nouvelles copies, théoriquement à l'infini.

Les infections sont souvent générées par des pirates.

4.1- Les fonctions particulières des infections

Comme dans toute application fonctionnant dans un micro-ordinateur, les infections informatiques sont caractérisées principalement par les cinq fonctions suivantes :

- **Fonction illicite** : il s'agit de la fonction d'un programme qui n'est pas autorisée (donc non déclarée par le développeur), non documentée et qui ne concourt pas aux objectifs du programme.
- **Fonction cachée** : il s'agit de la fonction d'un programme qui n'est pas documentée et qui va fonctionner à l'insu de l'utilisateur.
- **Fonction de déplacement** : il s'agit de la fonction d'un programme qui est capable de transférer un programme en cours d'exécution.
- **Fonction de déclenchement différé** : il s'agit de la fonction d'un programme qui est en attente d'une condition pour s'exécuter.
- **Fonction d'auto-reproduction** : il s'agit de la fonction d'un programme qui est capable de créer une réplique identique d'elle-même ou du programme, le principal but à atteindre étant la saturation du disque dur.

5 - Les mesures anti-infections

Il n'existe pas de mesures anti-infections miracles. Toutefois l'utilisation judicieuse de mesures simples et de campagnes de sensibilisation peut réduire les risques à un niveau acceptable. Ces mesures se répartissent en quatres catégories. **[ARM 01]**

5.1 - Les mesures de prévention

Leur but est de retarder l'intrusion afin de réduire la probabilité d'infection. Des exemples de ces mesures préventives sont :

- Ne pas utiliser de programmes à la provenance douteuse.
- Contrôler l'accès aux équipements informatiques.
- Mettre en place des procédures d'audit informatique.
- Mettre en place des procédures d'intervention sur infection déclarée.
- Mettre en place une procédure de contrôle de la qualité des logiciels afin de tester tous les produits utilisés.
- Sensibiliser les utilisateurs de l'informatique aux risques liés aux infections informatiques.

5.2 - Les mesures de protection

Elle consiste à limiter les conséquences d'une infection entre l'instant de l'intrusion et celui de la détection par exemples :

- Cryptographie : pour la confidentialité des informations et la signature électronique.
- Logiciels anti-virus sachant que 2/3 des attaques sont des virus.
- Programmes de tests de vulnérabilité et d'erreurs de configuration.
- Utilisation des supports de données amovibles.
- Des sauvegardes régulières.

5.3 - Les mesures de détection

Ces mesures nous permettent d'atténuer les effets d'une infection en réduisant le délai qui s'écoule entre l'instant de l'infection et de la détection. **[DES 00]**

Généralement, les infections se manifestent à travers un ensemble de symptômes dont :

- Les problèmes de performances.
- Les erreurs.
- Les problèmes de mémoire de masse.

• Les problèmes de mémoire vive.

Les principales mesures de détection qui peuvent être envisagées sont :

• **Les Firewall ou pare-feu [MAR 04]** (c'est essentiellement un dispositif matériel ou logiciel de protection qui constitue une barrière entre un réseau local et un autre réseau non sûr tel que l'Internet ou un autre réseau local):pour l processus de filtrage des trames transitant du réseau externe vers le réseau interne.

• Audit **[MCC01]** : études des fichiers de log pour repérer des anomalies.

•Les systèmes de détection d'intrusion **[DES 00]** : il s'agit d'une surveillance permanente ou régulière des systèmes mise en place. Ils ont pour but d'analyser tout ou une partie des actions effectuées sur le système afin de détecter d'éventuelles anomalies de fonctionnement, des comportements anormaux d'un utilisateur ou des attaques connues.

5.4 - Les mesures d'élimination

Elles ont pour but de limiter la portée d'une infection en réduisant le délai entre détection et expulsion.

Les principales mesures possibles sont :

•La limitation des échanges de données.

•L'élimination des virus : à l'aide des logiciels anti-virus ou en réinitialisant le système : reformater, recharge de système, des programmes, et des données.

Conclusion

En matière de sécurité des systèmes, le risque zéro n'existe pas et tous les systèmes sont exposés. En effet, Il est impossible de garantir la sécurité totale d'un système car Il existe une foule de moyens que les attaquants peuvent utiliser pour profaner le système .Cependant la gestion efficace de la sécurité est une démarche complexe car elle nécessite du personnel bien formé et vigilant, des moyens techniques souvent coûteux, un responsable de la sécurité qualifié et du temps. .D'où la nécessite d'établir les mesures suffisantes pour pallier aux risques encourus en faisant un compromis entre coût, temps et efficacité.

D'autre part, il est très important de souligner que pour qu'elle soit efficace, une politique de sécurité doit aborder le système dans sa globalité sans négliger un aspect au détriment d'un autre, faisant de cette dernière un choix dont les maillons sont d'importance équivalente.

CHAPITRE II

LES FICHIERS LOGS

Introduction

D'après ce qu'on a vu dans le chapitre précédent en ce qui concerne l'importance d'assurer la sécurité de tout système, il est important d'établir des journaux de sécurité (fichiers logs) efficaces et complets et de procéder à des vérifications fréquentes et systématiques de ceux-ci. Souvent, cette fonction importante est négligée et considérée comme un fardeau administratif, mais en réalité, elle constitue un outil efficace, proactif et réactif mis à la disposition des administrateurs. Sur le plan proactif, les utilisateurs qui savent que leurs actions sont surveillées risquent moins de tenter de miner ou de profaner la sécurité d'un système. Sur le plan réactif, les contrôles périodiques des journaux pourraient permettre de déceler un manquement à la sécurité qui, autrement, aurait pu passer inaperçu, comme l'accès non autorisé à des comptes à accès limité ou la suppression non autorisée d'informations précieuses.

Les journaux de sécurité par exemple ne sont utiles que si le système est en mesure d'identifier chaque utilisateur avec précision et si ces journaux peuvent être protégés contre la falsification. D'ou la nécessité et l'importance d'étudier les fichiers logs.

A) Les fichiers logs (log file)

1. L'audit de la sécurité

L'audit de sécurité permet d'enregistrer tout ou une partie des actions effectuées sur le système. L'analyse de ces informations permet de détecter d'éventuelles intrusions. Les systèmes d'exploitation disposent généralement de systèmes d'audit intégrés, certaines applications aussi. Les différents évènements du systèmes sont enregistrés dans un journal d'audit qui devra être analysé fréquemment, voire en permanence. Sur les réseaux, il est indispensable de disposer d'une base de temps commune pour estampiller les évènements. **[DES 00]**

2. Définitions des fichiers logs

a- Ce sont des fichiers ayant pour extension « .log », regroupant l'ensemble des événements survenus sur un logiciel, une application, un serveur, un SGBD, un firewall, ou tout autre système informatique. **[DIC 05].**

b- Un log file (en français, journal de bord des connexions ou encore historique des requêtes adressées à un serveur ou encore fichier de " journalisation ", fichier journal) est un fichier

créé par un logiciel spécifique qui permet de garder les traces des opérations qu'il effectue. Ce logiciel est conçu pour enregistrer (c.-à-d., pour consigner dans le journal de bord). **[AES 04]**

c- Fichier texte où est enregistré l'historique des communications entre un serveur et des postes clients. On retrouvera en particulier les requêtes demandées au serveur, les messages d'erreurs générés par l'application, et donc des informations indispensables à analyser en cas d'erreur.

[JOU 05]

d- Toutes les activités du système sont enregistrées dans un fichier de consultations, appelé "fichier log". Les informations de ce fichier reflètent la vie du système. Un fichier log est sous format texte, il est constitué d'une suite de lignes de code représentant les différents évènements survenus sur le système. **[ADC05]**

3. Les données contenues dans les fichiers logs

Les types d'informations qu'on retrouve dans les fichiers logs sont:

Les informations sur les accès au système (qui y a accédé, quand et comment), les informations sur l'usage fait du système (utilisation du processeur, de la mémoire ou des entrées/sorties) et les informations sur l'usage fait des fichiers. Il doit également permettre d'obtenir des informations relatives à chaque application (le lancement ou l'arrêt des différents modules, les variables d'entrée et de sortie et les différentes commandes exécutées), ainsi que les informations sur les violations éventuelles de la sécurité (tentatives de commandes non autorisées) ainsi que les informations statistiques sur le système.

Notons que ces nombreuses informations occupent beaucoup de place et sont très longues à analyser. Ces informations doivent être, au moins pour un temps, stockées quelque part avant d'être analysées. **[DES 00].**

B) Types des fichiers logs

Dans ce qui suit, nous allons classer les fichiers logs d'après les systèmes qui les génèrent. Ainsi, nous retiendrons :

1) les fichiers logs des firewalls.

2) les fichiers logs des Sites Web.

3) les fichiers logs des SGBD.

PARTIE I

LES FICHIERS LOGS DES FIREWALLS

1. Les fichiers logs des Firewalls

1.1. Définition du firewall

Un firewall, aussi appelé pare-feu ou garde-barrière, est un programme, ou un matériel, chargé de protéger du monde extérieur et de certains programmes malveillants placés sur ordinateurs.

Placé entre un utilisateur et Internet, le firewall contrôle tout ce qui se passe, et surtout tout ce qui ne doit pas passer de l'un vers l'autre. Un firewall est aussi un système ou un groupe de système qui gère les contrôles d'accès entre deux réseaux.

Deux mécanismes sont utilisés : le premier consiste à interdire le trafic, et le deuxième à l'autoriser.

Certains firewalls mettent beaucoup d'énergie à empêcher quiconque de passer alors que d'autres tendent à tout laisser passer. **[DES 00]**

1.2. De quoi protège un firewall?

Certains firewalls laissent uniquement passer le courrier électronique. De cette manière, ils interdisent toute autre attaque qu'une attaque basée sur le service de courrier. D'autres firewalls, moins strictes, bloquent uniquement les services reconnus comme étant des services dangereux par exemple le service **Systat [ISS 96]** d'Unix qui permet d'obtenir des informations du système telles que les noms d'utilisateurs que les attaquants peuvent utiliser pour pénétrer le système en déviant les mots de passes.

Généralement, les firewalls sont configurés pour protéger contre les accès non authentifiés du réseau externe.

Ceci, empêche des utilisateurs malhonnêtes de se logger sur les machines du réseau interne, mais autorise les utilisateurs à communiquer librement avec l'extérieur.

Les firewalls sont également intéressants dans le sens où ils constituent un point unique où l'audit et la sécurité peuvent être imposés. Tous les échanges passent par les firewalls. Donc ils peuvent donner des résumés et des statistiques sur le trafic, ou encore toutes les connexions entre les deux réseaux. **[DES 00]**

1.3. Description des fichiers logs du firewall

Un fichier log du firewall reprend de façon chronologique, l'ensemble des événements qui ont affecté un système informatique et l'ensemble des actions qui ont résulté de ces événements. Le fichier log du Firewall regroupe les tentatives de communication, les arrêts systèmes et les installations de firewalls.

Ainsi que les informations événementielles qui sont : la date et l'heure de la tentative d'accès survenue sur le système, la machine d'origine, le système d'exploitation et le navigateur utilisé sur cet accès en indiquant son adresse IP, les services appelés, les actions entreprises et les types d'alertes enregistrés.**[DIC 05]**

1.4. Analyse des fichiers logs du firewall

Il existe deux méthodes d'analyses des fichiers logs du firewall : une première manuelle, et une deuxième automatique (par logiciel).

a- L'étude manuelle des fichiers logs

Elle repose essentiellement sur la comparaison des fichiers logs édités et de la compétence de l'analyseur lui même. Cette méthode n'est pas très efficace du fait de la taille de la présentation de ces données et du temps que peut prendre une telle analyse.

b- Les logiciels d'analyse des fichiers logs

Comme nous l'avons vu, les fichiers se présentent sous la forme de lignes de codes laissées sur les firewalls. Pour bénéficier d'une présentation plus exploitable de ces fichiers, certains logiciels ont été conçus pour faire une synthèse graphique des données. Ces logiciels sont installés avec le firewall ou indépendamment sur la machine.

Les exemples de ces logiciels sont : **Webtrends**, **Analog, Webspy Analyzer,** qui permettent une analyse des fichiers logs et même une représentation graphique du trafic.

Dans ce qui va suivre nous allons exposer un analyseur de fichiers logs des Firewalls qui est le **WebSpy Analyser** et d'essayer de comprendre comment il fonctionne.

1.5. Etude d'un Analyseur de fichiers logs du firewall : Webspy Analyzer

Webspy Analyzer Standard version 4.1 qui s'installe sur les systèmes d'exploitation : Windows, OS/2, Unix, est le dernier-né des produits d'analyse et de reporting Internet et Email de WebSpy. Il permet d'analyser les fichiers log des firewalls et des Proxy afin de connaître comment est utilisée la bande passante, et aide à réduire les menaces de sécurité, à respecter la charte d'utilisation d'Internet et d'améliorer la productivité au sein de l'organisation. WebSpy Analyzer Standard renseigne sur le trafic Web. WebSpy supporte plus de 60 formats de fichiers log, et en ajoute continuellement de nouveaux. **[SPY 04]**

1.5.1. Principales Fonctions et propriétés de Webspy

· Permet une interrogation détaillée des fichiers logs.

· Comprend l'analyse du trafic Internet.

· Génère des rapports personnalisés dans de multiples formats.

· Permet une organisation logique des données utilisant des alias et des profils.

· Programme des tâches automatiques pour l'analyse des données.

On peut aussi citer les propriétés suivantes :

· Utilise les ressources existantes comme les Groupes Utilisateurs de WINDOWS NT®.

· S'installe et se paramètre facilement.

· Possède une interface intuitive analogue aux interfaces du Web. **[SPY 04]**

1.5.2. Importation des fichiers Log

Pour commencer avec *Analyzer Standard*, on doit importer les **fichiers log**.

Lorsque *Analyzer Standard* importe des **fichiers log**, qui sont stockés dans un **format compressé** appelé Espace de Stockage ('Storage'). L'architecture d'un Espace de Stockage permet le traitement efficace de volumes importants de données.

On peut importer les **fichiers log** à partir d'un dossier ou d'un site FTP, et aussi dans un Espace de Stockage existant ou dans un nouveau Espace de Stockage.

Une fois les **fichiers log** importés, on peut créer des rapports, et utiliser les fonctions de *Analyzer Standard* tels les Alias, les Profils ou les Tâches automatiques, et accéder à des analyses pertinentes et détaillées. **[SPY 04]**

1.5.3. Analyse des données par Webspy

Une fois les données importées dans les Espaces de Stockage, on peut utiliser les Sommaires ('Summaries') pour visualiser les données. *WebSpy Analyzer Standard* permet de circuler dans toutes les données et de voir les résultats graphiquement. Pour afficher les données des Espaces de Stockage, on doit exécuter l'Analysis Wizard (l'assistant d'analyse), qui va aider dans le choix des données qu'on veut analyser.

Une fois l'analyse de *Analyzer Standard* terminée, l'écran de vue d'ensemble apparaît. (Voir la Figure 2) **[SPY 04]**

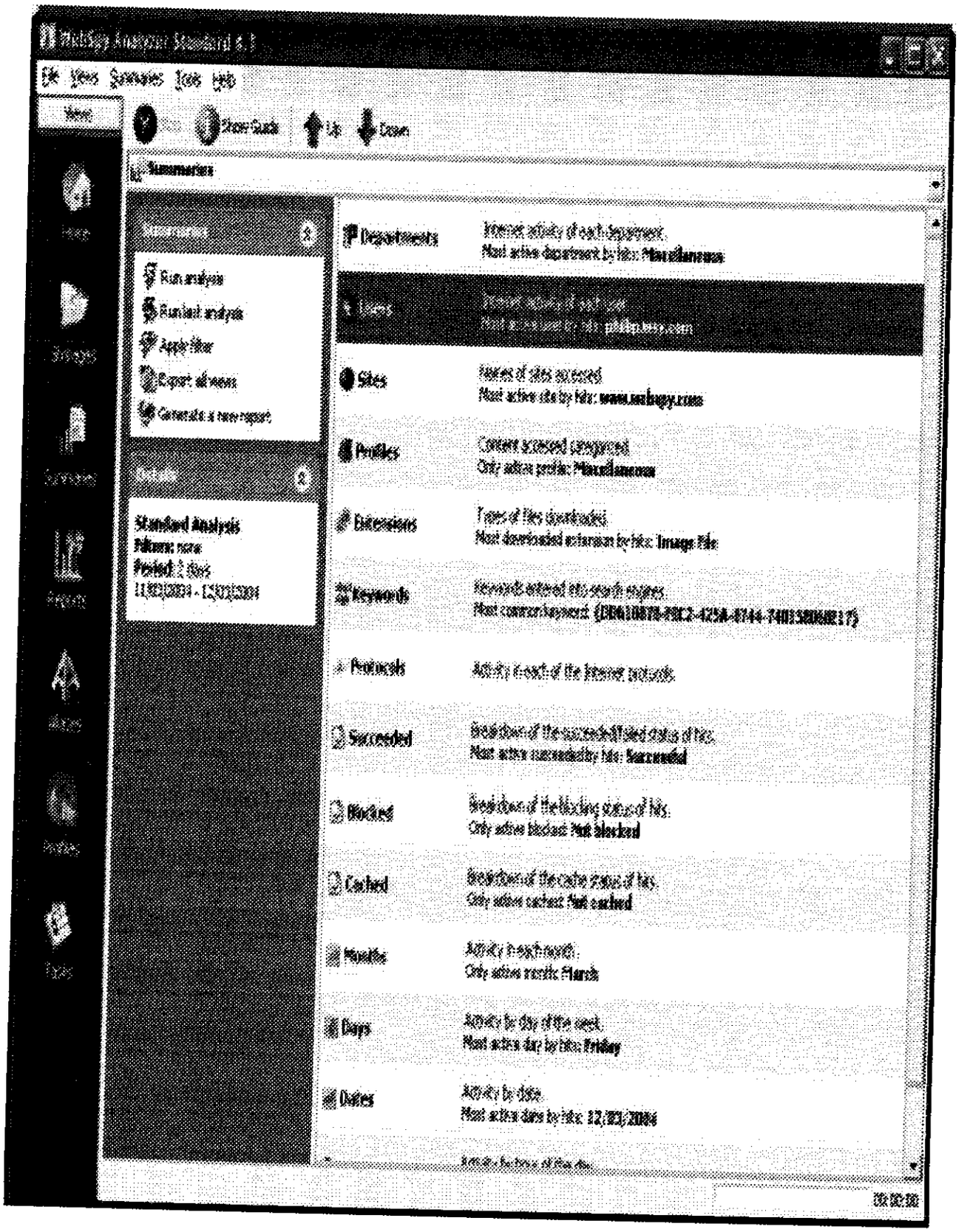


Fig-02: L'écran de vue d'ensemble des Sommaires [SPY 04]

1.6. L'importance des fichiers logs

Les fichiers logs permettent donc de recueillir des données, plus précisément, l'analyse des données contenues dans les log files permet, par exemple :

- De connaître les types d'alertes enregistrées.
- De connaître les tentatives de communications au firewall.
- De savoir en temps réel l'heure et la date de chaque événement produit et permettre d'établir un bilan sur les dates et heures les plus fréquentes d'attaques.
- De compter le nombre de visites.
- De connaître les sources et destination des information circulantes.
- De plus, quelles modifications faut-il apporter au firewall et quelles sont les règles qu'il doit appliquer après ces modifications.

Ces résultats, peuvent inciter les concepteurs à effectuer des modifications sur le firewall dans le but d'améliorer la sécurité, faciliter la navigation et d'optimiser la structure. Toutefois les log files sont peu utiles pour guider les modifications.

1.7. Limites des fichiers logs du firewall

Bien que les données fournies par les fichiers logs puissent être utiles pour l'évaluation ergonomique d'un firewall, notamment pour mesurer l'impact des modifications réalisées, il est important de prendre en compte les limites inhérentes à ces données lors de leur analyse et de leur interprétation.

Rappelons que les fichiers logs contiennent des informations sur l'activité du firewall relative à une communication entre les deux réseaux. Il faut donc être prudent quant aux interprétations que l'on peut être amené à faire sur l'activité ou l'intérêt des visiteurs.

On note à ce propos un certain nombre de limitations : Les firewalls et leurs protections d'accès à un réseau masquent l'adresse IP des utilisateurs. Toute demande de connexion provenant d'un serveur doté d'une telle protection aura la même adresse et ce, quel que soit l'utilisateur. Il est donc impossible, dans ce cas, d'identifier et de distinguer les visiteurs provenant de ce réseau. Alors, en tenant compte des limites de cet outil, l'évaluation ergonomique d'un firewall ne peut reposer sur les seules données issues de ces fichiers. L'évaluation doit faire appel à d'autres techniques. Pour obtenir des informations sur les comportements des visiteurs face à un firewall donné, rien ne remplacera l'observation directe et les tests utilisateurs. La technique des fichiers logs comme toute technique d'enregistrement automatique d'interactions est utile. Toutefois, si l'objectif est d'obtenir des données sur

l'activité réelle des visiteurs, alors l'enregistrement devrait se faire sur le poste de ce dernier. D'autres problèmes s'ajoutent à ces limites parmi eux, l'expiration du fichier log lui-même (connaître quand est-ce qu'il faut supprimer le fichier log pour libérer de l'espace). Ainsi, le temps de traitement et d'analyse des fichiers logs pose un problème même si aujourd'hui il y a des logiciels spécialisés (par exemple dans le cas d'une attaque, est-ce que l'outil d'analyse nous prévient t-il le plus tôt possible ?). **[AES 04]**

1.8. Exemples sur des fichiers logs du firewall

Prenant comme exemple le firewall *« Kerio Personal Firewall »* version 4.2 qui est un logiciel libre. En outre, ce logiciel est disponible pour la plupart des systèmes d'exploitation: Windows, Dos, OS/2, Unix. Il produit les quatre fichiers logs suivant :

a- Log Into File (filter.log) : Le journal des événements sera sauvegardé dans le fichier filter.log (dans le répertoire d'installation de *Kerio Personal Firewall*). La taille du fichier n'est limitée que par l'espace disponible sur le disque.

Le fichier `filter.log` est utilisé pour enregistrer les différentes actions du Firewall sur un ordinateur local. Il est créé dès le premier enregistrement dans le répertoire d'installation du Firewall (habituellement `C:\Program Files\Kerio\Personal Firewall`). **[SEC 01]**

```
En voici un exemple de ses enregistrements :

1, [08/Jun/2001 16:52:09] Rule 'Internet Information Services': Blocked: In
TCP, richard.kerio.cz

[192.168.2.38:3772]->localhost: 25, Owner: G:\WINNT\SYSTEM32\INETSRV\INETINFO.EXE.
```

Où :

- `1` représente la nature de la règle qui peut être soit : (`1` = refusé, `2` = permis)
- `[08/Jun/2001 16:52:09]` : est la date et heure où le paquet a été détecté (il est recommandé de configurer soigneusement l'heure sur le système)
- `Rule 'Internet Information Services'` : est un nom de la règle appliquée (nom contenu dans le champ Description)
- `Blocked: / Permittted` : indique si le paquet a été bloqué ou autorisé (correspond au nombre qui se trouve au début de la ligne)
- `In / Out` : indique un paquet entrant ou sortant

- `IP / TCP / UDP / ICMP, etc.` : permet de connaître quel protocole de communication est utilisé (pour lequel la règle a été crée)
- `richard.kerio.com [192.168.2.38:3772]` : c'est le nom DNS de l'ordinateur ayant émis ce paquet, entre crochets se trouve l'adresse IP et le port source, séparés par deux points.
- `localhost:25` : l'adresse IP de destination (Nom du DNS) et le port (`localhost` = This computer)
- `Owner` : est le nom de l'application locale (avec le chemin d'accès complet) à laquelle est adressée le paquet. Si l'application est un service système, le nom affiché est `SYSTEM`.

b- Log Into Syslog : il a pour emplacement (`C:\Program Files\Kerio\Personal Firewall`). C'est le journal d'événements qui ont été produits sur le firewall dans un intervalle de temps et qui sera envoyé à un serveur *Syslog* **[ENS 01]**, ayant l'IP définie et qui a pour but de collecter tous les logs de toutes les machines du réseau afin de les regrouper sur une seule machine.

c- Log Packets Addressed to Unopened Ports : ayant le même emplacement (`C:\Program Files\Kerio\Personal Firewall`) son rôle est d'enregistrer les paquets adressés à des ports qui ne sont utilisés par aucune application (typique d'un scan de ports).

d- Log Suspicious Packets : qui se trouve aussi dans le répertoire (`C:\Program Files\Kerio\Personal Firewall`) son rôle est d'enregistrer les paquets que *Kerio Personal Firewall* considère comme suspect. Ce sont par exemple des paquets ne se rapportant à aucune connexion et n'initiant pas de nouvelles connections (les fameux "TCP PING"). **[SEC 01]**

Conclusion

Dans ce chapitre nous avons étudié les de systèmes de protection firewalls, qui constituent un mécanisme de sécurité efficace contre la plupart des attaques initiées par les communautés crackers. Nous avons aussi exposé un outil d'analyse des fichiers logs des firewalls qui constitue un outil très puissant pour l'investigation. Les fichiers logs peuvent contenir les empreintes des attaquants et indiquer ainsi les attaques en cours. Nous avons également constaté que les fichiers logs des firewalls étaient pris en charge par des outils d'analyse.

PARTIE II

LES FICHIERS LOGS DES SITES WEB

2. Les fichiers logs des Sites Web

2.1. Description des fichiers logs des Sites Web

Toute communication entre un navigateur client et un serveur Web est consignée dans le fichier logs sous la forme d'un enregistrement. L'ensemble de tous ces enregistrements constitue la matière première des solutions d'analyse statistique. C'est en effet sur ces données que ces dernières vont s'appuyer pour délivrer les informations sur l'utilisation d'un site web. En règle générale, un fichier log de Site Web contient les données suivantes:

- L'adresse IP (ou le nom du domaine) de l'ordinateur demandant le fichier sur lequel se trouvait l'internaute (ex., 172.20.200.80).
- La date et l'heure de la requête ou de la connexion.
- L'adresse du fichier demandé (Le point d'entrée sur une page, c'est-à-dire l'endroit (page Web) ou le lien a été sélectionné ou encore l'URL de la page d'origine).
- Le protocole et la méthode utilisée pour la requête.
- Le nom du fichier requis ou demandé suivi du résultat de la requête (c.-à-d. les succès, les échecs, les erreurs de serveur, etc.), ainsi que la taille du fichier (le nombre d'octets envoyés).
- Les systèmes d'exploitation et de navigation utilisés par l'ordinateur pour soumettre la requête (Ex de navigateur utilisé : Netscape ou Internet Explorer, sous MacOS ou Windows).
- Transaction (c.-à-d. commande GET)
- Les services appelés. **[MTA 05]**

2.2. Raffinage des données en informations

De nombreuses solutions d'analyse statistique se basent sur les enregistrements contenus dans le fichier logs pour fournir un certain nombre d'interprétations sur l'utilisation d'un site web. Ces interprétations sont, pour la majorité, des estimations calculées selon des algorithmes propres à chaque solution. Il est donc légitime de s'interroger sur la fiabilité de ces estimations et de prendre en compte la marge d'erreur qui résulte du traitement statistique des fichiers logs. Pour réduire la confusion entre information, estimation et extrapolation, il est important

de bien différencier les données disponibles à l'état brut de celles qui apparaissent après l'application d'un traitement statistique spécifique et de fixer les limites d'un tel système.

2.3. Analyse des fichiers logs des Sites Web : estimations et extrapolations

C'est l'étape du processus de transformation des données brutes en informations exploitables par les différents responsables du Site Web. Les données enregistrées dans le fichier log sont compilées, croisées, triées et analysées pour offrir une vision globale de l'utilisation du site internet. Les logiciels d'analyse appliquent plusieurs filtres et traitements successifs sur les données pour déboucher au final sur la création de rapports plus ou moins détaillés. Les estimations qui sont alors délivrées correspondent à des périodes définies et répondent à plusieurs problématiques:

Estimation de la fréquentation du site

- Nombre de visites et de visiteurs.
- Nombre de pages vues.
- Nombre de requêtes ou hits.
- Fréquence des visites pour un même visiteur.

Visibilité du site

- Origine des visiteurs.
- Moteurs de recherche utilisés.
- Mots ou phrases clés utilisés sur les moteurs.

Informations d'ordre technique

- Pages servies avec succès.
- Echec de la liaison.
- Navigateurs utilisés.
- Bande passante utilisée. (Exemple : le débit descendant entre www.arrowbase.com et le terminal **212.31.242.103** est de 56 Kbps (7 Ko/sec))

Informations d'ordre ergonomique

- Première page vue sur le site.
- Dernière page vue sur le site.
- Chemin de navigation suivi par le visiteur.
- Popularité des différentes pages Web.

Bien entendu, cette liste non exhaustive dépend directement des données disponibles dans les fichiers logs. De plus, la prudence est de mise quant à la fiabilité de ces informations puisqu'un certain nombre de limites vient noircir l'ombre d'un tableau pourtant prometteur. **[MTA 05].**

2.4. Exemple de fichier log

Le fichier log peut se composer de plusieurs lignes similaires à l'exemple donné ci-dessous :

62.147.96.38 - - [07/Mars/2003:07:15:21 +0200] «GET /informatique_logiciels.html HTTP/1.1» 200 15288 «http://www.google.fr/search?q=norman+gratuit» «Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; FREE)» -

Où chaque partie de cet exemple a une signification précise :

- 62.147.96.38 : Adresse IP ou domaine (si la conversion en DNS est activée) de l'ordinateur hôte à l'origine de la requête.
- - (premier tiret) : Informations sur le client retournées par identd. Si cette donnée n'est pas disponible, affichage d'un tiret. Cette donnée est très rarement disponible.
- - (deuxième tiret) : Nom d'identification utilisé par l'utilisateur pour s'identifier sur le site (partie protégée par mot de passe, zone personnalisée,...). Si cette donnée n'est pas disponible, affichage d'un tiret.
- [07/Mars/2003:07:15:21 +0200] : Date et heure de la requête.
- «GET /informatique_logiciels.html HTTP/1.1» : Requête HTTP enregistrée et décomposée en trois partie. La partie principale (/informatique_logiciels.html) correspond au fichier demandé. Les deux autres parties (GET .... HTTP/1.1) correspondent respectivement à la méthode et au protocole utilisés.
- 200 : Code renvoyé par le serveur en réponse à la requête. Ce code indique si la requête est un succès ou un échec.
- 15288 : Poids en bytes du fichier transféré en réponse à la requête.
- «http://www.google.fr/search?q=norman+gratuit» : Lien suivi par l'utilisateur pour arriver jusqu'au serveur.
- «Mozilla/4.0 (compatible; MSIE 6.0 « Microsoft Internet Explorer 6.0 »; Windows NT 5.1; FREE)» : Navigateur et système d'exploitation utilisé par l'utilisateur.
- - (dernier tiret) : Valeurs renvoyées par le ou les cookies (selon configuration du serveur). Si cette donnée n'est pas disponible, affichage d'un tiret. **[MTA 05]**

2.5. Les outils statistiques d'analyse

Il existe un nombre important d'outils d'analyse des fichiers logs des Site Web tels que : HitBox Pro de WebSideStory, Site Server 3 de Microsoft, WebTrends Log Analyzer.

2.5.1. Etude détaillée de WebTrends Analyzer

WebTrends version 4.0b est l'outil le plus connu parmi les analyseurs de fichiers logs des Sites Web, il s'installe sur les systèmes d'exploitation : Windows, OS/2, Unix, il permet de surveiller le trafic et les consultations d'un Site Web, et d'en tirer des statistiques. Parmi ses caractéristiques on peut citer :

- La puissance brute du contrôle et de la présentation.
- L'interface simple et configurable.
- Les différents formats.
- La possibilité de l'utiliser autant que service windows. (Voir Figure 03)

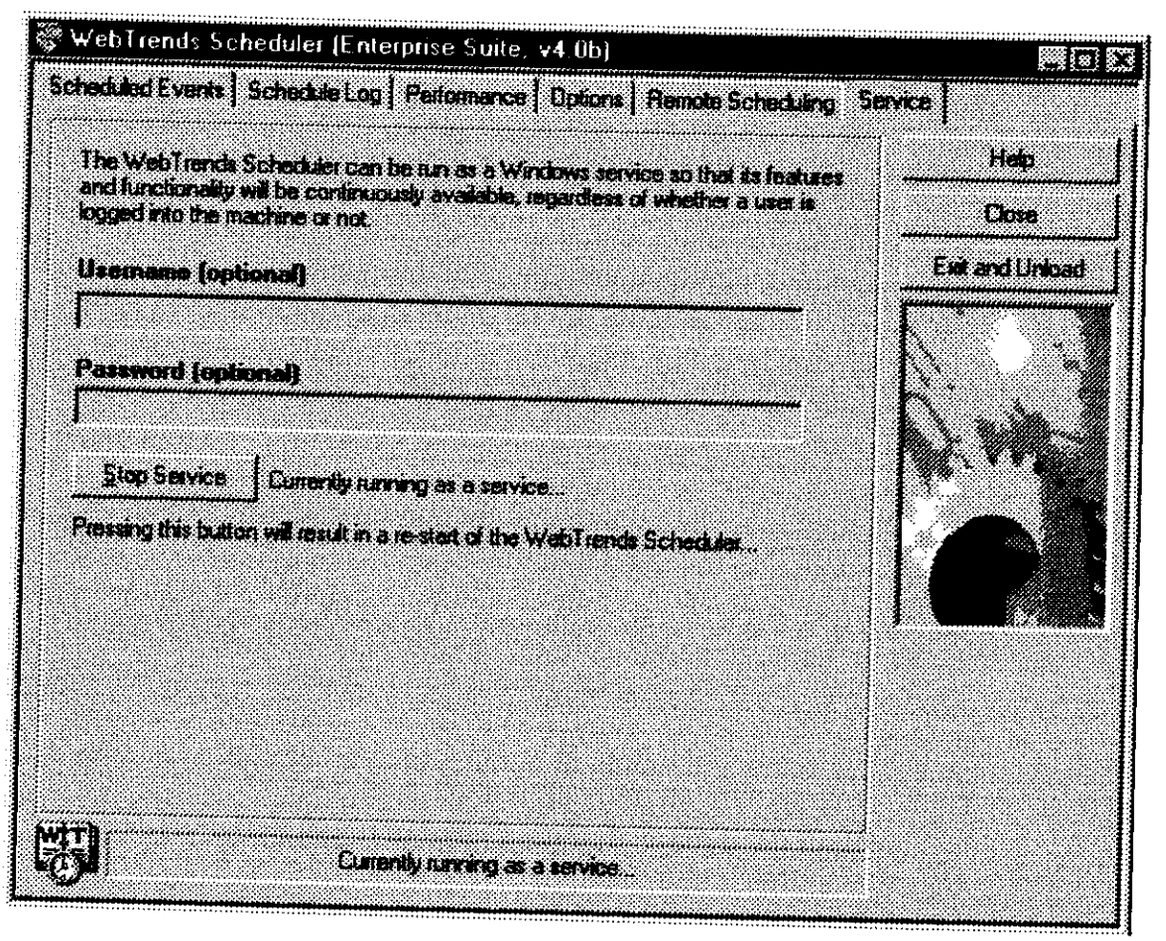


Fig-03 : Utilisation Du service de WebTrends. [WBT 04]

2.5.1.1. Principe de WebTrends

Le principe de ce logiciel est de lire les fichiers logs du Site Web pour en extraire les informations intéressantes et générer des rapports ou des alertes. Ses principales fonctions sont :

- L'interrogation détaillée des fichiers logs.
- L'analyse du trafic Internet.
- La création des rapports de consultation du Site Web.
- La génération des rapports personnalisés dans de multiples formats.
- La génération des alertes. **[WBT 04]**

2.5.1.2. Création du profil d'analyse

Pour que WebTrends construise un rapport de consultation du Site Web, il faut créer un profil d'analyse de trafic en suivant les étapes suivantes :

- Lancer WebTrends.
- [file]/ [new profile].
- choisir "Web trafic analysis".
- sélectionner que le site se trouve sur une machine.
- entrer les infos correspondant à la localisation du fichier log. (Voir Figure 04).
- DNS lookup ne rien changer.
- Dans hostpage préciser le nom de la page de base et de l'adresse internet du site.
- Ne rien changer dans la suite des options. **[WBT 04]**

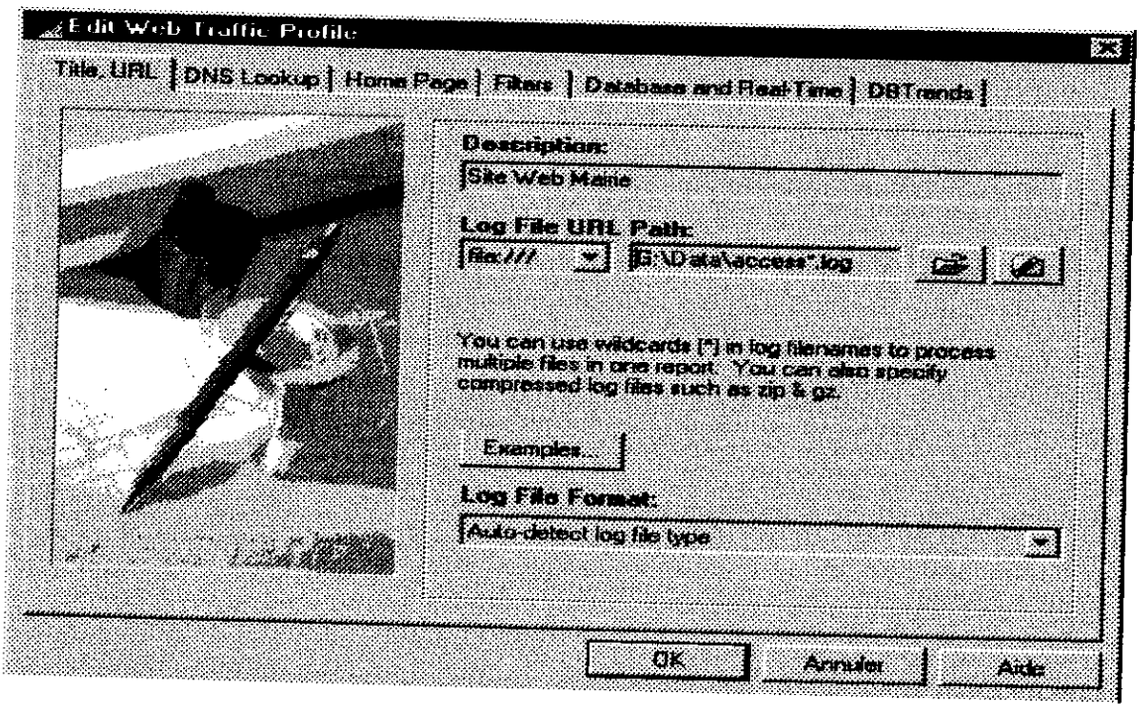


Fig-04 : Création du profil d'analyse [WBT 04]

2.5.1.3. Planifier et automatiser la création du rapport

La création d'un rapport d'analyse de consultation peut être effectuée de manière automatique, planifiée, et intervenir à des intervalles de temps réguliers.

- ouvrir webtrends.
- Onglet "Web trafic analysis" sélectionner le profil d'analyse créé pour analyser le site.
- cliquer sur le bouton "scheduler".
- cliquer sur le bouton "add".
- spécifier le profil d'analyse que l'on veut planifier (voir Créer un profil d'analyse) ainsi que les heures de démarrage.
- Dans l'onglet report choisir le rapport qu'il faut créer (voir Créer un rapport d'analyse).
- cliquer sur OK pour ressortir de cette fenêtre s'il ne faut pas changer d'autres paramètres.

Il faut ensuite s'assurer que le scheduler est lancé en tant que service:

- Dans la fenêtre du scheduler, choisir l'onglet "service".
- entrer les paramètres nécessaires au fonctionnement du service Windows (NT en général).
- cliquer sur le bouton "start service" si le service n'est pas lancé.

De cette manière, si le PC reboote accidentellement, le service scheduler sera relancé et le rapport sera généré de manière normale. **[WBT 04]**

2.6. Importance des fichiers logs des Sites Web

Les fichiers logs permettent donc de recueillir des données, plus précisément, l'analyse des données contenues dans les logs files, elle est pratiquement la seule façon de voir ce qui se passe sur le Site Web. En effet elle permet :

- De faire l'archivage et l'historique du site.
- De retracer un individu indésirable.
- De comprendre un mauvais fonctionnement.
- De planifier des développements.
- De déterminer les activités et certaines caractéristiques des usagers.
- De compter le nombre de visites.
- De savoir en temps réel l'heure et la date de chaque événement produit et permettre d'établir un bilan sur les dates et heures les plus fréquentes d'attaques.
- De connaître les sources et destinations des information circulantes. **[TAS 02]**

2.7. Limites d'une analyse basée sur les fichiers logs

Les limites de l'analyse des fichiers logs comme indicateur de l'utilisation d'un site Internet se situent à plusieurs niveaux.Tout d'abord, certaines données d'identification telles que l'identité de l'utilisateur ne sont tout simplement pas consignées dans le fichier log. De même, il n'est pas possible de connaître la destination de l'utilisateur à sa sortie du site analysé.

Deuxièmement, les requêtes enregistrées dans les fichiers logs ne donnent pas une image exacte de l'utilisation du site web. En effet, certaines pages déjà visitées par l'utilisateur sont automatiquement stockées dans le cache de son navigateur. Dès lors que l'utilisateur revient sur une page déjà visitée, cette dernière est extraite du cache et ne donne donc pas lieu à une interrogation du serveur. L'exemple le plus flagrant de mise en cache réside dans l'utilisation des fonctions "suivant" et "précédent" qui permettent d'accéder directement a des pages mises en cache et n'est donc pas consignée dans les fichiers logs. Enfin, il réside une marge d'erreur importante de part les hypothèses et les méthodes utilisées par les différentes solutions d'analyse pour transformer les données brutes en informations. Ainsi, la plus controversée de ces hypothèses est sans aucun doute de considérer qu'à chaque adresse IP d'une machine ou d'un réseau puisse correspondre un individu puisqu'en réalité un même ordinateur est souvent utilisé par plusieurs personnes (facultés, cybercafés, etc...). Un dernier facteur à prendre en compte est l'indexation des sites Internet par les moteurs de recherche. Ce type de trafic "inhumain" peut artificiellement gonfler le nombre de visiteurs ou de pages vues. **[MTA 05]**

Conclusion

Dans ce chapitre nous avons étudié les Sites Web et leurs fichiers logs. Nous avons aussi exposé un outil d'analyse des fichiers logs des Sites Web qui constitue un outil très puissant pour l'investigation et permet aussi d'établir des statistiques sur la fréquentation, le nombre de visites, sources et destinations des informations circulantes sur des pages Web. Les fichiers logs peuvent contenir les empreintes des attaquants et déterminer les activités et les caractéristiques de certains usagers. Nous avons également constaté que les fichiers logs des Sites Web étaient largement pris en charge par des outils d'analyse.

PARTIE III

LES FICHIERS LOGS DES SGBD

3. Les fichiers log des SGBD

3.1. Définition d'un SGBD

Un système de gestion de bases de données (SGBD) est une collection de logiciels permettant de créer, de gérer et d'interroger efficacement une base de données indépendamment du domaine d'application.

D'un point de vue fonctionnel, les apports escomptés d'un SGBD sont les suivants :

-Supporter les concepts définis au niveau du modèle de données.

-Rendre transparent le partage des données entre les différents utilisateurs.

-Assurer la confidentialité des données.

-Assurer le respect des règles de cohérence définies sur les données.

- Fournir différents langages d'accès selon le profil de l'utilisateur.

-Etre résistant aux pannes.

-Posséder une capacité de stockage élevée.

- Pouvoir répondre à des requêtes avec un niveau de performance adapté.

-Fournir des facilités pour la gestion des méta-données.

Il existe des SGBDS de complexité variable qui possèdent tout ou partie des propriétés ci-dessus. Prenons en exemple deux produits assez caractéristiques : le SGBD relationnel Oracle 9i et le SGBD relationnel Access. Le SGBD Oracle 9i est un SGBD relationnel utilisé pour des applications critiques et qui offre un maximum des caractéristiques présentées ici. Le SGBD Access est un SGBD dans le monde de l'informatique individuelle qui présente l'avantage d'une grande facilité d'utilisation et qui peut convenir à des applications de taille réduite ou moyenne. L'aspect convivial de ce dernier étant évident. En revanche, les niveaux de performance et de sécurité ne sont pas comparables. **[MBD 00]**

Les SGBDs génèrent un ensemble de fichiers logs pour les différents produits qui les composent. Pour notre étude, le choix s'est porté sur le SGBD ORACLE version 9i parce qu'il représente le plus performant et le plus complet des SGBDs.

3.2. Le SGBD ORACLE

Oracle est un système de gestion de bases de données édité par la société Oracle Corporation, leader mondial des bases de données.

Oracle est écrit en langage C et est disponible sur de nombreuses plates-formes matérielles (plus d'une centaine) dont :

- AIX (IBM).
- Solaris (Sun).
- HP/UX (Hewlett Packard).
- Windows NT (Microsoft).
- Oracle et depuis seulement la version 8i est disponible sous Linux. **[ORA 09].**

3.2.1. Architecture du SGBD Oracle

Une base de données Oracle est constituée de plusieurs éléments :

- Des processus chargés en mémoire sur le serveur.
- Des fichiers physiques stockés sur le serveur.
- D'un espace mémoire sur le serveur appelé *SGA* (*System Global Area*).

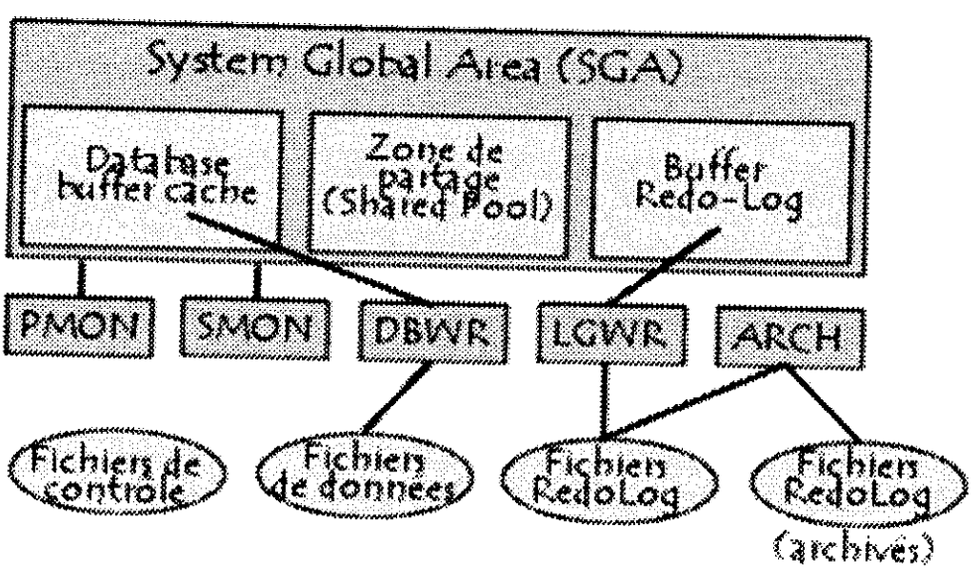


Fig-05 : Architecture du SGBD ORACLE

On appelle instance Oracle les processus et la SGA d'une base de données Oracle. **[ORA 09].**

3.2.2. Les fichiers physiques d'une base Oracle

Les fichiers physiques d'une base Oracle permettent de stocker de manière persistante les données manipulées par Oracle, tandis que la mémoire sert à optimiser la vitesse de fonctionnement de la base de données. **[ORA 09]**

On distingue généralement deux types de fichiers :

- Les fichiers servant à stocker les informations de la base. Tous ces fichiers sont des fichiers. binaires, ce qui signifie qu'ils sont inexploitables avec un éditeur de texte.
- Les fichiers destinés à la configuration et au fonctionnement de la base Oracle.

Oracle a défini une architecture permettant de définir une méthode d'organisation standard des fichiers de la base Oracle. Cette architecture est nommée OFA (Optimal Flexible Architecture).

Les fichiers d'une base de données Oracle sont les suivants :

3.2.2.1. Les fichiers de données (dont l'extension est .dbf). Ces fichiers contiennent l'ensemble des données de la base (les tables, les vues, les procédures stockées, ...). Ils occupent la majeure partie de la base de données, leur taille peut osciller entre quelques Mégaoctets et plusieurs gigaoctets. Ceux-ci contiennent en effet toutes les données relatives à la base Oracle dans un format propriétaire. Ainsi pour modifier les informations contenues dans la base de données il est impossible d'intervenir directement sur ces fichiers; la bonne procédure à adopter consiste à modifier le contenu de la base de données par l'intermédiaire d'ordres SQL.

Les fichiers de données contiennent des informations de deux types :

- Le dictionnaire de données et de travail.
- Les données des utilisateurs.

La lecture de ces fichiers de données est faite à l'aide des processus utilisateurs tandis que l'écriture est assurée par le processus DBWR (Database Writer). **[ORA 09]**

3.2.2.2. Les fichiers de contrôle (dont l'extension est .ctl). Ces fichiers permettent de stocker les informations sur l'état de la base de données (emplacement des fichiers, dates de création, ...). Ils sont créés lors de la création de la base.

Ces fichiers permettent, lors de l'initialisation de la base, de savoir si la base de données a été arrêtée correctement, ainsi que de connaître l'emplacement des fichiers de données et des fichiers Redo Log. Les fichiers de contrôle sont eux-mêmes repérés par le fichier d'initialisation.

Le fichier de contrôle contient les informations suivantes :

- Nom de la base de données.
- Date et heure de création de la base.
- L'emplacement des fichiers journaux (Redo-Log).

Des informations de synchronisation. **[ORA 09]**

3.2.2.3. Le fichier d'initialisation

Ce fichier est un fichier au format texte contenant l'ensemble des paramètres de démarrage de la base (il est généralement nommé initSID.ora, où SID représente le nom donné à l'instance). Son existence n'est toutefois pas majeure car il peut être facilement reconstruit.

Un fichier d'initialisation par défaut est créé lors de la création d'une base. Celui-ci est largement documenté et des exemples de valeurs sont donnés pour chaque paramètre. Toutefois parmi ces paramètres, seul un nombre limité d'entre eux est réellement utile. **[ORA 09]**

3.2.2.4. Les fichiers Log et Redo Log (dont l'extension est .rdo ou .log). Ces fichiers contiennent l'historique des modifications effectuées sur la base de données Oracle. Ils enregistrent les modifications successives de la base de données afin de pouvoir restaurer la base de données en cas de défaillance d'un disque dur. Ainsi le cas échéant, la base de données Oracle est à même de simuler l'ensemble des commandes n'ayant pas été sauvegardées pour rétablir le contenu de la base de données.

Au même titre que les fichiers de données, ces fichiers sont dans un format propriétaire Oracle et l'écriture dans ces fichiers est assurée par le processus LGWR (Log Writer).

Oracle propose également un mode archivage permettant la sauvegarde du fichier Redo-log avant sa réutilisation pour restaurer la base. Si ce mode n'a pas été activé, le contenu du fichier Redo Log est supprimé après utilisation.

Enfin ces fichiers peuvent être multiplexés (comprenez dupliqués dans des répertoires de groupe) afin de fournir un maximum de sécurité. **[ORA 09]**

Notre étude se base sur ces fichiers log que nous allons détailler par la suite.

Une base de données Oracle nécessite au minimum un fichier de données, deux fichiers redo Log et un fichier de contrôle.

3.3. Définition du fichier log

Les fichiers log, sont en fait ce que l'on appelle en français les "fichiers de journalisation". Ils contiennent toutes les transactions effectuées dans la base de données (INSERT, UPDATE, DELETE..) enregistrée en séquence. Ce sont des fichiers en croissance forte qui peuvent parfaitement dépasser de beaucoup la taille des bases de données. Par exemple si, partant d'une base vide, on ajoute et supprime un millier de lignes dans une table, et cela plusieurs centaines de fois, la base sera quasi vide et le journal des transactions important. **[SQL 04]**

3.4. Description des fichiers log

La base de données a la possibilité de réaliser un audit de toutes les actions exécutées dans la base. Trois types différents d'actions peuvent être contrôlés : les tentatives d'accès, les accès aux objets et les actions de la base de données.

Le SGBD ORACLE génère un grand ensemble de fichiers logs. Pour notre étude, le choix s'est porté sur les fichiers suivants que nous avons identifié comme très importants et vitaux pour le bon fonctionnement et le bon contrôle de la Base de Données, ces fichiers sont :

a. Fichier SQL.log

b. Fichier SQLNET .LOG

c. Le fichier alert_oracl.log

d. Fichier alert_orlbase.log

e. Fichier Listner.log

a. Fichier SQL.log

Ce fichier sert a enregistrer les informations suivantes :

- L'adresse de la base de données (ex : ORLBASE.WWW.Mabase.DZ).
- La date et l'heure des opérations effectuées sur la base de données.
- Les opérations effectuées sur la base de données (tel que les modification et changement sur les tables « Alter », création, suppression, mise à jour, insertion, sélection ou gestion des affectations de privilège ou droit d'accès à la base de données).

Exemples :

- 20 juillet 2004 10:28:50 ORLBASE.WWW.Mabase.DZ

ALTER TABLE "Matable"."AUT" RENAME COLUMN "PASSWORD_ID" TO "PASSWORDID"

- 21 juillet 2004 15:08:23 ORLBASE.WWW.Mabase.DZ

INSERT INTO "Matable"."AUT" ("USER_ID" ,"PASSWORD_ID") VALUES ('Matable' ,'woodoo')

- 21 juillet 2004 10:36:09 ORLBASE.WWW.Mabase.DZ

REVOKE ALTER SESSION FROM "Utilisateur"

- 21 juillet 2004 10:36:15 ORLBASE.WWW.Mabase.DZ

GRANT ALTER SESSION TO "Utilisateur" WITH ADMIN OPTION

b. Fichier SQLNET .LOG

Ce fichier signale les tentatives d'accès à la base de données erronées seulement en indiquant les informations suivantes :

- Le message d'erreur dans la tentative de connexion à la base de données et son numéro (ex : Fatal NI connect error 12541, connecting to:) suivi d'une description de la connexion qui contient les informations suivantes :
- La donnée à laquelle l'utilisateur veut se connecter, le nom du service appelé, le chemin d'accès et le fichier demandé, le nom du hôte,et les renseignements concernant l'utilisateur qui sont le nom d'utilisateur, l'adresse ,le nom de l'hôte appelant et le numéro du port.
- On peut aussi visualiser la version du SGBD.
- La version du Système d'exploitation (ex : Windows NT).
- Le protocole utilisé (ex : TCP/IP).
- La date et heure de la connexion.
- L'activation du traçage ou non (ex : Tracing not turned on.)
- La structure du message d'erreur qui est composé de huit champs.

Exemple :

```
Fatal NI connect error 12541, connecting to:
(DESCRIPTION=(CONNECT_DATA=(SID=*)(SERVICE_NAME=Malek)(CID=(PROGRAM=G:\oracle\ora92
\bin\sqlplusw.exe)(HOST= Mabase -SERVER) (USER=Administrateur)))
(ADDRESS=(PROTOCOL=TCP)(HOST= Malek)(PORT=1521))).
VERSION INFORMATION:
      TNS for 32-bit Windows: Version 9.2.0.1.0 - Production
      Windows NT TCP/IP NT Protocol Adapter for 32-bit Windows: Version 9.2.0.1.0-
Production
  Time: 06-04-2003 16:03:44
  Tracing not turned on.
Tns error struct:                          //structure d'erreur Tns
    nr err code: 0                         //code erreur .
    ns main err code: 12541                //code erreur principal de la connexion.
    TNS-12541: TNS:no listener             //type d'erreur.
    ns secondary err code: 12560           //code erreur secondaire.
    nt main err code: 511                  //code erreur principal dans le réseau.
    TNS-00511: No listener                 //type d'erreur.
    nt secondary err code: 61              //code erreur secondaire du réseau.
    nt OS err code: 0                      //code erreur de l'OS.
```

c. Le fichier alert_oracl.log

Ce fichier sert à enregistrer toutes les alertes oracle, il a pour emplacement (g:\oracle\admin**oracl**\bdump\alert_oracl.log).

L'alert_oracl.log enregistre chronologiquement des messages et des erreurs résultant des opérations quotidiennes de la base de données. En outre, il y a des indicateurs pour tracer des dossiers et des dossiers de décharge.

Ces messages incluent :

• La date et heure de l'opération en SQL (Il s'agit de l'action « alter »).

• Messages à la console d'opérateur.

• Erreurs causant des dossiers de trace.

• Créez, changez et baisse rapports de SQL sur des bases de données, des tablespaces et des segments.

Ainsi, ce fichier peut contenir les informations suivantes :

• Le nom de l'opération en SQL (Il s'agit de l'action « alter ») qui est soit sur la base de données ou sur un tablespace (unité logique qui rentre dans la constitution de la BD. En effet, une BD est constituée d'au moins un tablespace nommé SYSTEM qui contient le dictionnaire de données et des informations relatives au système Oracle **[ORL 04]**).

• Emplacement de la table à modifier et son nom.

• Les séquences de log (numéros de séquence log qui sont incrémentées).

Exemple :

```
Thu Feb 12 04:14:24 2004
Thread 1 advanced to log sequence 2
  Current log# 1 seq# 2 mem# 0: G:\ORACLE\ORADATA\ORACL\REDO01.LOG
Fri Feb 13 06:54:13 2004
Thread 1 advanced to log sequence 3
  Current log# 2 seq# 3 mem# 0: G:\ORACLE\ORADATA\ORACL\REDO02.LOG
```

• Le démarrage des instances oracle et leurs arrêts (ex : Starting ORACLE instance (normal), Shutting down instance: further logons disabled).

• Signalisation des erreurs en indiquant la date, l'heure et l'emplacement.

```
Fri Mar 19 22:59:39 2004
Errors in file g:\oracle\admin\oracl\udump\oracl_ora_3528.trc:
```

De plus, on peut trouver d'autres processus ou évènements qui peuvent générer des linges de code dans le fichier Alert_oracl.log, et qu'on peut citer ce qui suit :

• L'arrêt du processus d'archivage. (Peut se faire par l'utilisateur ou de façon automatique).

• Les attentes d'arrêt et activation des serveurs ou des dispatchers (Ces processus dispatcher» sont optionnels. Ils n'existent que si le serveur Oracle est configuré en mode multi-threads et servent d'interprète entre les processus utilisateurs et les processus serveurs **[ORL 04]**).

Exemple:

```
Waiting for dispatcher 'D000' to shutdown
Waiting for shared server 'S000' to die
starting up 1 shared server(s) ...
starting up 1 dispatcher(s) for network address '(ADDRESS=(PARTIAL=YES)(PROTOCOL=TCP))'...
```

• Initialisation des licences de session maximal et des sessions d'alerte a zéro.

LICENSE_MAX_SESSION = 0
LICENSE_SESSIONS_WARNING = 0

• Utilisation des valeurs par défaut des paramètres du log_archive_dest.

```
Using log_archive_dest parameter default value.
```

• Désactivation du système d'audit (SYS auditing is disabled)

• Mise en marche d'Oracle RDBMS (Starting up ORACLE RDBMS Version: 9.2.0.1.0.).

• L'introduction de tous les paramètres système effectués par l'administrateur.

• Ouverture des threads pour les séquences de log.

```
Thread 1 opened at log sequence 1
  Current log# 3 seq# 1 mem# 0: G:\ORACLE\ORADATA\ORACL\REDO03.LOG
Successful open of redo thread 1.
```

• L'annulation d'action d'enregistrement dans le fichier log ou d'un retour arrière contrôlé.

```
SMON: enabling cache recovery //Activation du recouvrement.
Wed Feb 11 10:56:57 2004
Undo Segment 1 Onlined
Undo Segment 2 Onlined
Undo Segment 3 Onlined
Undo Segment 4 Onlined
Undo Segment 5 Onlined
Undo Segment 6 Onlined
Undo Segment 7 Onlined
Undo Segment 8 Onlined
Undo Segment 9 Onlined
Undo Segment 10 Onlined
Successfully onlined Undo Tablespace 1.
```

• Création d'un tablespace en lui indiquant son emplacement de création, sa taille et d'autres propriétés selon l'utilisation de cette table.

Exemple :

```
Wed Feb 11 11:11:55 2004
/* OracleOEM */ CREATE TABLESPACE OEM_REPOSITORY
DATAFILE 'G:\ORACLE\ORADATA\ORACL\oem_repository.dbf'
 SIZE 20975616
 REUSE
 AUTOEXTEND ON
   NEXT 5M
   MAXSIZE 2000M
EXTENT MANAGEMENT LOCAL
PERMANENT ONLINE
Wed Feb 11 11:12:08 2004
Completed: /* OracleOEM */ CREATE TABLESPACE OEM_REPOSITORY
```

• Détection et recouvrement des données endommagées après avoir scanné ses blocs.

Exemple :

```
Wed Apr 09 10:53:16 2003
Beginning crash recovery of 1 threads
Wed Apr 09 10:53:17 2003
Started first pass scan
```

```
Wed Apr 09 10:53:30 2003
Completed first pass scan
 3309 redo blocks read, 64 data blocks need recovery
Wed Apr 09 10:53:43 2003
Started recovery at
 Thread 1: logseq 4, block 78978, scn 0.0
Recovery of Online Redo Log: Thread 1 Group 3 Seq 4 Reading mem 0
  Mem# 0 errs 0: G:\ORACLE\ORADATA\ORLBASE\REDO03.LOG
Wed Apr 09 10:53:49 2003
Ended recovery at
 Thread 1: logseq 4, block 82287, scn 0.891510
 64 data blocks read, 64 data blocks written, 3309 redo blocks read
Crash recovery completed successfully
```

- Le système d'exploitation utilisé, sa version, et le type de CPU.

```
Windows 2000 Version 5.0 Service Pack 3, CPU type 586
```

d. Fichier alert_orlbase.log

Ce fichier est identique au fichier alert_oracl.log, on y trouve le même type d'information vue dans ce fichiers (alert_oracl.log), seulement son emplacement est différent qui est le (G:\oracle\admin\orlbase\bdump\alert_orlbase.log), les enregistrements, les signaux d'alertes et d'erreur concernant la base de donnée oracle sont aussi dans le même fichier.

Exemples :

1-**Message d'erreur :**

```
Sat Nov 15 09:42:26 2003
Errors in file g:\oracle\admin\orlbase\udump\orlbase_ora_3056.trc:
```

2-**Recouverement:**

```
Tue Apr 29 16:58:48 2003
Beginning crash recovery of 1 threads
Tue Apr 29 16:58:49 2003
Started first pass scan
Tue Apr 29 16:59:01 2003
Completed first pass scan
 3410 redo blocks read, 65 data blocks need recovery
Tue Apr 29 16:59:14 2003
Started recovery at
 Thread 1: logseq 24, block 23574, scn 0.0
Recovery of Online Redo Log: Thread 1 Group 2 Seq 24 Reading mem 0
  Mem# 0 errs 0: G:\ORACLE\ORADATA\ORLBASE\REDO02.LOG
Tue Apr 29 16:59:21 2003
Ended recovery at
 Thread 1: logseq 24, block 26984, scn 0.5412332
 65 data blocks read, 65 data blocks written, 3410 redo blocks read
Crash recovery completed successfully
```

e. Fichier Listener.log

Ce fichier comme son nom l'indique sert à enregistrer tous les messages de journalisation dans G:\oracle\ora92\network\log\listener.log. C'est-à-dire qu'il fait de l'écoute sur toute la base de données. Les enregistrements que l'on peut visualiser sur ce fichier sont :

-Les connexions à la base de données, les enregistrements de ce fichier respectent la structure suivante :

TIMESTAMP * CONNECT DATA [* PROTOCOL INFO] * EVENT [* SID] * RETURN CODE

Ce message contient les informations suivantes séparées par des asterix * :

- La date et l'heure.
- La Connect data qui contient à son tour le type de serveur, le nom du service (service name), le chemin du programme à exécuter, l'hôte, et le nom de l'utilisateur.
- Le Protocole info qui contient le type du protocole, l'adresse IP (l'hôte), et le numéro du port.
- L'évènement EVENT qui est de type Etablie (establish).
- Le nom de la base de données.
- Le code de retours qui est 0 si la connexion est établis ou un autre code par exemple 12514 qui est le cas d'une erreur TNS (12514 correspond à une erreur dans le type du server name).

Exemple :

```
06-AVR-2003 13:55:04 *
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORLBASE.DSI)(CID=(PROGRAM=G:\
oracle\ora92\bin\sqlplusw.exe)(HOST=DSI-SERVER)(USER=Administrateur))) *
(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.1.69)(PORT=2752)) * establish *
ORLBASE.DSI * 0
```

-Les types d'événements : ils sont principalement de trois types :

```
06-AVR-2003 12:49:33 * service_register * ORLBASE * 0
06-AVR-2003 12:49:43 * service_update * ORLBASE * 0
06-AVR-2003 12:49:50 * service_died * ORLBASE * 12537
```

- La description de l'écoute qui est de deux ligne comme le montrent les lignes de code suivantes :

```
1) Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dsi-
server.www.dsi.dz)(PORT=8080))(Presentation=HTTP)(Session=RAW))
2) Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dsi-
server.www.dsi.dz)(PORT=2100))(Presentation=FTP)(Session=RAW))
```

3.5. Importance des fichiers log d'un SGBD

En effet les différentes sauvegardes : base complète, base différentielle ou journal des transactions permettent de récupérer une base corrompue.

Tant que ces fichiers ne sont pas tronqués, la récupération et l'interrogation sont en principe toujours possible.

C'est pourquoi tant que la sauvegarde base + journal n'a pas eu lieu, il n'est pas possible de tronquer les fichiers.

Les fichiers logs permettent donc de recueillir des données, c'est pratiquement la seule façon de voir ce qui se passe sur la base de données. En effet ils permettent :

- De faire l'archivage et l'historique de la base de données.
- De comprendre un mauvais fonctionnement, ou de signaler les erreurs.
- De planifier des développements.
- De déterminer les activités et certaines caractéristiques des usagers.
- De visualiser les requêtes effectuées sur la base de données.
- De visualiser les tentatives d'accès à la base de données.
- De savoir en temps réel l'heure et la date de chaque requête et permettre d'établir un bilan sur les dates et heures qui les concernent.
- De connaître les sources et destinations des information circulantes (Insertion, MAJ, Suppression.). Ainsi que d'autres opérations effectuées sur la base de données.

Conclusion

Dans ce chapitre nous avons exposé le SGBD ORACLE, les fichiers logs qu'il génère (les fichiers logs de l'outil Oracle Entreprise Manager OEM), nous avons aussi montré leurs importance, leurs caractéristiques. Mais il n'existe pas des outils permettant une exploitation de ces derniers.

Des efforts en matière de conception des systèmes ainsi que les précautions d'usage pour la sécurité demeurent nécessaires mais toujours insuffisants lorsqu'on considère le nombre, sans cesse, croissant des attaques.

Il nous paraît très réaliste d'utiliser les fichiers logs pour la détection d'intrusion ou d'autres types d'attaques avec plus de rigueur. En effet, la possibilité, pour un système donné, de journaliser dans une machine distante est un acquis très important. Pour cela Il faudra, alors, déterminer ce minimum d'information d'audit à conserver sur celle-ci pour pouvoir les analyser rapidement. Aussi, à quelle fréquence raisonnable, on devrait effectuer ce transfert de données et sur quel type de support écrire? Et le plus important est avec quel outil analyser ces informations, d'où la nécessité de mettre en place des mécanismes d'analyse des fichiers logs des SGBD et pour notre cas des fichiers ORACLE.

Également, la machine distante ne devrait, en aucun cas être accessible à l'intrus. Le seul processus d'application qui devrait y accéder serait le processus de transfert des données d'audit.

CHAPITRE III

LES APPROCHES POSSIBLE

Introduction

Apres avoir étudié les différents types de fichiers logs on a pu remarquer la particularisation de la saisie et des traitements en fonction des fichiers, le contrôle en différé des données ce qui augmente les délais et le risque d'erreur ainsi que la grande redondance des données. Ce qui nous a pousse à restructurer ces fichiers logs, les uniformiser et standardiser les traitements pour avoir un contrôle immédiat de la validité des données et un partage de données entre plusieurs traitements, ce qui facilite les accès et l'interprétation.

L'utilisation de fichiers log impose la connaissance de l'organisation (séquentielle, indexée, ...) des fichiers qu'on utilise afin de pouvoir accéder aux informations dont on a besoin. Les données associées sont mal définies et mal désignées, redondantes, peu accessibles.

On a donc cherché des solutions tenant compte à la fois des désirs des utilisateurs et des progrès techniques. Cette recherche a abouti à deux approche l'une se base sur les bases de données classiques et l'autre sur l'utilisation d'XML et ses composants.

1. Approche 1 : Base de Données classique

Le traitement de masses d'informations (Cas des fichiers logs) est de plus en plus grand dans les différents environnements. Dans ce contexte, les bases de données sont utilisées de façon intensive pour de nombreux domaines d'application. Ces applications concernées par l'utilisation d'un SGBD possèdent des caractéristiques différentes tant au niveau du volume de données concernées qu'au niveau de la complexité de ces données et des traitements informatiques à réaliser. Néanmoins, le regroupement des données et la collection d'informations sur un sujet qui est exhaustif, non redondant, structuré et persistant dans une base de données gérée par un système de gestion de base de données apporte de nombreux avantages dans la plupart des cas.

1.1. Objectifs et avantages des BD classiques et des SGBD

Les bases de données et les systèmes de gestion de bases de données ont été créés pour répondre à un certain nombre de besoins et pour résoudre un certain nombre de problèmes.

Les principaux objectifs et avantages offerts par les bases de données concernant notre cas sont :

- Manipulations des données par des non informaticiens.
- Efficacité des accès aux données (résultats de consultation sous forme de listes et de tableaux).

- Administration centralisée des données.
- Non redondance des données et des informations.
- Cohérence des données.
- Partageabilité des données.
- Sécurité des données.
- Résistance aux pannes.

Malheureusement, ces objectifs ne sont pas toujours atteints à cause de la complexité de quelques applications. En ce qui concerne les inconvénients des bases de données on cite :

- L'absence de commentaire ce qui rend la compréhension du fichier log plus difficile.
- La non portabilité des bases de données a cause de leurs tailles.

2. Approche 2 : La Norme XML

XML est l'acronyme d'eXtensible Markup Language, Le Langage de balisage extensible.
XML permet de structurer l'information dans des fichiers textes. On peut l'utiliser typiquement comme fichier de configuration pour des programmes mais aussi pour enregistrer des résultats. XML s'impose de plus en plus car il permet de structurer l'information sous une forme plus robuste que les fichiers binaires ou tabulaires. Cette forme permet à la fois de faciliter le traitement informatique (Internet, intranet, tableur, base de données,...) tout en conservant un support texte lisible et éditable sans outil particulier par l'être humain ce qui garantit une meilleure pérennité de l'information.

2.1. Principaux avantages et inconvénients d' XML

Parmi les avantages on peut citer que XML est :

- Standard : Cela signifie qu'il existe de nombreux outils informatiques qui permettent de lire ou d'écrire du code XML.
- Strict : Une représentation XML obéit impérativement à une syntaxe. Ça permet de garantir que le fichier soit toujours lisible. Si le fichier est incorrect, Internet Explorer indiquera l'endroit de l'erreur.
- Structuré et hiérarchique : Le fichier contient des <BALISES> qui peuvent contenir d'autres balises et ainsi de suite (hiérarchie). L'ordre d'apparition des balises est conservé.
- On peut ajouter des commentaires : Les commentaires sont des éléments prévus par la spécification. On peut en rajouter dans le fichier sans casser la structure. Ceci permet de commenter les fichiers log afin de garantir une meilleure pérennité de l'information.

• Lisible : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML.

• Universel et portable : les différents jeux de caractères sont pris en compte.

• Déployable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http.

• Intégrable : un document XML est utilisable par toute application pourvue d'un parser (un logiciel permettant d'analyser un code XML).

• Extensible : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

• Ainsi, XML est particulièrement adapté à l'échange de données et de documents.

Parmi les inconvénients on trouve que :

• Le principal inconvénient est le format texte lui même Les données stockées au format texte sont en général plus volumineuses que celles stockées au format binaire.

• Le tabulaire est mieux compris par Excel : mais les choses ne peuvent pas forcément toujours être décrites par des tableaux 2-dimensions. De plus on peut difficilement ajouter des commentaires dans les fichiers tabulaires.

Conclusion

Le format XML n'enregistre que les données textuelles et la structure des données un peu comme une base de données. Il peut donc être particulièrement utile pour des données structurées.

XML étant de plus en plus utilisé (pour la rédaction de documents, la structuration de données) il s'est rapidement fait ressentir le besoin de trouver un moyen de stocker ces informations de la manière la plus efficace possible.

XML ne pourras pas remplacer les BD relationnelles. Ce n'est d'ailleurs pas son but, puisqu'ils ont tous deux des objectifs différents et sont parfaitement complémentaires. Alors que les BDR sont plus adaptés au stockage de données, le XML est plus adapté au stockage de documents. Donc nous constatons que XML est plus adéquat pour la structuration des données contenues dans les fichiers logs. Donc nous optons pour l'utilisation d'XML et de ces fonctions pour mieux gérer les fichiers logs.

CHAPITRE IV

LA NORME XML

Introduction

XML est l'acronyme d'eXtensible Markup Language, Le Langage de balisage extensible. C'est le standard soutenu par le World Wide Web Consortium (W3C) pour le balisage de documents. Il a été conçu pour rendre l'utilisation de SGML (Standard Generalized Markup Language) sur le Web facile et directe, il simplifie le degré d'optionalité de SGML tout en permettant de développer sur le Web des types de documents créés par l'utilisateur. Avec XML, il sera facile de définir des types de documents, facile de créer et de gérer des documents définis en SGML et facile de les transmettre et de les partager sur le Web. Il permet de supprimer deux contraintes qui limitent les développements du Web :

- dépendance envers un type de document unique et non flexible (HTML).
- complexité du SGML intégral, dont la syntaxe autorise un grand nombre d'options puissantes mais difficiles à programmer.

Ainsi, dans la mesure où il permet aux sociétés et à leurs ordinateurs de communiquer plus facilement, XML est le socle d'un ensemble de nouvelles façons de communiquer au travers d'Internet.

1. Origine et objectifs de XML

Le développement de XML a commencé en 1996 et il a été approuvé par le W3C en février 1998. Il a été développé par un groupe de travail (GT) XML [XML Working Group] (initialement connu sous le nom de comité d'examen éditorial SGML [SGML Editorial Review Board]). **[W3C 05]**

Année	Langage	Description
1950	Hypertexte	Premier développement théorique par Ted Nelson
1969	Generalized Markup Language (GML)	Développement du GML par IBM
1986	Standard Generalized Markup Language	Di dans la norme ISO 8879
1989	Hypertext Markup Language (HTML)	Conçu par Bernes à Genève
1994	Hypertext Markup Language (HTML 2.0)	Adopté comme norme sous l'égide du W3C
1994	Cascading Style Sheets (CSS1.0)	Premier complément apporté au HTML destiné à permettre l'utilisation de format dans les documents HTML
1996	Hypertext Markup Languge (HTML 3.0)	La version 3.2 a été adoptée à la suite de la version 3.0, qui n'a jamais été appliquée
1996	Extensible Markup Language (XML 1.0)	Présentée par le W3C sous forme de proposition de discutions
1997	Hypertext Markup Language (HTML 4.0)	Adopté en décembre 1997 par le W3C sous forme de directive
1998	Cascading Style Sheets (CSS2.0)	Poursuite du développement de la version 1.0
1998	Extensible Markup Language (XML 1.0)	Adopté comme norme sous l'égide du W3C
1998	Extensible Markup Language (XML 1.0)	Présenté en août 1998 sous forme de proposition par le W3C
1998	Extensible Markup Language	Comporte de nombreuses extensions du XML ainsi que plusieurs autres définitions de langage

Tab-01 : Naissance de XML

1.1. SGML : (Standard Généralized Markup Language : langage normalisé de balisage généralisé) ISO 8879 :1986.

Il constitue la première tentative de combinaison d'un format de données universellement échangeable avec une importante capacité de stockage d'informations. Il s'agit donc d'un langage de type texte s'utilisant pour baliser des données (donc leur ajouter des métas donnés) de manière auto-descriptive.

SGML a été conçu pour être un standard de balisage de donnée multi-usages.

Avantages :

- Utilisé dans l'industrie pour la création des grandes documentations technique ou lexicographique.

• Impose une stricte séparation entre la description structurelle et la mise en forme d'un document.

• Possibilité d'utiliser le même document sur tout type de matériel et de pouvoir l'échanger facilement.

• Assure la stabilité de la norme officielle.

• Indépendance par rapport aux matériels informatiques.

Inconvénients :

• Trop complexe pour l'implémentation des navigateurs par les industriels.

• Tout document doit être conforme à une structure de document type spécifiée et validée.

• N'offre pas - par lui-même – des liens bien adaptés à la création d'un hypertexte ouvert qui associe plusieurs documents (L'existence de ces liens est liée à la norme complémentaire HyTime : ISO 10744 :1992.

• Manque d'éditeurs et de visualisateurs SGML. **[RAH 03]**

1.2. HTML: (Hyper Text Markup Language)

HTML, issu de la famille SGML, est un langage universel de description permettant de structurer et d'afficher différents objets sur un écran.

Actuellement, HTML est considéré comme standard de l'édition électronique en devenant le premier format de production de documents hypermédia sur Internet, il s'est développé comme un langage de mise en page de documents pour le web.

Avantages :

• Facilité d'apprentissage.

• Disponibilité d'outils de visualisation et de création (éditeur HTML).

• Possibilité de faire générer automatiquement des documents par une application, inclusion d'image, de vidéo ou de sons (inclusion d'objets).

• Langage de script permettant de donner un comportement dynamique aux documents.

• Permet des liens hypertextes vers des documents : autres pages Web, images, sons, recherche dans une base de données... etc.

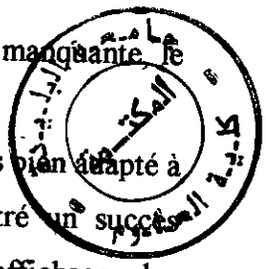
Inconvénients :

• Trop restrictif : impossibilité de créer ses propres balises (langage sémantiquement figé).

• Trop porté vers la présentation et non sur la description, ce qui pose problème pour les navigateurs textes, les PC de poche, ...etc.

• Pour une même page, le HTML donnera un résultat sensiblement différent suivant le navigateur utilisé

• Pas assez sévère au niveau de la syntaxe, malgré une balise non fermée ou manquante, le document sera correctement affiché avec la plupart des navigateurs.



HTML reste malheureusement un langage d'une complexité telle, qu'il n'est pas bien adapté à l'échange de données sur le web. En outre même si le HTML a rencontré un succès incroyable il demeure d'une portée limitée : Sa seule fonction était l'affichage des informations dans un navigateur.

C'est pour cette raison que le XML, a vu le jour. **[RAH 03]**

1.3. XML :

1. Définition :

XML (eXtensible Markup Language) est un sous-ensemble de SGML, débarrassé de toute complexité superflue afin d'être utilisé sur le web. Il a été conçu pour être totalement compatible avec SGML. **[W3C 05]**

Le langage XML possède dix règles de base qui ressortent des objectifs fixés par le W3C :

• XML doit être directement utilisable sur Internet et sans difficultés.

• XML doit pouvoir supporter une large variété d'applications.

• XML doit être compatible avec SGML.

• Il doit être facile d'écrire des programmes qui puissent traiter les documents XML.

• Le nombre des caractéristiques optionnelles d'XML doit être minimum, idéalement il doit même être nul.

• Les documents XML doivent être humainement lisibles et raisonnablement clairs.

• La construction de documents XML doit pouvoir être préparée rapidement.

• Elle doit être formelle et concise.

• Les documents XML doivent être facile à créer. **[W3C 05]**

2. Règles d'XML :

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

• Il permet de définir ses propres balises et ses propres attributs. Il est donc plus flexible que HTML qui, lui, ne possède qu'un nombre limité de balise.

• Un document XML peut être validé par des règles strictes, contenues par des DTD ou des Schémas, décrivant sa structure et la hiérarchie de ses données.

• Les informations ainsi que le traitement de la mise en forme sont rigoureusement séparés de la structure du document XML.

• XML est un format standardisé ouvert ne nécessitant aucune licence, intégralement basé texte et qui peut être associé à n'importe quel jeu de caractère.

• XML est un document portable, il peut être lu sur n'importe quelle plate forme car c'est du texte et n'importe quel outil pouvant lire un fichier texte peut lire un document XML.

• De plus en plus d'applications utilisent le format XML ; C'est le cas de certains SGBD (Système de Gestion de Base de Données) mais aussi d'outils de bureautique comme Microsoft Office 2003 ou Sun Open Office. XML est également au cœur de la nouvelle plate-forme de développement de Microsoft.Net.

• Enfin, son interopérabilité et le fait que de grands acteurs de l'informatique dont IBM, Microsoft et Sun préconisent l'utilisation de ce puissant langage. **[W3C 05]**

3. Avantages de XML :

• **La lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML

• **Une structure arborescente** : permettant de modéliser la majorité des problèmes informatiques

• **Universalité et portabilité** : les différents jeux de caractères sont pris en compte.

• **Déployable** : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme http.

• **Intégrabilité** : un document XML est utilisable par toute application pourvue d'un parser (un logiciel permettant d'analyser un code XML).

• **Extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications.

• Ainsi, XML est particulièrement adapté à l'échange de données et de documents. L'intérêt de disposer d'un format commun d'échange d'information dépend du contexte professionnel dans lequel les utilisateurs interviennent. C'est pourquoi, de nombreux formats de données issus de XML apparaissent (il en existe plus d'une centaine) :

• **OFX** : Open Financial eXchange pour les échanges d'informations dans le monde financier

• **MathML** : **Math**ematical **M**arkup **L**anguage permet de représenter des formules mathématiques

• **CML** : **C**hemical **M**arkup **L**anguage permet de décrire des composés chimiques

• **SMIL** : **S**ynchronized **M**ultimedia **I**ntegration **L**anguage permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte,...

4. Concepts de base :

4.1. Document structuré [W3C 05]

L'approche des documents structurés est basée sur deux principes:

1. Un document est un fichier texte (en format texte, exp. ASCII), auquel on superpose des conventions additionnelles qui permettent de représenter le document sous forme d'une structure hiérarchique (en arbre).
2. La structure hiérarchique du document doit correspondre le mieux et le plus explicitement possible à la nature et à la structure de l'information qui doit être véhiculée par le document.

4.2. Document XML

La norme XML permet de stocker dans un fichier des informations structurées. On parle alors de document XML. Ce dernier est alors composé de texte libre et de balises possédant éventuellement des attributs.

Le langage XML crée des documents qui sont bien structurés et par extension tous les langages basés sur le XML sont aussi correctement structurés, ce qui signifie que les données XML sont plus faciles à utiliser.

4.3. Document XML bien formé (Well-formed document)

Un document XML est dit "bien formé" si celui-ci ne respecte que les règles de la grammaire XML (balises fermées, correctement imbriquées...) .Le balisage des éléments est librement choisi. Autrement dit, un document XML est bien formé s'il obéit à toutes les contraintes de forme données dans la spécification XML du W3C. Il doit comprendre la déclaration XML, un seul élément racine, conformité des noms des éléments et des attributs, les balises doivent être correctement imbriquées (le respect de l'imbrication stricte des éléments). Ce document est déclaré correct par un parseur XML. **[W3C 05]**

4.4. Document XML valide (valid document)

Un document XML est dit "valide" s'il est "bien formé " et qu'il possède une DTD (Document Type Definition) .l'utilisation d'une DTD est nécessaire si l'on souhaite valider un document XML. Donc, si le document est bien formé et s'il obéit aux contraintes de structuration et de format définies dans la DTD, il est dit « valide ». Si une DTD est associée à un document et que ce dernier ne respecte pas les contraintes décrites par sa structure alors le document est bien formé mais non valide. **[W3C 05]**

4.5. Document XML minimal

Un document XML peut être minimal c'est à dire ne contenant qu'un élément vide sans attributs. Ce document minimal ne contient pas de prologue.

Terminologie et syntaxe d'un document XML

En réalité un document XML est structuré en deux parties :

- Le prologue.
- l'arbre des éléments XML.

I) Le prologue

Le prologue est la partie introduction dans un document XML. Il concerne tous ce qui se trouve avant la balise de début de l'élément racine du document XML. Il n'est pas obligatoire, mais vivement recommandé (de part la recommandation XML 1.0). Il contient des informations utiles pour le traitement des données qui y sont contenues. Il est, de plus, subdivisé en plusieurs sous-parties. Il inclut trois types de balises :

- la balise de Déclaration XML.
- La balise des Instructions de traitement.
- La balise de déclaration de type de document.

1. La déclaration XML

Elle permet d'indiquer la version de la norme XML utilisée pour créer le document, le jeu de caractères (en anglais encoding) utilisé et l'autonomie du document.

Cette balise a la forme suivante :

```
<?xml version='numéro' encoding='encodage' standalone='yes|no'?>
```

Les constituants de la balise sont :

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

Tab-02 : Constituants de la balise d'une déclaration

L'attribut *version* spécifie la version de XML, requise pour traiter le document, telle que '1. 0'. Cet attribut ne peut être omis. Il indique au navigateur que ce qui suit est un document XML selon sa version'1. 0'.

L'attribut *encoding* indique le type de codage ou bien le jeu de caractères d'encodage utilisé dans le document XML. Le jeu de caractères "ISO-8859-1" par exemple, a l'avantage d'accepter la plupart des lettres avec des accents. Mais il existe d'autres jeux de caractères comme UTF-8 ou UTF-16 plutôt destinés aux anglo-saxons car ils ne reprennent pas les accents. Le tableau suivant nous donne quelques types de codages :

Norme	Correspondance
UTF-8	Jeu de caractère universel sur 8 bits(Unicode compressé).
UTF-16	Jeu de caractère universel sur 16 bits.
ISO-8859-1	Latin1-langues d'Europe de l'ouest et d'Amérique latine.
ISO-8859-2	Latin2-langues d'Europe centrale et Slaves
ISO-8859-3	Latin3-langues Espéranto, Galicienne, Maltaise et Turc.
ISO-8859-4	Latin4-langues Estonienne, Lettonne et Lithuanienne.
ISO-8859-5	Langue Cyrilliques.
ISO-8859-6	Langue Arabe.
ISO-8859-7	Langue Grecque.
ISO-8859-8	Langue Hébraïque.
ISO-8859-9	Latin5-Langue Turc.
ISO-8859-10	Latin-6Langue Groenlandaises et Lapones.

Tab-03 : Tableau des Types de Codages

Si la déclaration XML ne comporte pas l'attribut *encoding* alors l'encodage par défaut sera le jeu de caractères Unicode compressé **[W3C 05]** .

L'attribut *standalone* fait référence à l'autonomie du document. Si la DTD est interne, le document est autonome et la valeur de l'attribut standalone peut être définie à *yes*. Si la DTD référencée est externe la valeur de cet attribut doit être définie à *no*. Si cet attribut est omis, c'est la valeur no qui est prise par défaut.

Cette balise est obligatoire pour avoir un document considéré comme « bien formé » et il est impératif de respecter sa casse.

2. Les instructions de traitement

La déclaration XML se poursuit avec des informations facultatives sur des instructions de traitement à destination d'applications particulières.

Les instructions de traitement permettent aux documents de contenir des instructions pour des applications afin de fournir des informations supplémentaires sur le document aux analyseurs syntaxiques. Leur syntaxe est la suivante :

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

Les constituants de la balise sont :

<?	**Cible**	**Données de l'instruction de traitement**	**?>**
Ouverture de la balise	Un nom XML	chaîne de caractères que l'analyseur XML passera inchangé à l 'application identifiée.	Fermeture de la Balise

Tab-04 : Tableau des Constituants de la balise

La plus petite de ces instructions est sûrement celle constituant le prologue d'un document XML : **< ? XML version="1. 0" ?>**

3. La déclaration de type de document :

Dans un document valide, on trouve généralement un moyen qui nous permet de faire référence à notre DTD et de trouver la racine du document (à l'aide d'un fichier appelé DTD. Cet outil est la déclaration de type de document. Elle permet de spécifier une DTD pour un document XML. Selon le type de la DTD, il existe trois types de déclaration **[W3C 05]** :

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

3.1. La déclaration d'une DTD externe :

La déclaration de l'utilisation d'une DTD externe doit se faire avant l'élément racine et elle est introduite dans un document XML par l'instruction DOCTYPE immédiatement suivi par le nom de l'élément racine, suivi par l'identifiant de la DTD (SYSTEM ou PUBLIC). Pour une référence externe, deux façons existent :

- Grâce à l'identifiant **SYSTEM** :

```
<!DOCTYPE Nom-racine SYSTEM "chemin-DTD-externe">
```

Les constituants de la balise sont :

<	!DOCTYPE	Nom-racine	SYSTEM	chemin-DTD-externe	>
Ouverture de la balise	Instruction XML	Le nom de l'élément racine du document. C'est un nom XML.	On fait appel à une DTD externe	Le chemin de la DTD.	Fermeture de la Balise

Tab-05 : Tableau des Constituants DTD externe

- Grâce à l'identifiant **PUBLIC** :

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

Les constituants de la balise sont :

<	!DOCTYPE	Nom-racine	PUBLIC	Non-identifiant-public	FPI-DTD-externe	>
Ouverture de la balise	Instruction XML	Le nom de l'élément racine du document. C'est un nom XML.	la DTD est publiée et accessible au plus grand nombre d'utilisateurs. C'est une ressource disponible pour tous sur un serveur Web distant.	Le Nom de la DTD (identifiant public)	Le chemin de la DTD externe.	Fermeture de la balise

Tab-06 : Tableau des Constituants DTD externe

Ce type de déclaration est utilisé lorsqu'on fait référence à une DTD publiée à un usage élargi c'est-à-dire une DTD standard.

Nom-identifiant-public : le nom de l'identifiant public. Il se compose de quatre éléments séparés par le caractère « // »et désignant dans l'ordre :

- **type_enregistrement** : un signe + si c'est selon la norme ISO 9070, un signe - sinon ;
- **propriétaire** : nom du propriétaire (entreprise ou personne) ;
- **DTD description**: description textuelle pour laquelle les espaces sont autorisés ;
- **langue** : un code de langue ISO 639.

3.2. La déclaration d'une DTD interne :

Elle doit être déclaré avant l'élément racine, elle est introduite par l'instruction DOCTYPE suivi par le nom que porte l'élément racine. Il est encadrée par les deux crochets [et]. La syntaxe est la suivante :

<!DOCTYPE Nom-racine [déclarations]>

Les constituants de la balise sont :

<	!DOCTYPE	Nom-racine	déclarations	>
Ouverture de la balise	Instruction XML	Le nom de l'élément racine du document. C'est un nom XML.	Les règles de la DTD interne	Fermeture de la Balise

Tab-07 : Tableau des Constituants DTD interne

3.3. La déclaration d'une DTD mixte:

La déclaration de type de document a la forme suivante:

<!DOCTYPE Nom-racine SYSTEM "chemin-DTD-externe" [déclaration]>

Les constituants de la balise sont :

<	!DOCTYPE	nom_racine	SYSTEM	Chemin_ DTD_ externe	déclaration	>
Ouverture de la balise	Instruction XML	Le nom de l'élément racine du document. C'est un nom XML.	On fait appel à une DTD interne au site	Le chemin de la DTD.	L'ensemble des balises de la DTD interne	Fermeture de la Balise

Tab-08 : Tableau des Constituants DTD mixte

4. Les SCHEMAS XML

Les Schémas XML sont une solution alternative aux DTD, d'une autre façon des DTD améliorées, car ils font appel à la syntaxe du XML et supportent les types de données et les espaces de noms. Ils sont employés pour identifier un ensemble de composants à utiliser dans les documents XML et pour fournir les règles de leurs combinaisons correctes.

Comme les DTD les Schémas ont pour rôle de définir :

- Les éléments et les attributs qui apparaissent dans le document.
- Les éléments qui sont des éléments enfants.
- L'ordre de séquence dans lequel les éléments enfants peuvent apparaître.
- Le nombre d'éléments enfants.
- Si un élément est vide ou peut inclure du texte.
- Les valeurs par défaut des attributs.

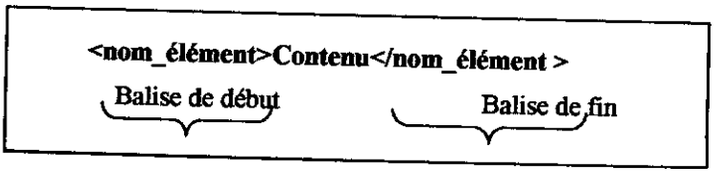
II) L'arbre des éléments XML

Tout document XML est représenté sous la forme d'un arbre d'éléments. Comme tout arbre, il comporte une racine, des branches et des feuilles et qui représentent respectivement l'élément racine, les éléments et les attributs. Ces composants sont les constituants les plus élémentaires d'un document XML .Les commentaires représentent a leurs tours des composants élémentaires.

Cet arbre est constitué d'éléments imbriqués les uns dans les autres (ayant une relation père_ enfant) et d'éléments adjacents. Il représente le véritable contenu du document XML.

1. Les éléments

L'unité de base d'un document XML est dite « élément ». Les éléments permettent d'identifier les différentes sections d'un document XML. Il se présente sous la forme suivante :



Il est pointé par deux marqueurs : un marqueur d'ouverture qui place le nom de l'élément entre un chevron ouvrant (<) et un chevron fermant (>) et un marqueur de fermeture qui ressemble au premier à l'exception du slash (/) qui apparaît devant le nom de l'élément. Le

marqueur d'ouverture est appelé balise ouvrante et le marqueur de fermeture est appelé balise fermante. Comme XML est extensible, il est possible de créer des balises, donc écrire soi-même le nom des balises utilisées. Ces balises XML encadrent le contenu de l'élément :
La syntaxe d'un élément est la suivante :

<Nom_élément Nom-attr$_1$=' val$_1$' ... Nom-attribut$_n$='val$_n$'>
Contenu de l'élément
</Nom_élément >

Toutes les balises doivent être équilibrées : tous les éléments qui pourraient contenir des données textuelles doivent avoir une balise de début et une balise de fin (l'omission est interdite sauf pour les éléments vides). Comme le XML est stricte, toutes les balises ouvertes doivent être fermées.
Dans un document XML, les éléments peuvent être imbriqués mais ne doivent en aucun cas se recouvrir. Les marqueurs doivent être emboîtés correctement. Le XML étant très préoccupé par la structure des données, des balises mal imbriquées sont des fautes graves de sens.

2. Les noms XML

Les noms de balises ou bien les noms XML ou bien encore les noms d'éléments reflètent le type de leur contenu et non la façon dont le contenu devrait s'afficher. Pour composer un nom XML on doit respecter les règles suivantes :

- Les noms peuvent contenir des lettres, des chiffres ou d'autres caractères.
- Les noms ne peuvent débuter par un nombre ou un signe de ponctuation.
- Les noms ne peuvent commencer par les lettres xml (ou XML ou Xml. . .).
- Les noms ne peuvent contenir des espaces.
- La longueur des noms est libre mais on conseille de rester raisonnable.
- Certains signes qui pourraient selon les logiciels, prêter à confusion comme "-", ";", ". ", "<", ">" sont interdits.
- Le caractère ":" est autorisé dans un nom XML (dans le cas des espaces de noms ou bien les domaines nominaux qui sera décrit plus loin).
- Les caractères spéciaux pour les francophones comme é, à, ê, ï, ù sont à priori permis mais pourraient être mal interprétés par certains programmes. **[W3C 05]**

Un document XML est sensible à la casse, un nom d'élément minuscule diffère du même écrit en majuscule ou du mixage des deux. Dans ce cas le respect de la logique de chaque élément est indispensable c'est à dire qu'un nom de balise de début doit être écrit sous la même forme qu'un nom de balise de fin. Plus précisément, La balise d'ouverture et la balise de fermeture doivent être identiques, donc il faut être très rigoureux dans l'écriture de leurs libellés. Toutefois une tendance se dégage pour n'écrire les balises qu'en minuscules, limitant ainsi les erreurs possibles.

3. Elément racine

La première paire de balises d'un document XML sera considérée comme la balise de racine. Elle est unique et obligatoire. Elle encadre le contenu du document XML à produire : tous les autres éléments seront imbriqués entre les balises de l'élément racine.

3.1. Contenu d'un élément

Le texte placé entre les deux balises d'ouverture et de fermeture présente le contenu d'un élément, il est considéré comme partie intégrante de l'élément et il est formaté en conformité avec les règles qui gouvernent cet élément (selon le type de l'élément). un élément, s'il n'est pas vide, peut contenir des données textuelles, des commentaires, des espaces (retour de chariot, retour à la ligne ou tabulation), des références à des entités, des sections littérales et peut contenir aussi d'autres éléments.

Certains caractères ayant une signification particulière dans la grammaire du XML restent interdits tel que « < », « > », « & », « " » et « ' » dans un contenu. De cet effet, les caractères pouvant poser des problèmes à l'affichage, le XML utilise d'autres caractères de masquage ou d'échappement pour les remplacer soit sous forme d'entités nommées, soit sous forme d'entités codées.

Caractère	Entité nommée	Entité codée (entité caractères)
<	<	&#60 ;
>	>	&#62 ;
&	&	&#38 ;
"	"	&#34 ;
'	'	&#39 ;

Tab-09 : Tableau des Caractères interdits

Lorsqu'un document XML inclut des exemples de code source XML ou HTML, les caractères '<' et '&' doivent être encodés et cette solution n'est pas souhaitable pour un document contenant de nombreuses sections de code. Pour palier à ce problème, et pour qu'il ne soit nécessaire de structurer les caractères spéciaux par des caractères de masquage la section littérale **CDATA** intervient. Toute partie comprise entre **< ![CDATA[** et **]]>** est traitée comme une données textuelle brute et elle ne peut pas être imbriquée La mise en œuvre d'une section CDATA s'écrit ainsi :

< ! [CDATA [Données textuelles]]>

4. Les éléments vides

En dehors des éléments standard, XML supporte également des éléments vides c'est-à-dire des éléments qui peuvent être dépourvus de contenu. Un élément vide ne comporte pas de texte entre la balise d'ouverture et celle de fermeture. L'élément vide peut avoir des attributs et il est représenté par la syntaxe suivante :

<Nom-element Nom-attribut$_1$='val$_1$' ... Nom-attribut$_n$='val$_n$'></Nom-element>

Ou bien

<Nom-element Nom-attribut$_1$='val$_1$' Nom-attribut$_2$='val$_2$'...Nom-attribut$_n$='val$_n$'/>

[RAH 03]

4.6. Les attributs

Un attribut est un mécanisme qui permet d'ajouter des informations descriptives à un élément. Ils sont souvent utilisés pour affiner ou modifier le comportement d'un élément. C'est une paire « nom="valeur" » associée à la balise de début de l'élément. Les noms sont séparés des valeurs par le signe « = » et parfois des blancs.

<Nom_element Nom_attribut$_1$=' val$_1$' Nom_attribut$_2$ .....Nom_attribut$_2$='val$_n$'>
contenu de l'élément
</Nom_element >

Les noms d'attributs sont aussi des noms XML et un élément ne doit pas contenir deux attributs avec le même nom.

En XML, la valeur de l'attribut doit obligatoirement être entre des guillemets «" » ou bien des apostrophes « ' » quand la valeur de l'attribut contient elle même des guillemets mais jamais par un mélange des deux. Les attributs ne peuvent être hiérarchiques : ils ne peuvent contenir de sous éléments. La valeur d'un attribut est une donnée textuelles brute, mais elle ne peut inclure les caractères « & » , « < » et «" » et « ' », ils doivent être échappés. Les valeurs des attributs dépendent elles aussi de la casse. **[RAH 03]**

4.6.1. Les attributs prédéfinis

Il existe en XML deux attributs prédéfinis que nous allons présenter : il s'agit de **xml:lang** et **xml:space**. Ils permettent, respectivement, d'indiquer la langue utilisée dans une partie du document et de dire ce que l'on va faire des caractères de séparation.

Dans les deux cas, il faut les redéfinir dans la DTD pour pouvoir utiliser un système de validation (malgré le fait qu'ils soient prédéfinis en XML).

Si l'un de ces deux attributs est utilisé sur un tag (balise), cet attribut et sa valeur seront aussi cascadés sur tous les sous-tags. Il est donc possible de ne spécifier qu'une unique fois l'attribut **xml:lang** (par exemple) pour tout le document : il suffit de le définir sur le tag racine.

- **xml:lang** :

Permet de spécifier la langue utilisée par les données.

```
<Nom_element xml:lang = "Code">contenu</Nom_element>
```

L'attribut **Code** est représenté sous la forme suivante : **code- sous_code**

code : représente la langue du pays.

Sous_code : il indique le code du pays. La valeur de cet attribut est codée sur les deux premiers caractères du nom du pays.

Ces deux sous_attributs sont représentés chacun par les deux premières lettres. La convention veut que le code s'écrit en minuscule et le sous_code en majuscule. Ils sont représentés par les deux premières lettres de la langue et du code du pays.

- **xml:space** : Cet attribut indique si un espace blanc à l'intérieur d'un élément est significatif et ne doit pas être altéré par le processeur XML. Il accepte deux valeurs. La valeur par défaut est "default" et la seconde est "preserve".
- **default :** Le processeur XML est libre de faire ce qu'il désire avec les espaces blancs à l'intérieur de l'élément.
- **preserve :** Les espaces seront préservés tels qu'ils sont dans le document source. Les caractères de séparation seront préservés : les outils de traitement de données XML

pourront ainsi les utiliser. Par caractères de séparation, on entend les espaces, les tabulations et les caractères de passage à la ligne. **[RAH 03]**

4.8. Les Commentaires

Le XML nous permet d'introduire des commentaires. Ce ne sont pas des éléments, ils sont utilisés pour éclaircir le code source et le rendre lisible pour une personne. Ils commencent par < !-- et se termine par -- >. Leur forme est la suivante :

<!-- commentaire -->

Le signe «--» ne doit pas apparaître dans le commentaire ainsi que le triple trait d'union. Ils peuvent être placés avant ou après l'élément racine et dans la donnée textuelle. Par contre, ne doivent pas apparaître à l'intérieur des balises ou dans un autre commentaire (commentaires imbriqués). Le contenu des commentaires n'est pas obligatoirement transmis. Les parseurs peuvent ne pas le faire. **[W3C 05]**

4.9. Inclusion conditionnelle

Les sections conditionnelles permettent d'inclure ou d'ignorer des portions de déclarations dans la DTD :

a) La directive IGNORE

Il existe des directives pour mettre en commentaire une partie des déclarations. Pour que le parseur ignore un bloc d'éléments ou d'attributs, il suffit d'introduire la directive IGNORE dans la syntaxe :

<![IGNORE Bloc de déclaration]>

b) La directive INCLUDE

Pour inclure des blocs de déclarations et pour indiquer que des éléments sont effectivement utilisées dans la DTD, la directive INCLUDE est appelée

<![INCLUDE Bloc de déclaration]>

- **Espaces de noms :**

Dans certains cas, on peut trouver un document XML combinant des balises provenant de plusieurs DTD. Ces balises peuvent avoir le même nom mais réfèrent à des choses différentes. Cela signifie qu'un nom d'élément peut avoir plusieurs significations dans diverses parties du document ce qui rend la validation, l'affichage et le tri, des taches confuses. Donc pour éviter la surcharge sur ces noms d'éléments, le W3C a introduit la notion d'espace de noms. Les espaces de noms représentent une addition récente aux spécifications XML.

L'utilisation d'espace de noms n'est pas obligatoire dans XML mais elle est recommandée. Les espaces de noms ont été crées pour assurer l'unicité parmi les élément XML, et cela en ajoutant un préfixe à chaque élément et attribut provenant d'une même application (assigner les éléments et les attributs à un URL, les éléments et les attributs qui sont rattachés au même URL sont dans le même espace de noms). Il est recommandé de préfixer chaque nom d'élément et ses attributs par une abréviation adéquate unique indiquant la source de sa déclaration en utilisant la forme suivante :

Préfixe:nom-élément

Une déclaration d'espace de noms peut apparaître comme attribut de tout élément, elle reste confinée entre les marqueurs d'ouverture et de fermeture. **[W3C 05]**

4.10. Entités

Un document XML peut être constitué d'une ou plusieurs unités de stockage appelées **entités**, chaque entité a un contenu et toutes (à l'exception de l'entité document et du sous-ensemble externe de la DTD) sont identifiées par un **nom d'entité.**

Chaque document XML possède une **entité document** qui sert de point de départ pour le processeur XML, elle peut contenir le document au complet, elle correspond à la déclaration de type de document et l'élément racine. Il existe deux principaux types d'entités :

- **Entités générales** : utilisables dans un document XML.
- **Entités paramètres** : utilisables uniquement dans les DTD.

Ces deux types occupent des espaces de noms distincts, une entité paramètre est une entité générale de même nom sont deux entités distinctes.

Les entités générales peuvent être internes ou externes c'est-à-dire situées dans la DTD ou dans un fichier externe. Il existe deux sortes d'entités générales :

- **Analysables :** si elles contiennent du texte XML bien formé.
- **Non-analysables :** si elles contiennent du texte non- XML ou des données binaires.

A) Entités générales internes :

Il est possible de définir ses propres entités et d'y faire référence dans le contenu des éléments. Cette définition se fait dans la partie déclaration DOCTYPE.

Syntaxe :

```
< !DOCTYPE nom_type[< !ENTITY nom_entité " valeur_de remplacement ">]>
```

Exemple :

```
< !DOCTYPE TEST[< !ENTIY SI "  département SI " >]>
<P>&SI ;</P>
```

Qui donne : département SI

La référence à une entité générale se fait en préfixant son nom du caractère & et en rajoutant à la fin du nom (;), toute référence à l'entité " SI " dans le document XML utilisant cette DTD sera remplacée par le texte : " département SI".

B) Entités générales externes :

Permettant de construire un document XML à partir de plusieurs autres documents XML complémentaires.

Syntaxe :

```
< !DOCTYPE nom_type[< !ENTIY nom SYSTEM  " URL ">]>
```

Les références d'entités externes sont utilisées dans le document XML de la manière suivante : &nom_entité ;

Exemple :

```
< !DOCTYPE sujet [< !ENTITY documentation  SYSTEM "http://_www.site.com/doc.xml">]>
```

La référence à l'entité documentation se fait :

<aide >&documentation ;<aide/> qui entraînera l'instruction du fichier " doc.xml " dans le document principal. **[RAH 03]**

C) Référence à des entités prédéfinies :

Elles sont définies pour permettre d'éviter les conflits avec les caractères de balisage spéciaux comme l'esperluette (&) et le symbole (<).Voici la liste des prédéfinies :

Appel d'entité	Caractères d'échappement
&lt ;	< (ligher)
&gt ;	> (greater)
&	& (ampersand)
'	` (apostrophe)
&quot ;	" (quotation mark)

Tab-10 : Référence des entités

Exemple :

<**auteur**> Dupon &amp ; al</**auteur**>

Qui donne : Dupon & al

D) Référence à des caractères :

Pour pouvoir insérer un caractère non disponible sur une plate-forme ou sur un clavier, on utilise le numéro de ce caractère dans la table ISO10646 ou UNICODE. Le numéro sera précédé des caractères & # suivi du (;).

Le numéro peut être donné en décimale ou en hexadécimale (on utilisera pour ce dernier le préfixe x " &#x").

Exemple :

<**auteur**> Dupon &#x2267 ; al </**auteur**>

Où : <**auteur**> Dupon &# 38 ; al </**auteur**>

Qui donneront : Dupon & al

5. Les feuilles de style :

Le XML est un langage très attrayant pour écrire et servir des pages Web. Pour pouvoir afficher des documents XML, il serait trop prétentieux d'attacher des feuilles de style à un document XML pour lui donner les instructions nécessaires au rendu de chaque élément. Le contenu doit être formaté et présenté aux utilisateurs. Donc, pour ceci des informations de formatage sont appliquées au document XML et le balisage sémantique est transformé en un langage de présentation.

Les feuilles de style permettent de mettre en forme une page Web ou un fichier log d'une manière équivalente à celle d'un magazine ou d'un journal de la presse écrite créé par un logiciel de publication assistée par ordinateur (PAO). A l'aide des feuilles de style, la gestion des différents éléments de présentation comme les titres, les paragraphes, les images ou bien les tableaux, d'un document XML devient plus pratique et améliore la cohérence et l'ordre au sein de l'ensemble des pages d'un site Internet. Donc, une feuille de style permet à l'auteur de mieux contrôler la mise en page d'un document XML. Or, comme dans tout système où les fichiers peuvent être visualisés au hasard par des utilisateurs arbitraires, l'auteur ne peut pas connaître les ressources (polices par exemple) disponibles sur le système de l'utilisateur; des précautions sont donc de rigueur. La feuille de style est associée au document par une instruction de traitement xml-stylesheet :

<?xml stylesheet href=" URL-feuille" type=" type-feuille"?>

La feuille de style est associée au document par une instruction de traitement xml-stylesheet dans le prologue : après la déclaration XML et avant la balise de début de l'élément racine du document. Cette instruction de traitement utilise des pseudo_attributs pour décrire la feuille de style. **[RAH 03]**

Il existe deux types de pseudo-attributs :

Pseudo-attributs obligatoires

href : indique l'URL ou la feuille de style peut être trouvée.

type : indique le type MIME de la feuille de style : text/css pour une feuille de style CSS, text/xml ou application/xml pour une feuille de style XSLT.

Pseudo-attributs optionnels

Media : contient une information sur le media utilisé par la feuille de style. Il peut être seul ou dans une liste ou les éléments sont séparés par des virgules.

Par exemple : screen, tty, projection,...

Charse : indique dans quel encodage est écrite la feuille de style.

Alternate : indique si la feuille de style est la principale pour un media donné ou une alternative dans des cas particuliers. Elle peut prendre la valeur « no » ou « yes ». Sa valeur par défaut vaut « no », elle indique qu'il s'agit de la feuille de style principale, si elle vaut « yes » le navigateur peut donner a l'utilisateur le choix d'autres feuilles de style dans ce cas il utilise le pseudoattribut title.

Title : il est utilisé quand alternate vaut « yes », il indique à l'utilisateur en quoi la feuille de style diffère (par exemple le choix de la police grande, moyenne,...).

Quand le choix d'une feuille de style n'est pas demandé par un navigateur, le premier choix qui correspond le mieux au type media de l'environnement est pris en considération.

Les langages de feuilles de style actuels majeurs sont :

- **Cascading StyleSheet (CSS).**
- **XSL Formatting Objects (XSL-FO).**

6. les liens XML

L'une des raisons de la croissance phénoménale d'Internet est de permettre à des documents d'être reliés entre eux.

Un lien exprime une relation entre des ressources.

Il existe des techniques qui permettent de relier des documents XML parmi elles :

Xlink et **Xpointer**

6.1. Xlink :

Permet de définir la manière dont un document est lié à un autre (liens simples) ou lié à d'autres documents (liens étendus).

a) les liens simples :

Les liens simples sont à sens unique et impliquent seulement deux ressources : la source (qui est le document origine) et la destination (la cible), la description de la destination du lien se trouve toujours à l'interieur du document source.

Le lien simple est déclaré à l'aide de l'attribut type défini dans " xlink" et prend la valeur " simple».

Les autres attributs d'un lien simple sont :

Xlink :href : Identificateur de la cible du lien.

Xlink :title : caractéristique de la cible (de façon visible à l'homme).

Xlink : role : Caractéristique de la cible (utilisé par une machine).

Xlink : show : Sémantique et compréhension de lien (affichage de la ressource une fois extraite).

Xlink : acuate : Sémantique et compréhension de lien (indique le moment ou la ressource doit être extraite).

Exemple :

```
<personne xmlns :xlink = "http//www.w3.org/2000/xlink/namespace/">
<nom> Dupon </nom>
<image xlink :type ="simple"
        xlink :herf="image.jpg"
        xlink : acuate = "onRequest"
        xlink : Show = "embed "
        xlink :title = " Cliquez ici pour voir l'image ! ">
  Cliquez ici pour voir l'image !
</image>
</personne>
```

b) les liens étendus :

Un lien étendu permet d'associer plusieurs ressources, il est défini sur un élément père déclaré à l'aide de l'attribut **type** qui prend la valeur **"extended"**. Cet élément aura des éléments fils dont le rôle dans le lien sera déterminé par l'attribut xlink :type pouvant prendre les valeurs suivantes :

- **Locator** : indique les ressources distantes (hors ligne).
- **Arc :** regroupe les règles qui permettent de passer d'une ressource à une autre dans un lien.
- **Ressource** : créer des ressource locales.
- **Title** : permet de donner un titre au lien pour faciliter l'emploi de ce dernier. **[RAH 03]**

6.2. Xpointer :

Le pointeur XML est utilisé pour identifier des positions et des fragments dans des documents XML (accéder à des parties des documents) pour les déclarer cibles de liens. Il s'exprime à l'aide du langage Xpath.

Il peut être sous forme de :

1) Un nom isolé.
2) Une séquence d'entier séparé par des slaches (/).
3) Xpointer (expression).
4) Xpointer (expression) Xpointer (expression).
5) Xpointer (expression to expression). **[RAH 03]**

7. Les outils XML

Les outils XML liés au XML se répartissent en une grande variété de catégories. Les groupements majeurs sont :

- **Les éditeurs :** Il existe des éditeurs pour les DTD, pour les schémas et pour les documents XML complets.
- **Les convertisseurs :** Les convertisseurs sont conçus pour traduire des documents d'un langage de balisage à un autre.
- **Les parseurs :** Le principale rôle des parseurs est de parser et d'interpréter les documents XML.
- **Les outils de stockage et de gestion :** Les outils pour le stockage et la gestion des documents vont des bases de données aux moteurs de recherche. Il s'agit d'un domaine de développement rapide, et de nouveaux produits font leurs apparitions fréquemment.
- **Les outils de restitution :** La restitution des documents est également un domaine de développement passionnant. Cette catégorie inclut les outils de publication, les navigateurs Web et les agents logiciels.

8. XQUERY

La recherche en mode texte se révèle beaucoup trop simple et le langage SQL ne peut s'adapter correctement à XML.

Depuis le mois d'octobre 1999 le W3C **[XQL 05]** travaille sur ce problème. Le fruit des efforts du consortium est le langage XML Query ou bien le XQuery.

Ce langage a donc été conçu pour permettre de créer des requêtes précises tout en pouvant s'adapter à tout type de source de données XML, qu'il soit question de bases de données, documents XML ou autres.

XQuery peut être utilisé avec des documents XML validés par des schémas, des DTD ou encore simplement des documents XML bien formés. **[ART 52]**

7.1. Les bases d'XQuery

XQuery est un langage basé sur les expressions. Un script ou programme XQuery contiendra toujours une ou plusieurs expressions et optionnellement des fonctions et des définitions.

En XQuery, il existe plusieurs types d'expressions dont voici une partie :

Type d'expression	Syntaxe
Séquence	expression, expression, ...
Variable	$a, $variable
Constante	'a525g', "XQuery", 100
Numérique	+, -, *, div, idiv, mod
Comparaison générale	=, !=, <, >, <=, >=
Comparaison de valeurs	eq, ne, lt, le, gt, ge
Comparaison de noeuds	is, isnot
Comparaison d'ordre	<<, >>
Logique	and, or
Conditionnelle	if expression then expression else expression
Quantitative	some/every $variable in expression satisfies expression
Switch	typeswitch expression case type $variable return expression default $variable return expression
Ensemble	union, intersect, except
FLWR	for, let, where, return
Validation	validate {expression}
Chemin	$variable/livre[@quantite='5']/auteur
Constructeur	
Fonction	

Tab-11 : Tableau des Expression Xquery

7.2. Expressions de chemin

Les expressions de chemin ressemblent beaucoup à celles que l'on retrouve dans le langage XPath. Prenons par exemple le document suivant dans lequel l'attribut «num» représenterait un numéro associé à un élève et la valeur de la balise, la note de celui-ci.

```
<examen>
 <note num="001">80</note>
 <note num="012">75</note>
 <note num="525">99</note>
 <note num="601">60</note>
</examen>
```

Examinons maintenant l'expression de chemin suivante :

```
//examen/note[@num=$a]/text()
```

Assument que la variable $a contienne une valeur numérique constitué de trois chiffres, l'expression retournerait le texte contenu dans le nœud dont la valeur de l'attribut num est égale à celui de la variable. **[ART 52]**

A) Les expressions FLWR (prononcé flower)

Le nom provient de for, let, where et return.

- for : Fournit un mécanisme d'itération.
- let : Permet l'assignation de variable.
- where : Les clauses for et let génèrent un ensemble de noeuds qui peuvent être filtré par un ou plusieurs prédicats dans une clause where.
- return : Génère le résultat de l'expression FLWR.

Voici un exemple simple d'une requête FLWR. Cette requête a pour but de présenter une comparaison des prix des livres similaires (ayant le même titre) dont l'auteur est Stephen King dans deux librairies affichant leurs produits sur le web.

```
<livres>
{
for $a in document("http://www.libraire1.com/livres.xml")//livres/livre[auteur='Stephen
King'],
$b in document("http://www.libraire2.com/produits.xml")//produits/livre[@auteur='Stephen
King']
where $a/titre = $b/titre
return
<livre>
<prix1>{$a}</prix1>
<prix2>{$b}</prix2>
```

```
<livre>}
</livres>
```

B) Les expressions conditionnelles

Comme dans la majorité des langages de programmation, XQuery offre la possibilité d'utiliser les mots-clef if, then et else. Par exemple, pour un document XML comme celui-ci :

```
<livres>
  <livre qte="1">
    <titre></titre>
  </livre>
  <livre qte="0">
    <titre></titre>
  </livre>
  <livre qte="3">
    <titre></titre>
  </livre>
</livres>
```

Si nous désirons écrire «oui» dans le cas où la quantité en stock est supérieure à 0 et non dans le cas où elle est égale à 0, nous pourrons écrire :

```
for $a in document("livres.xml")//livres
return
  <titre>{$a/titre}</titre>
  <enstock>
    if ($b[@qte='0'])
    then 'oui'
    else 'non'
  </enstock>
```

7.3. Les constructeurs

La construction de nouveaux contenu XML est fondamentale en XQuery. XQuery contient des constructeurs pour des éléments, des attributs, des sections CDATA, des instructions de traitement et des commentaires utilisant une syntaxe qui est la même sinon presque, que le

XML lui-même. Le contenu des éléments et les valeurs des attributs peuvent contenir des expressions placées entre accolades {} qui seront évaluées. Par exemple :

```
let $a := 'a525g.com'
let $b := 'Portail a525g'
return
  <site url="{$a}">
    {$b}
  </site>
```

Le résultat de l'évaluation de ce code donnerait ceci :

```
<site url="a525g.com">
  Portail a525g
</site>
```

[ART 52]

7.4. Les fonctions

XQuery inclut un grand nombre de fonctions et d'opérateurs. Il existe des fonctions pour :

- chaînes de caractères
- mathématiques
- comparaison de dates
- expressions régulières
- noeuds XML
- conversion de types
- etc...

Il est aussi possible de définir nos propres fonctions. Les fonctions sont la plupart du temps dans l'espace de nommage (namespace) «fn». Cet espace de nom est associé à «http://www.w3.org/11/xquery-functions». Celui-ci est utilisé dans le but d'éviter les collisions au niveau de la définition de noms.

Comme mentionné ci-dessus, il est possible de définir nos propre fonctions. Reprenons le document XML utilisé ci-dessus pour démontrer les opérations de comparaison et ajoutons des éléments au document.

```
<livres>
  <livre qte="1">
    <titre>Harry Potter et l'Ordre du Phénix</titre>
    <auteur>Joanne K. Rowling</auteur>
```

```
  </livre>
  <livre qte="0">
   <titre>Sac d'os</titre>
   <auteur>Stephen King</auteur>
  </livre>
  <livre qte="3">
   <titre>Une seconde chance </titre>
   <auteur>Mary Higgins Clark</auteur>
  </livre>
  <livre qte="5">
   <titre>Carrie</titre>
   <auteur>Stephen King</auteur>
  </livre>
  <livre qte="2">
   <titre>Trente ans déjà</titre>
   <auteur>Mike Gayle</auteur>
  </livre>
</livres>
```

Il est possible de définir une fonction qui calculera la moyenne des quantités des livres figurant dans le document dont l'auteur est Stephen King.

```
define function moyenne-qte($auteur)
{
  let $doc := document("livres.xml")//livres/livre[auteur=$auteur]
  return avg($doc/.[@qte])
}
```

[ART 52]

Conclusion

Dans cette partie nous avons présenté en détail XML et on a vu combien il est parfaitement adapté à la standardisation des fichiers logs, la génération des prochaines d'applications Internet, au commerce électronique et aux DNS des entreprises. XML est un langage de balisage allégé, simplifié, souple, d'utilisation aisée et compatible avec les documents internationaux et les fichiers logs. XML constitue le moyen idéal de stockage des données et d'envoi des messages, et permet la validation des documents XML.

CHAPITRE V

CONCEPTION

Introduction

Le but de notre travail est de simplifier, de clarifier et homogénéiser le format des fichiers logs générés par les SGDBs en utilisant la norme XML afin de générer un format standard accessible à travers une interface simple et agréable facilitant ainsi la tâche d'investigation. Dans le présent chapitre, nous allons présenter l'approche de conception que nous allons adopter pour aboutir au fichier standard visé à travers ce travail. Nous allons travailler sur les cinq types de fichier log de SGBD oracle déjà vus au deuxième chapitre.

1. Démarche de Conception :

Dans la figure suivante on peut clairement voir le schéma global du système à mettre en place. Il se présente sous une architecture en couches.

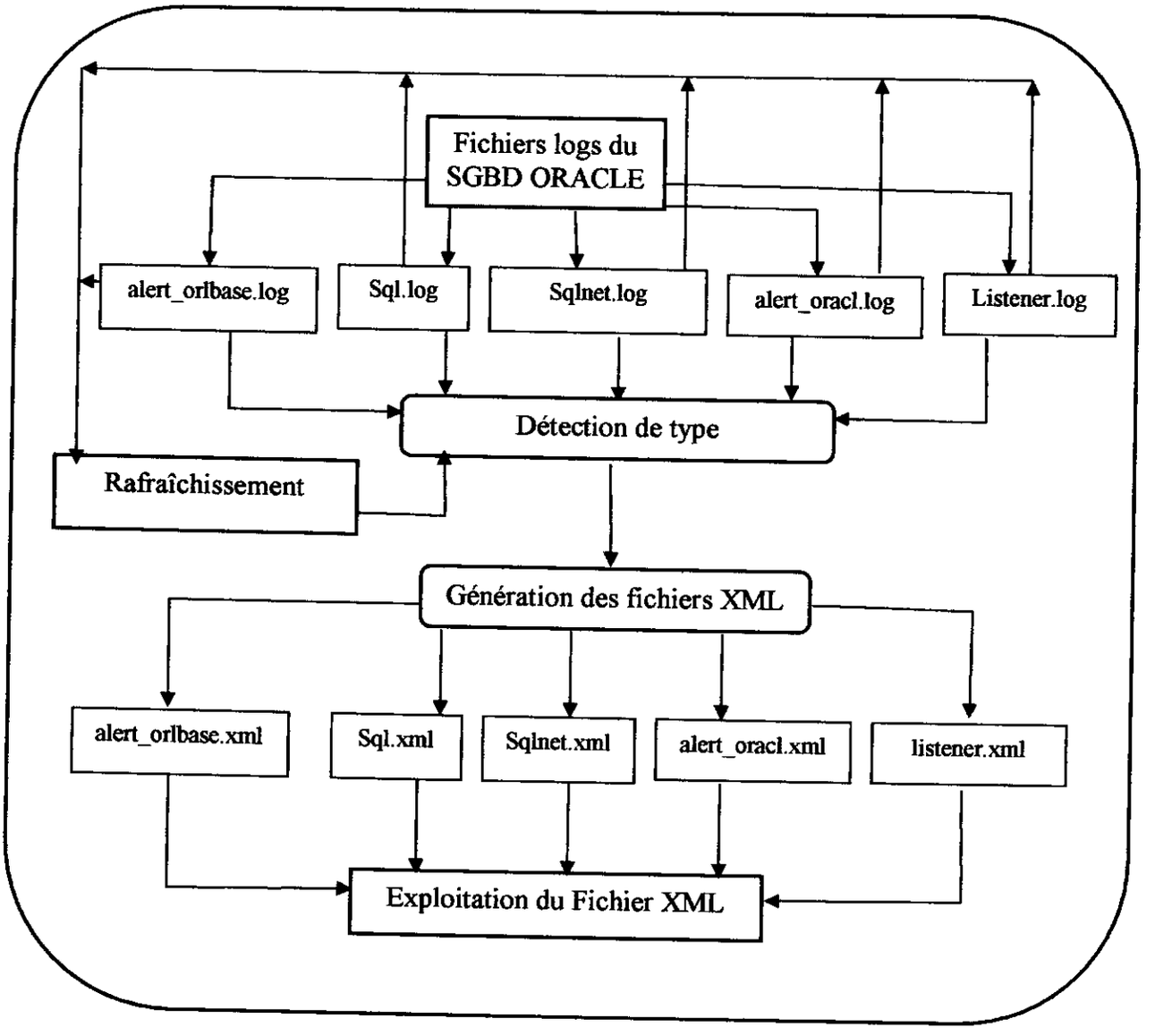


Fig-06 : Démarche Générale

2. Description des différents Couches du système

Après avoir décrit le schéma général de notre système, nous allons ci-présent détailler les différentes couches :

- Couche Détection de Type de log de SGBD.
- Couche Génération des fichiers XML.
- Exploitation des fichiers XML.

2.1. Couche Détection de Type :

Notre travail prend en charge plusieurs fichiers logs et vu qu'un fichier diffère d'un autre, nous avons opté pour une classification en catégories en se basant sur les différences qui existent entre ces fichiers.

Ces différences peuvent être caractérisées par des séparateurs : (' '/','/' ;',''*',....) ou bien par les formats des données comme c'est le cas de la date et l'heure écrite sous format anglais 'mois/ jour /heure/année' ex : Apr 06 12:52:37 2003 ou dans le format français : 'jj/mois/année/heure' ex : 17 juillet 2004 14:14:36.

Mais dans notre cas nous avons opté pour les séparateurs pour la détection de type des différents fichiers logs du SGBD, à noter les fichiers logs alert_orlbase.log, alert_oracl.log, sqlnet.log, sql.log et Listener.log qui sont en quatre catégories pour la détection de leur type :

- **Catégorie 1 :** dont les séparateurs sont des blancs ' ' et des étoiles '*'. Cas du fichier sqlnet.log.
- **Catégorie 2 :** dont les séparateurs seront des tiret '-'.Cas du fichier sql.log.
- **Catégorie 3 :** dont les séparateurs sont la date. Cas Alert_orlbase.log et Alert_orlbase.log.
- **Catégorie 4 :** dont les séparateur sont seulement des blancs ' ' suivis par une ligne de code spécifique au fichier log. Cas du fichier Listener.log.

La figure suivante représente le fonctionnement de la couche Détection de type :

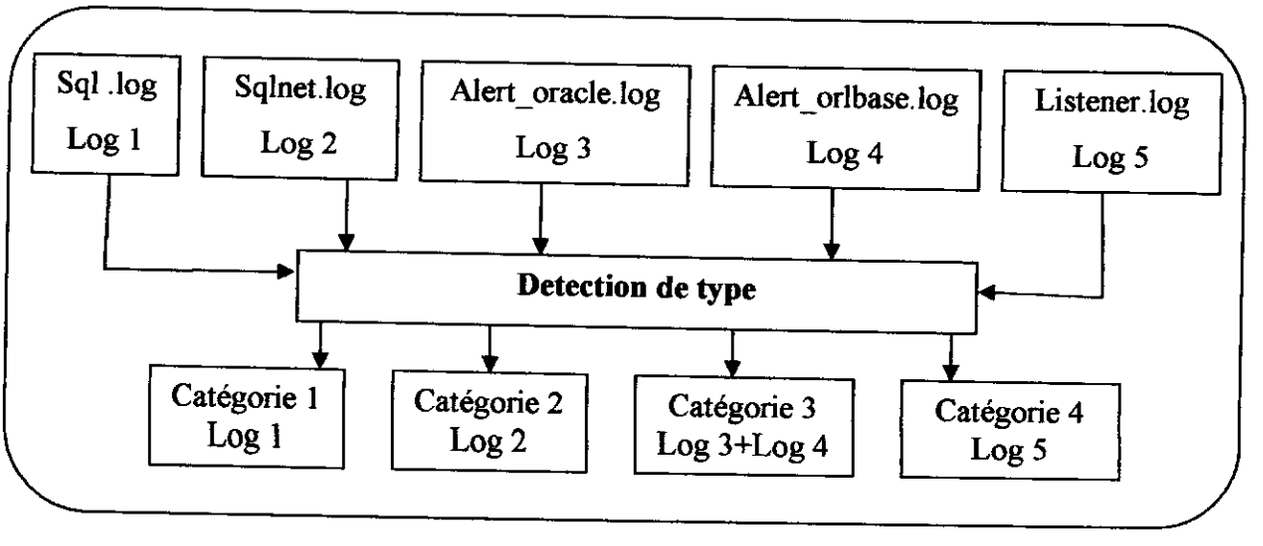


Fig-07: Fonctionnement de la couche détection type

2.2. La phase Génération XML :

2.3.1. Génération du document XML Valide :

Un document XML se compose, d'une part, de texte et d'autre part d'informations de structure. Les informations de structure servent le plus souvent à délimiter le texte, pour identifier essentiellement la sémantique et ce par le biais de balises. Afin d'aboutir à la génération d'un document XML respectant notre structure d'affichage du fichier log, notre système utilise les données déjà extraites, et les mets dans le fichier modèle pour respecter une hiérarchie et structure bien définies du document à générer.

Rappelons qu'un document XML est constitué essentiellement de deux parties :

- Prologue.
- Arbre XML.

La figure suivante schématise la génération du document XML :

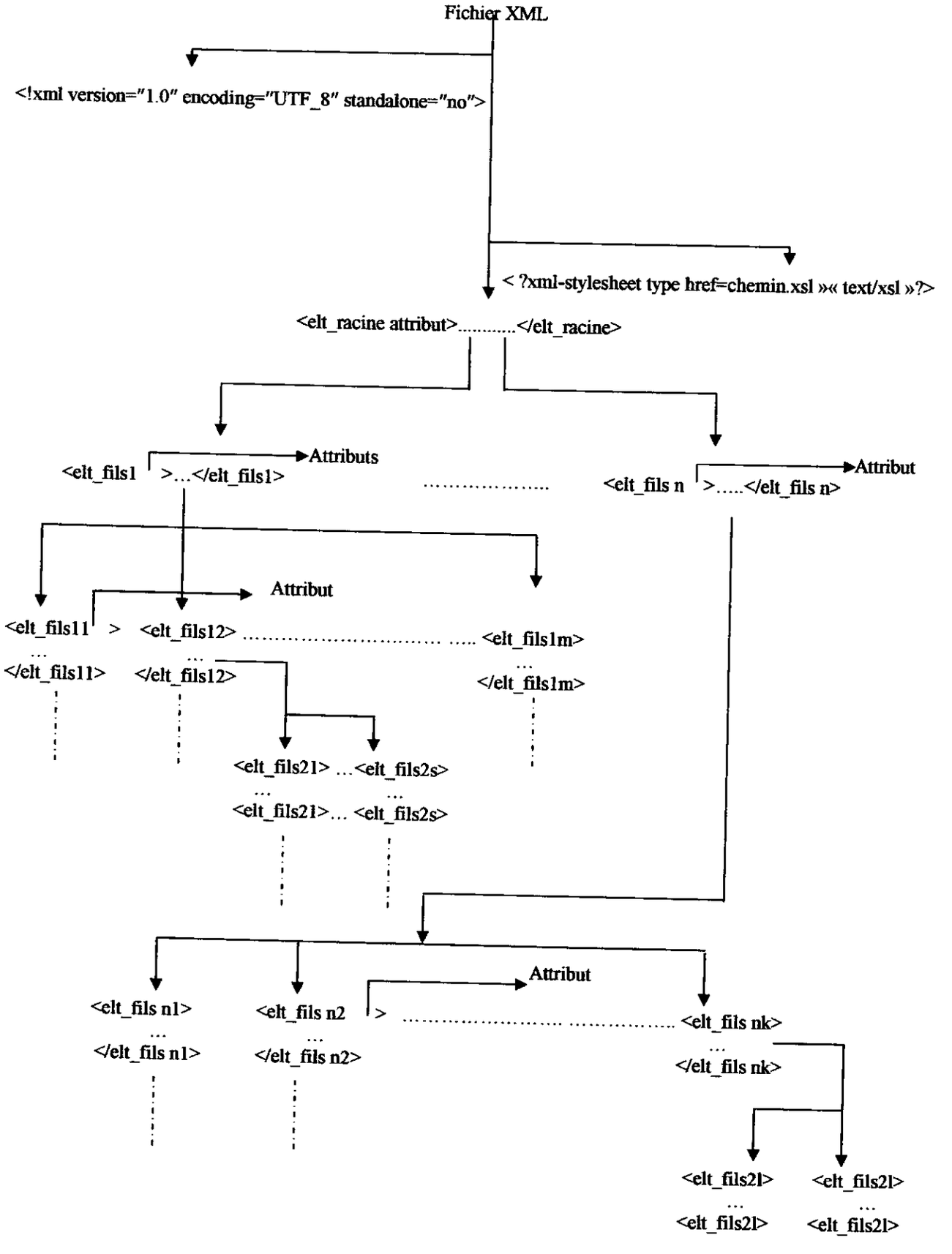


Fig-08 : Génération d'un document XML

Nous allons dans ce qui suit détailler la procédure de la génération du document XML correspondant :

A) Génération du prologue

Il s'agit de la partie introduction d'un document XML. Cette partie inclut les deux balises :

- La balise de Déclaration XML.
- La balise des Instructions de traitement.

1. Génération de la balise Déclaration XML

Etant donné que cette balise fournit des informations sur la version de la norme XML, le codage utilisé et l'autonomie du document. Le système doit impérativement la générer, en premier, afin de respecter les contraintes du XML. L'affectation de valeur pour ces attribut est faite comme ci-dessous :

- L'attribut *version* : le système lui affecte la valeur "1.0" par défaut car c'est la version traitée par notre système.
- L'attribut *encoding* : il prend la valeur "ISO-8859-1" car ce jeu de caractères a, pour les francophones, l'avantage d'accepter la plupart des lettres avec des accents. Si ce codage ne répond pas au besoin de l'utilisateur (l'utilisation d'autre jeu de caractères), le système lui donne la main pour inclure le codage qui lui convient par le biais d'une boite de dialogue qui contient la liste de tous les codages.

La balise générée a la syntaxe suivante :

```
<?xml version="1.0"encoding="ISO-8859-1"standalone="no"?>
```

2. Génération de la balise feuille de style

L'instruction de traitement permet de fournir des informations supplémentaires sur le document XML. Les feuilles de styles sont des instructions de traitement apportant une information sur la mise en forme et l'affichage d'un document XML. Afin de permettre à l'utilisateur d'associer une feuille de style au document s'il le désire, le système met à sa disposition une boite de dialogue pour introduire le chemin de la feuille (Chemin_fichier.xsl). Cette balise à la forme suivante :

```
<?xml-stylesheet  href ="Chemin_fichier.xsl" type = " text/xsl "?>
```

B) Génération de l'arbre XML

Un document XML est une imbrication de balises d'éléments qui doivent apparaître dans l'ordre de leurs déclarations dans le fichier XML. Afin d'assurer la production des documents XML valides, le système procède à la génération des balises de l'élément racine suivie de la génération des balises des éléments fils. Le mécanisme de génération se base sur la récupération des données contenues dans le fichier log renseigné en les mettant en correspondance avec le fichier modèle.

1. Génération de la balise élément racine :

L'élément racine est obligatoire et unique dans un document XML. Il encadre l'arbre du document. Le système récupère le nom de cet élément du fichier modèle et le chemin du fichier XML associé ensuite génère ses balises comme suit :

```
<Nom_racine xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
            xsi : noNamespaceFichierLocation="Chemin_Fichier.xsd" >
                    :
                    :
                    :
                </Nom_racine>
```

La couche génération qui a pour rôle de convertir les fichiers logs de leur format vers le nouveau format XML ce qui est crucial pour notre système vient directement après la détection du type de chacun des fichiers logs, et l'extraction des informations. Cette phase consiste à définir ce qui suit :

1) **La structure :** la définition de la structure pour chaque fichier log consiste à :

- Reconnaître les diverses colonnes du listing de sélection.
- Choisir quelles lignes et quelles colonnes doit-on retenir pour la sortie, ce qui caractérise une opération de filtrage.
- Transformer les données retenues en une structure de sortie convenant aux besoins, ce qui est une opération de transformation.

2) **Le contenu :** la description du contenu de chaque fichier log, en définissant les éléments et enregistrements de chacun des fichiers. (les éléments et informations concernant chaque fichier ont été étudiés dans la Partie III du chapitre II).

3) **La présentation :** la mise en page du fichier log sous sa nouvelle forme standards XML.

Chacun des fichiers logs a un format approprié selon ses champs d'informations comme le montre chacune des figures suivantes :

Catégorie 1 : « Fichier sql.xml» :

Après avoir définis sa structure et les champs qu'il doit contenir le fichier sql.log, la génération de ce fichier nous est présenté sous la forme d'enregistrement comme suit :

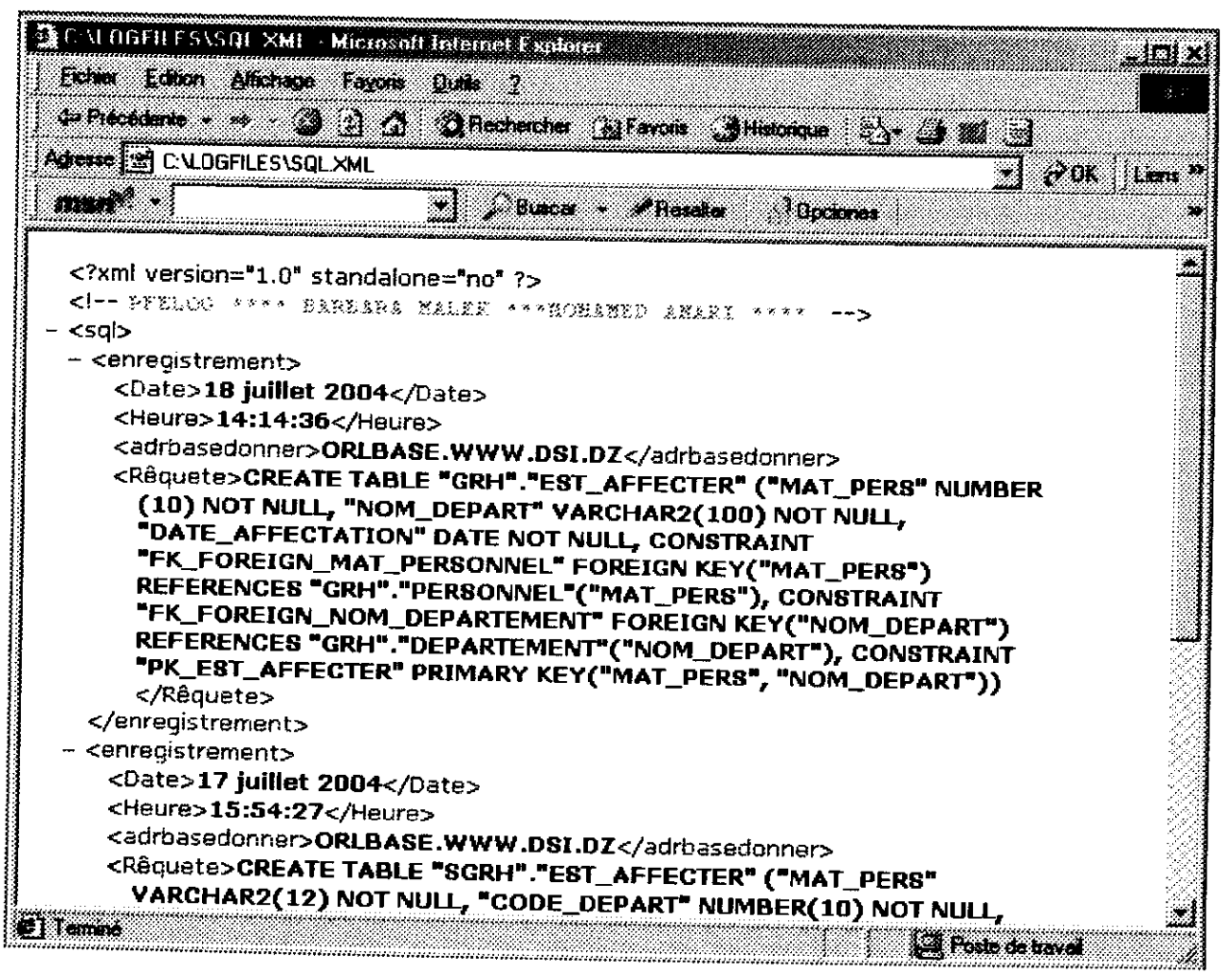


Fig-09: Format en XML du fichier sql.log

La figure suivante nous montre la table en XSD du fichier sql.xml qui est généré par XML de façon automatique :

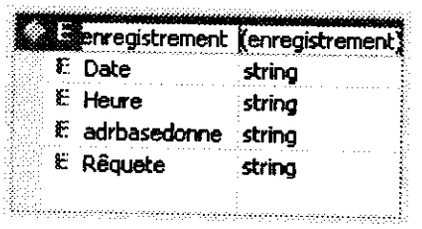


Fig-10: Table XSD du fichier sql.xml

Catégorie 2 : « Fichier sqlnet.xml»

La couche génération retourne le fichier XML correspondant au fichier SQLNET.LOG présenté sous la forme d'enregistrement dans la figure suivante comme suit :

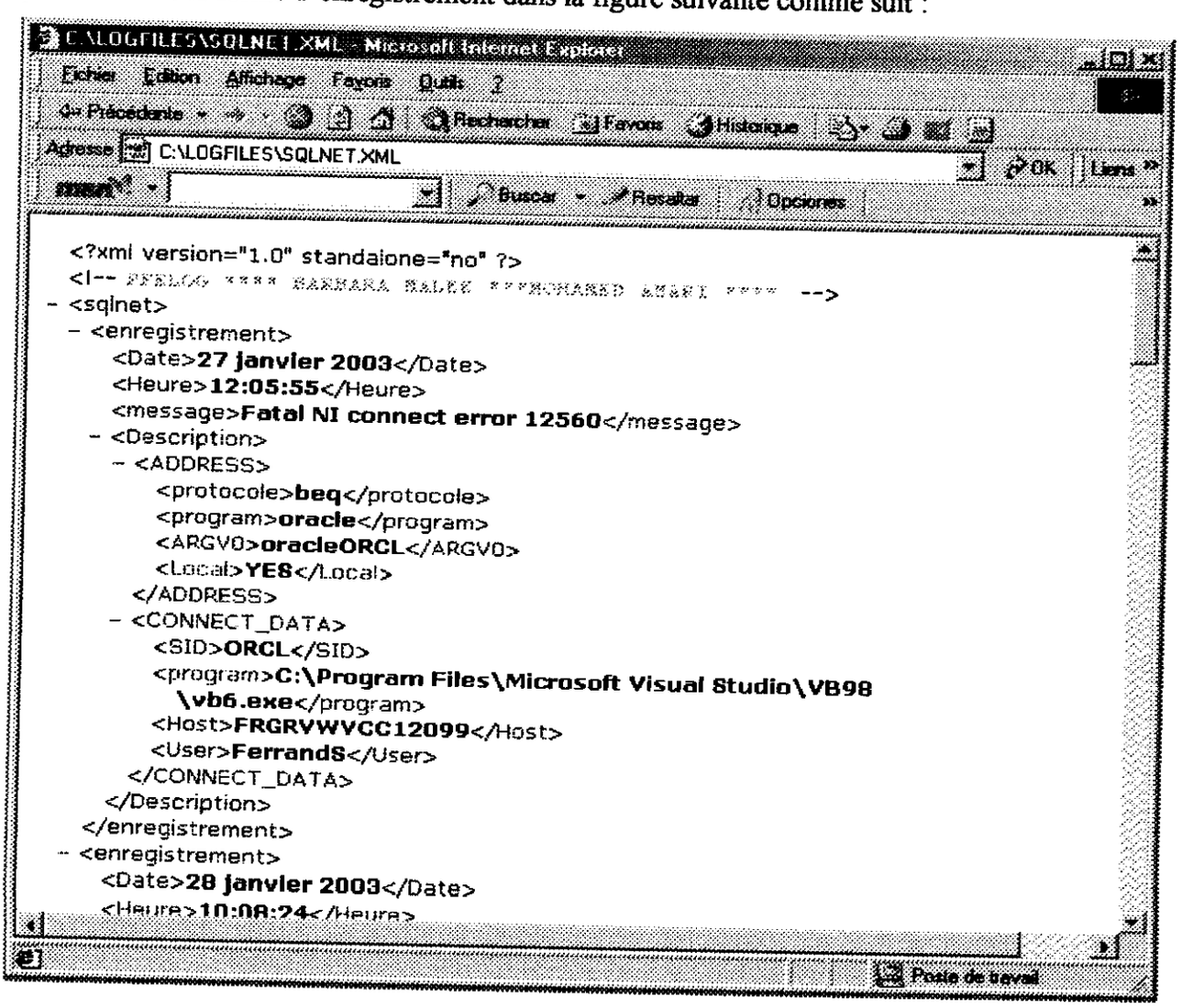


Fig-11 : Format en XML du fichier sqlnet.log

La génération de la table XSD du fichier Sqlnet.xml nous donne la figure suivante qui nous permet de connaître la structure du fichier en montrent les principales tables du fichier sqlnet.xml, ses sous tables ainsi que les champs des tables :

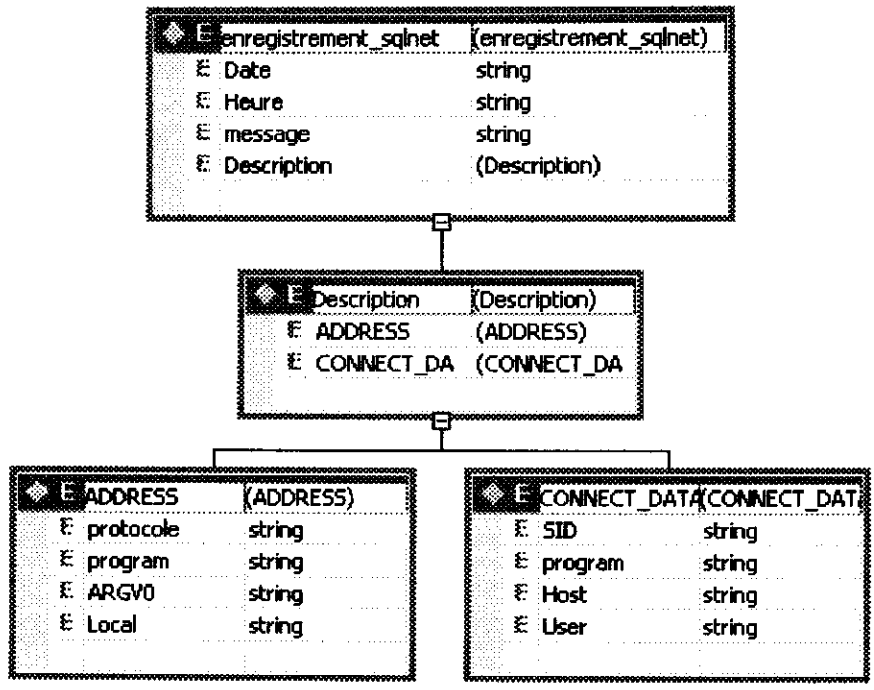


Fig-12 : Table XSD du fichier sqlnet.xml

Catégorie 3 : « Fichier Alert_oracl.xml & Alert_orclbase.xml»

Ces deux fichiers ont la même forme qui figure ci-après :

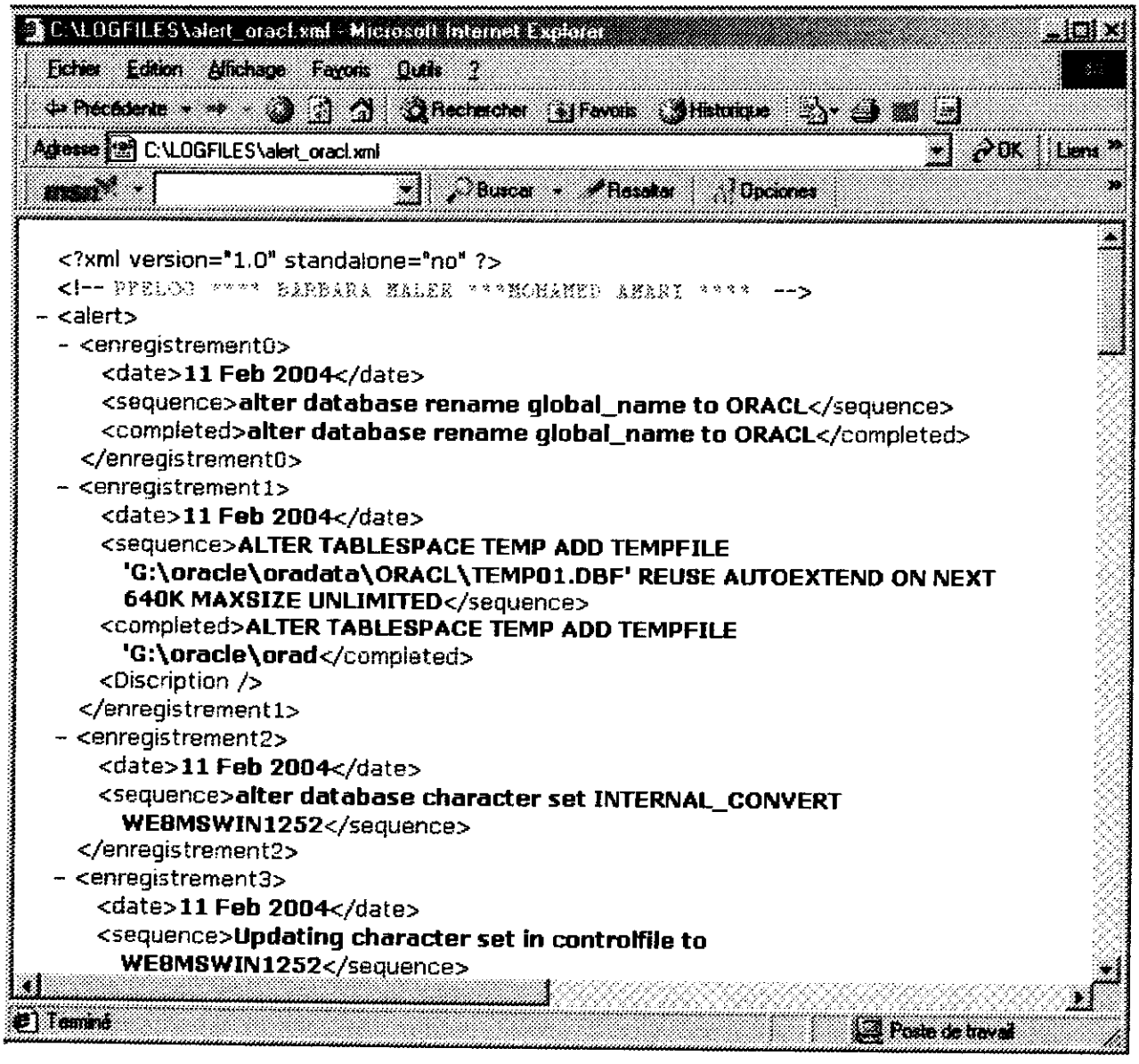


Fig-13: Format en XML du fichier Alert_oracl.log

Catégorie 4 : « Fichier Listener.xml »

La couche génération du fichier Listener.log a qui nous avons définis sa structure et les champs qu'il doit contenir nous est présenté aussi sous la forme d'enregistrement comme suit :

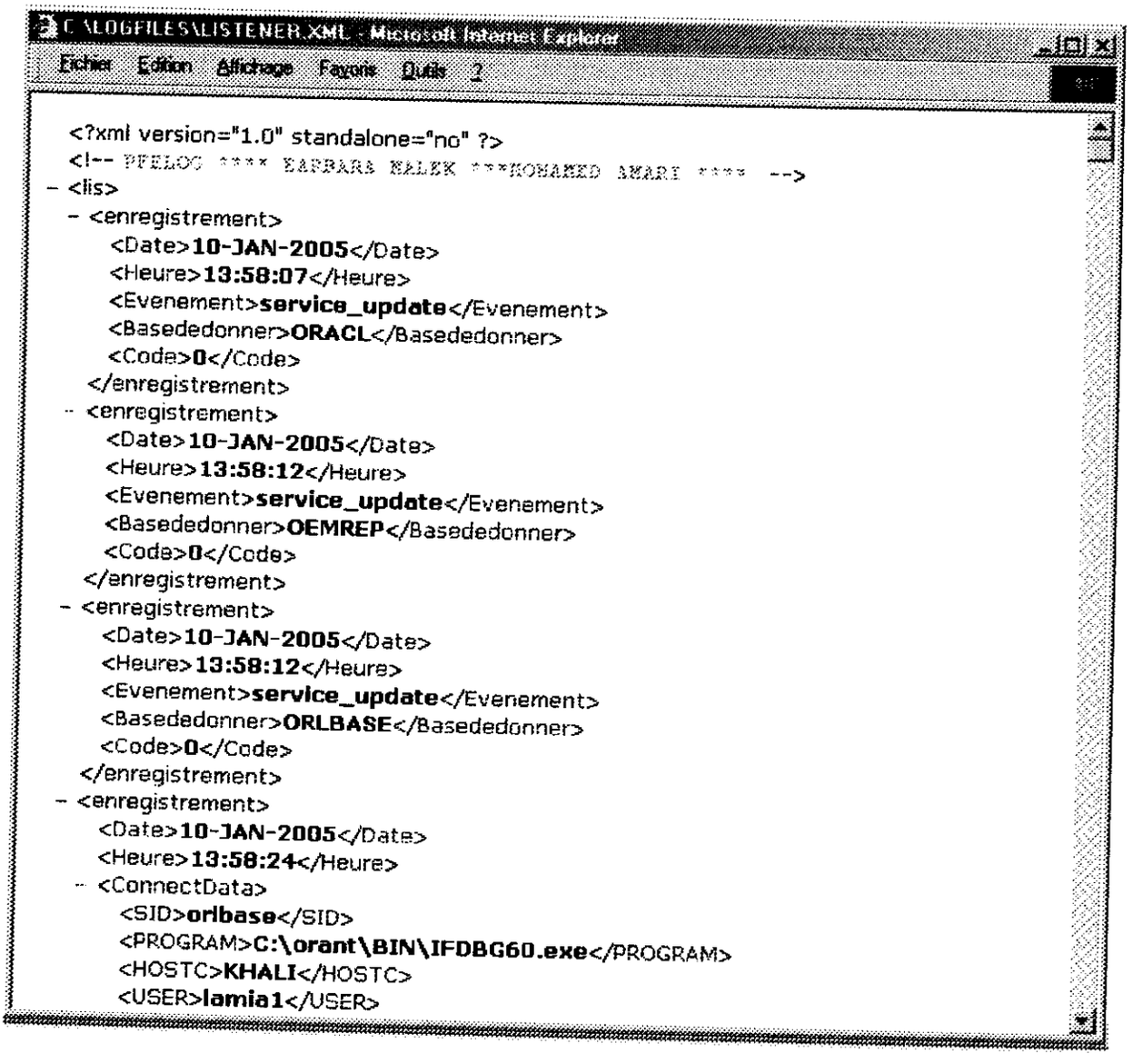


Fig-14: Format en XML du fichier Listener.log

De plus la génération de la table XSD du fichier Listener.xml nous donne la figure suivante qui nous permet de connaître en détail la structure du fichier en montrons les principales tables du fichier, ses sous tables ainsi que les champs des tables :

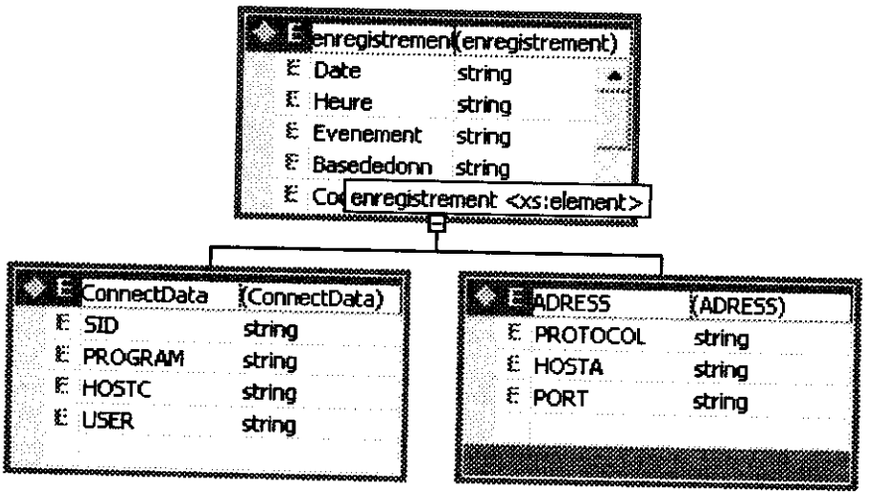


Fig-15 : Table XSD du fichier Listener.log

3. Exploitation Des fichiers logs

Cette phase est cruciale car elle permet d'interroger la base XML constituée lors des étapes précédentes. Pour cela nous devons proposer à l'administrateur de la base de données : plusieurs critères de recherche, des statistiques sur le trafic ainsi qu'un certain nombre d'indicateurs afin de lui permettre de diagnostiquer la situation, de détecter les dysfonctionnements et erreurs et de les signaler au temps opportun.

Pour disposer d'informations de synthèse, il est nécessaire d'effectuer une recherche efficace sur les fichiers logs générés.

3.1. La Recherche

Nous avons cité dans les paragraphes précédents que les fichiers log contiennent toutes les informations concernant les opérations effectuées sur la base de données. L'interrogation des données de différents champs des fichiers log déjà générés peuvent offrir une meilleure exploitation de ces données car permettant de reconstituer les événements qui se sont réellement produits et d'avoir une vision nette sur toutes les opérations effectuées sur la base de données.

Une recherche peut s'effectuer sur les champs suivants :

- La Date et l'heure.
- Le Protocole utilisé.
- Les ports sources et destinations des différents services de communication utilisés.
- Le type des la requêtes.
- Le nom des tables.
- Le nom de la base de données.
- Les types d'événements.

Cette phase consiste à effectuer des recherches sur les fichiers générés selon notre choix, par exemple :

Une recherche sur les Dates des événements dans une date précise, avant cette date, après cette date, ou bien entre deux dates différentes.

Effectuer des recherches sur les types de requêtes SQL utilisées, ainsi sur les tables interrogées.

Effectuer des recherches sur les types d'événements produits dans la base de données comme modification ou MAJ ou d'autres.

Les résultas des recherches sont affichées en XML et dans des tables ordonnées.

4. La Phase de Rafraîchissement :

Cette phase est dédiée à l'actualisation des fichiers logs générés. Etant donné qu'il y a des mises à jour dans les fichiers logs courants du SGBD, il est nécessaire de répercuter ces mises à jours sur les fichiers XML correspondants à ces derniers. Ces mises à jour consistent en l'ajout de nouveaux enregistrements relatifs à des événements qui se sont produits dans la base de données.

Cette mise à jour doit se faire de manière automatique et transparente à l'utilisateur, en effet, le principe consiste à comparer le dernier enregistrement du fichier log déjà généré et celui du fichier log mis à jour. Si ces enregistrements sont différents cela signifie qu'il y a eu ajout d'un nouvel enregistrement qui devra être converti au format XML est ajouté à la fin du fichiers XML déjà généré. Cette phase nous permet de rendre la phase d'exploitation plus efficace car elle permet de manipuler des informations relatives à la situation réelle du système sans recourir à la génération du fichier de nouveau.

5. Association d'une feuille XSL au document XML

Etant donné que la norme XML ne permet pas de décrire le format d'affichage ou d'impression d'un document, le W3C l'a complété par un langage de mise en page XSL (eXtended Style Language). Une description XSL (écrit en XML) s'applique à un document comme le fait sa DTD, mais pour définir l'interprétation physique (Affichage par exemple) des divers éléments. XSL est un complément « feuille de style » d'XML comme CSS est un complément feuille de style pour HTML.

Conclusion

A travers ce chapitre, nous avons présenté l'approche XML adoptée pour la conception de notre application. Nous avons décrit les différentes étapes que nous avons développées ainsi que leurs principe de fonctionnement afin d'homogénéiser et exploiter les différent fichiers logs générés par le SGBD Oracle 9i, en un format XML standard.

Nous aborderons dans le chapitre suivant l'aspect Implémentation, ce dernier portera sur les outils utilisés et la présentation de l'application développée.

CHAPITRE VI

IMPLEMENTATION

Introduction

Dans ce chapitre nous allons aborder l'implémentation de notre système. Ceci se traduit par le développement d'une application qui génère des documents XML à partir des fichiers log, d'autre part, cette application permet la visualisation et l'exploitation des fichiers XML résultants à travers des recherches.

1. Outils utilisés

Nous avons utilisé comme environnement de développement pour la réalisation du système le *Visual Studio.Net 2003*, il offre tous les avantages de la programmation orientée objet tel que la modularité et l'héritage tout en nous donnent la possibilité d'accéder aux services de la plateforme .NET (.NET FrameWork).

Nous avons utilisées plusieurs bibliothèques de classes offertes par la plateforme .NET tel que :

- ***System.Xml :***

L'espace de noms System.Xml fournit une prise en charge standard du traitement XML. Les standards pris en charge sont :

• XML 1.0 - http://www.w3.org/TR/1998/REC-xml-19980210 - y compris prise en charge DTD.

• Espaces de noms XML - http://www.w3.org/TR/REC-xml-names/ - tant au niveau des flux que de DOM.

• Schémas XSD - http://www.w3.org/2001/XMLSchema

• Expressions XPath - http://www.w3.org/TR/xpath

• Transformations XSLT - http://www.w3.org/TR/xslt

• Noyau DOM Niveau 1 - http://www.w3.org/TR/REC-DOM-Level-1/

• Noyau DOM Niveau 2 - http://www.w3.org/TR/REC-DOM-Level-2/

- ***System.Xml.Schema :***

L'espace de noms System.Xml.Schema contient les classes XML qui assurent la prise en charge standard des schémas XSD (XML Schema Definition). Les standards pris en charge sont :

• Prise en charge de XML Schemas for Structures http://www.w3.org/TR/xmlschema-1/ - pour schémas de mappage et de validation.

• Schémas XML for Data Types - http://www.w3.org/TR/xmlschema-2/ - prend en charge les types de données pour les définitions XML Schema (XSD).

- ***System.Data***

Se compose principalement des classes qui constituent l'architecture ADO.NET. L'architecture ADO.NET permet de construire des composants qui gèrent efficacement les données provenant de plusieurs sources de données. Dans un scénario déconnecté (tel qu'Internet), ADO.NET fournit les outils permettant de demander, mettre à jour et rapprocher les données de systèmes à plusieurs couches. L'architecture ADO.NET est également implémentée dans les applications clientes, telles que Windows Forms ou les pages HTML créées par ASP.NET. Elle comprend aussi la classe **DataSet** qui représente les tables mutiples et leur relations.

- ***System.IO***

L'espace de noms System.IO contient des types qui permettent la lecture et l'écriture dans des fichiers et des flux de données, ainsi que des types qui permettent la prise en charge de fichiers et de répertoires de base.

- ***System.Windows.Forms*** :

L'espace de noms **System.Windows.Forms** contient des classes permettant la création d'applications Windows qui tirent parti des fonctionnalités d'interface utilisateur évoluées disponibles dans le système d'exploitation Microsoft Windows.

Les classes dans cet espace de noms se répartissent dans les catégories suivantes :

- Control, User Control, and Form.
- Controls.
- Components.
- Common Dialog Boxes.

Toutes ces bibliothèques de classes et d'autres ont été utilisées avec le langage de programmation **C#.**

1.1. Le langage C#

Dans le contexte de notre travail, le langage C# s'impose comme un langage de programmation capable de nous mener à nos objectifs. Le langage C# offre la portabilité du code, XML la portabilité des données. On pourra donc générer des documents XML portables et accéder à leurs données à l'aide des classes C# portables elles aussi. XML permet un marquage spécifique des données et en plus il les stockent sous format purement ASCII qui est portables sur tous les machines et architectures. C# et XML sont donc complémentaires.

Dans C# il existe des programmes d'analyse syntaxique d'XML écrits en C#. Des documents XML peuvent être facilement analysés par des objets C#.

2. Présentation de L'application

L'interface principale de notre application comporte trois barres, la barre de titre, la barre de menu et la barre d'outils, comme la montre la figure ci-dessous :

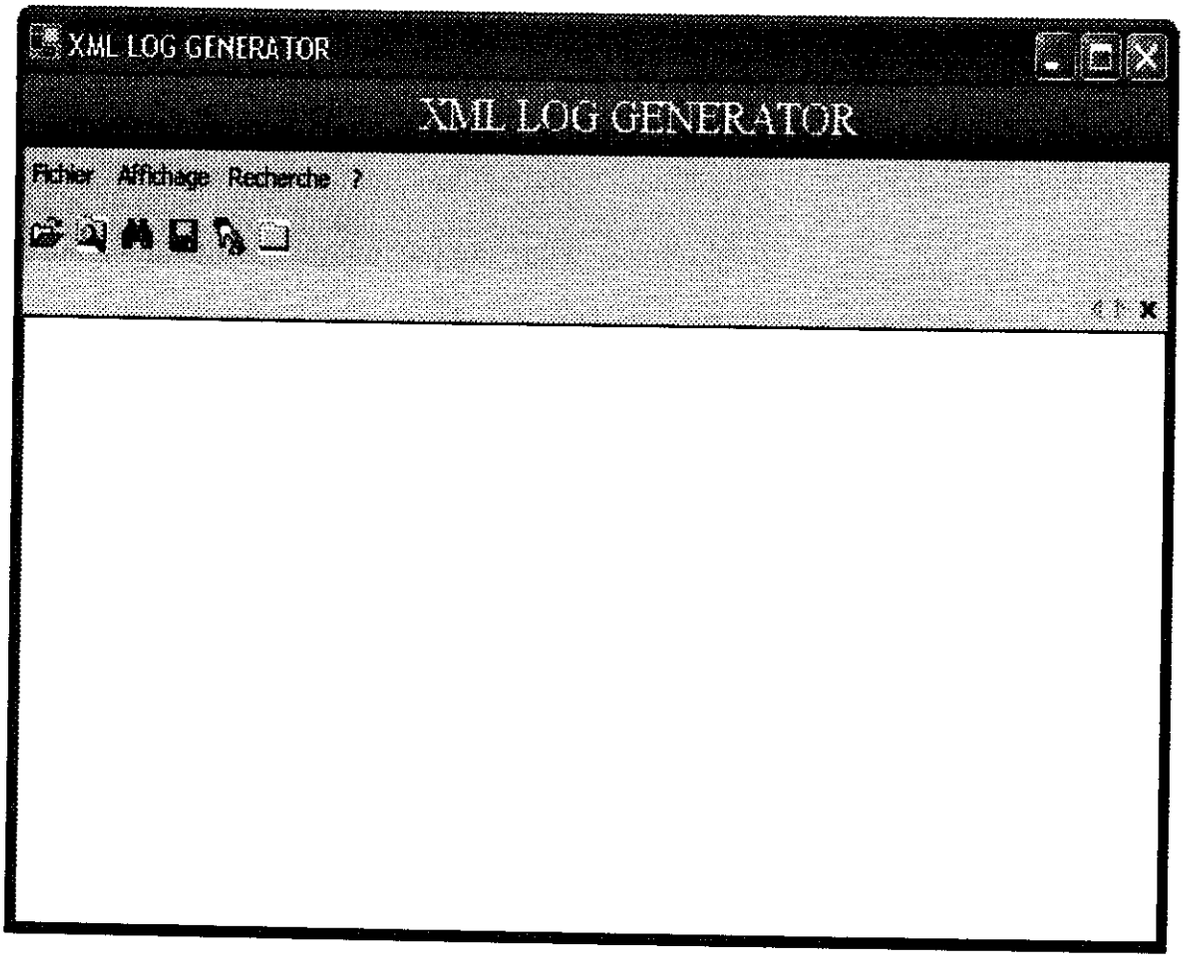


Fig-16: Interface principale de l'application.

Chacune de ces barres se compose de ce qui suit :

La Barre de titre : Elle contient :

- Le logo de l'application.
- Le nom de l'application : XML_LOG_GENERATOR.
- Le nom du Fichier log dans le cas ou l'utilisateur l'agrandit.

- Des icônes qui permettent de manipuler la fenêtre du document :

 Fermer provisoirement.

 Agrandir.  Réduire.

 Fermer définitivement.

La barre de menu : donne accès a toutes les fonctionnalités de l'application. Elle contient les menus de l'application. Ils représentent les opérations offertes par le système. Chaque menu contient un ensemble d'options avec des raccourcis clavier. L'interface de XML_LOG_GENERATOR comporte les menus suivants :

- Le menu Fichier : Il contient les options suivantes :
  - Option *Ouvrir un Fichier Log* : Permet de spécifier un ou plusieurs Fichiers logs à ouvrir à partir d'un emplacement afin de le générer en XML.
  - Option *Ouvrir un* Fichier XML: Permet à l'utilisateur d'ouvrir un Fichier XML crée précédemment.
  - Option enregistrer : Permet à l'utilisateur d'enregistrer le fichier ou une recherche.
  - Option *Quitter* : Permet à l'utilisateur de quitter d'une façon définitive l'application.

- Le Menu Affichage : permet d'afficher les fichiers logs.
- Le Menu Recherche : il sert à effecteur les recherches selon le choix sur n'importe quels fichiers XML.

La barre d'outil : contient les boutons qui donne accès a toutes les fonctionnalités de l'application.

Les fichiers logs générés par le SGBD sont sauvegardés dans un répertoire nommé **LOGFILES,** l'ouverture de ce répertoire et le choix du fichier log est disponible à partir de la fenêtre représentée par la figure ci-après :

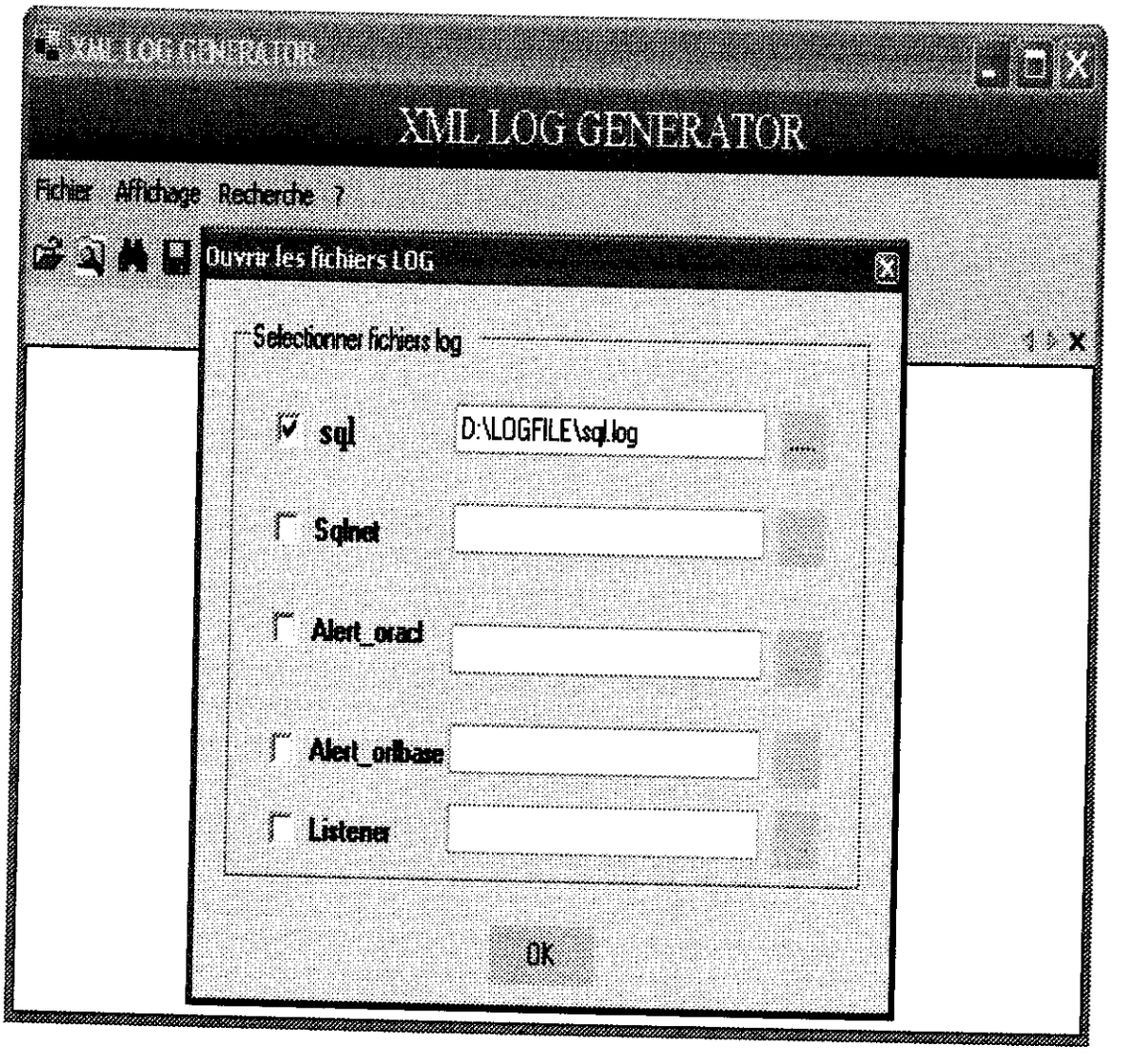


Fig-17 : L'ouverture des fichiers logs.

Les fichiers sélectionnés seront ouverts dans une zone de texte avec la possibilité de visualiser chacun des fichiers au format log ou xml comme suit :

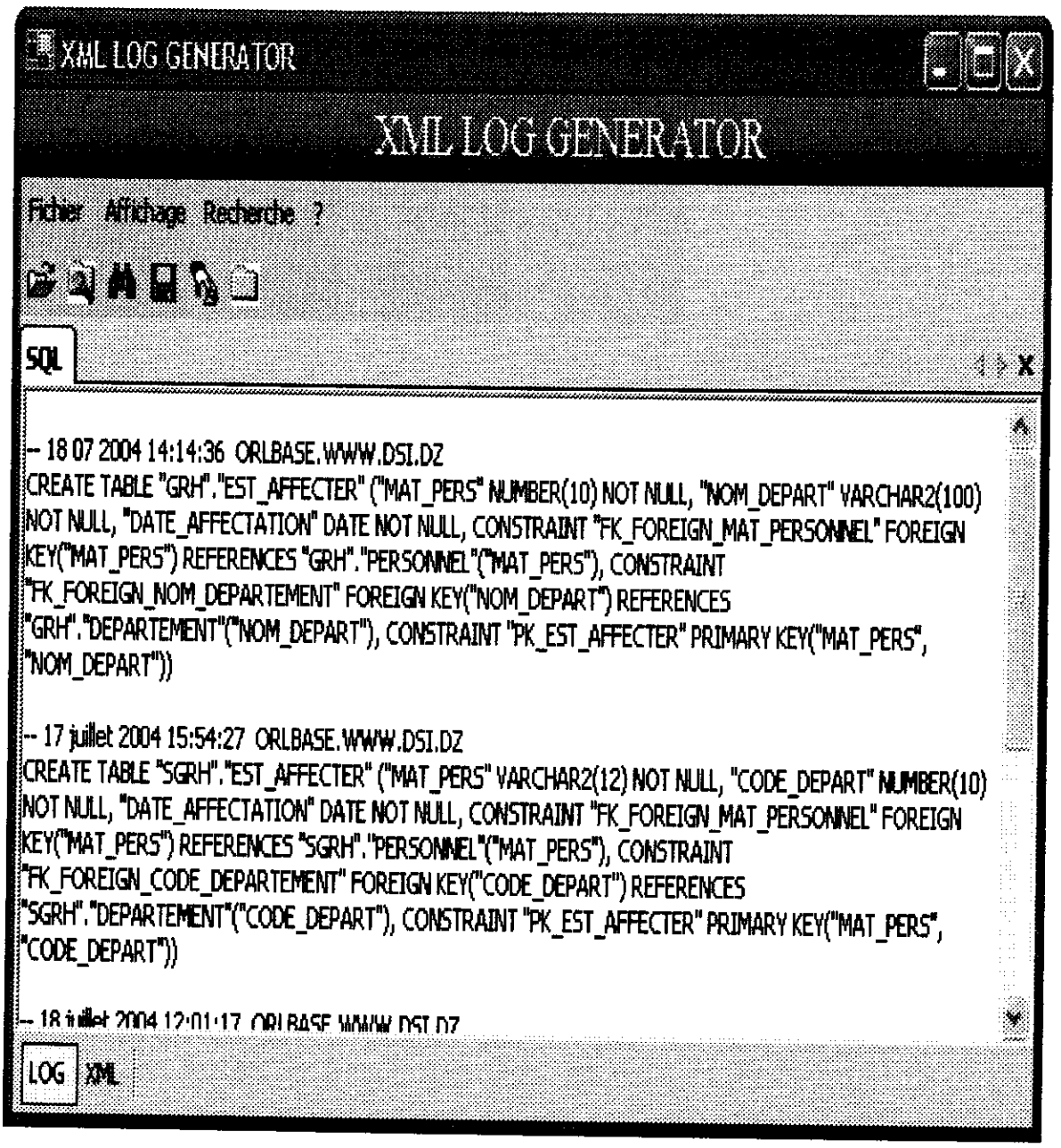


Fig-18 : L'ouverture du fichier logs sql.log.

Après avoir traité le fichier log, les informations significatives extraites seront insérées dans le nouveau fichier XML. Comme la montre la figure suivante :

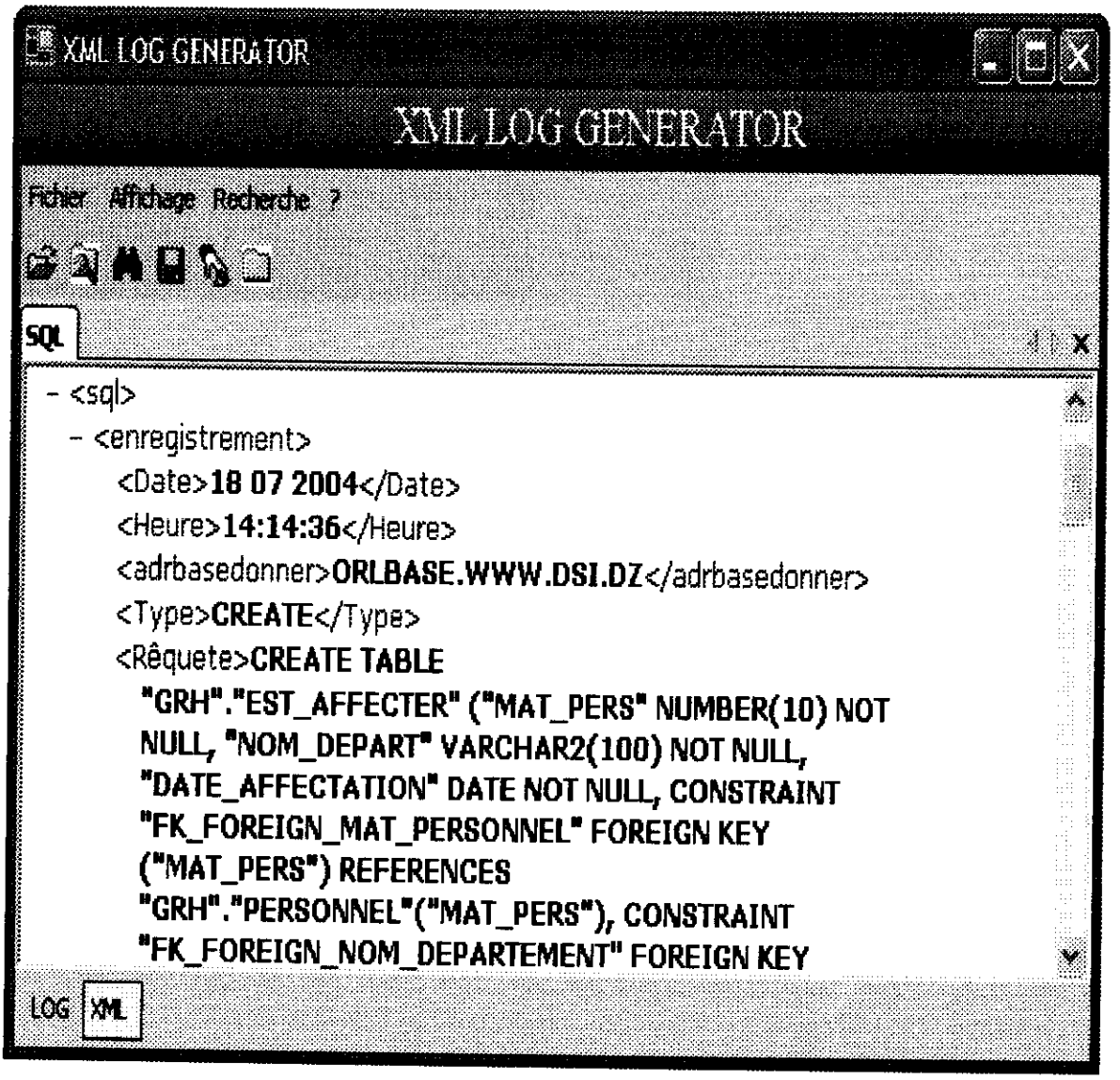


Fig-19 : La conversion du fichier log sql.log

Nous pouvons aussi faire plusieurs conversions de fichiers logs déjà ouvert et les visualiser sur une même fenêtre comme la montre la figure suivante :

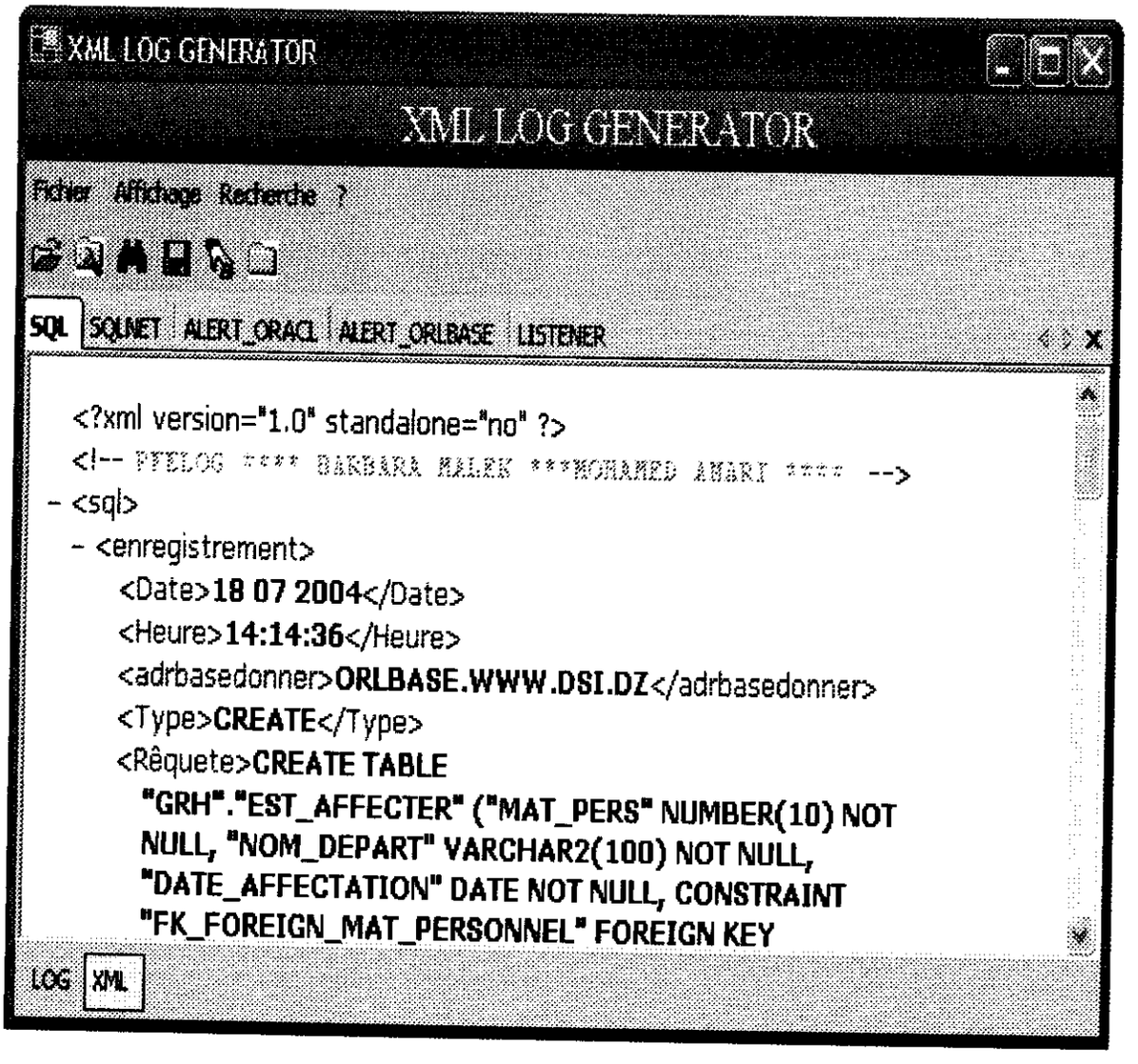


Fig-20 : La conversion de plusieurs fichiers logs.

L'administrateur à l'opportunité de chercher, visualiser, ou de consulter seulement le fichier XML qui est auto descriptifs en l'ouvrant soit dans l'application comme on la vu dans la figure précédente ou de l'ouvrir avec Internet Explorer comme la montre la figure ci-après :

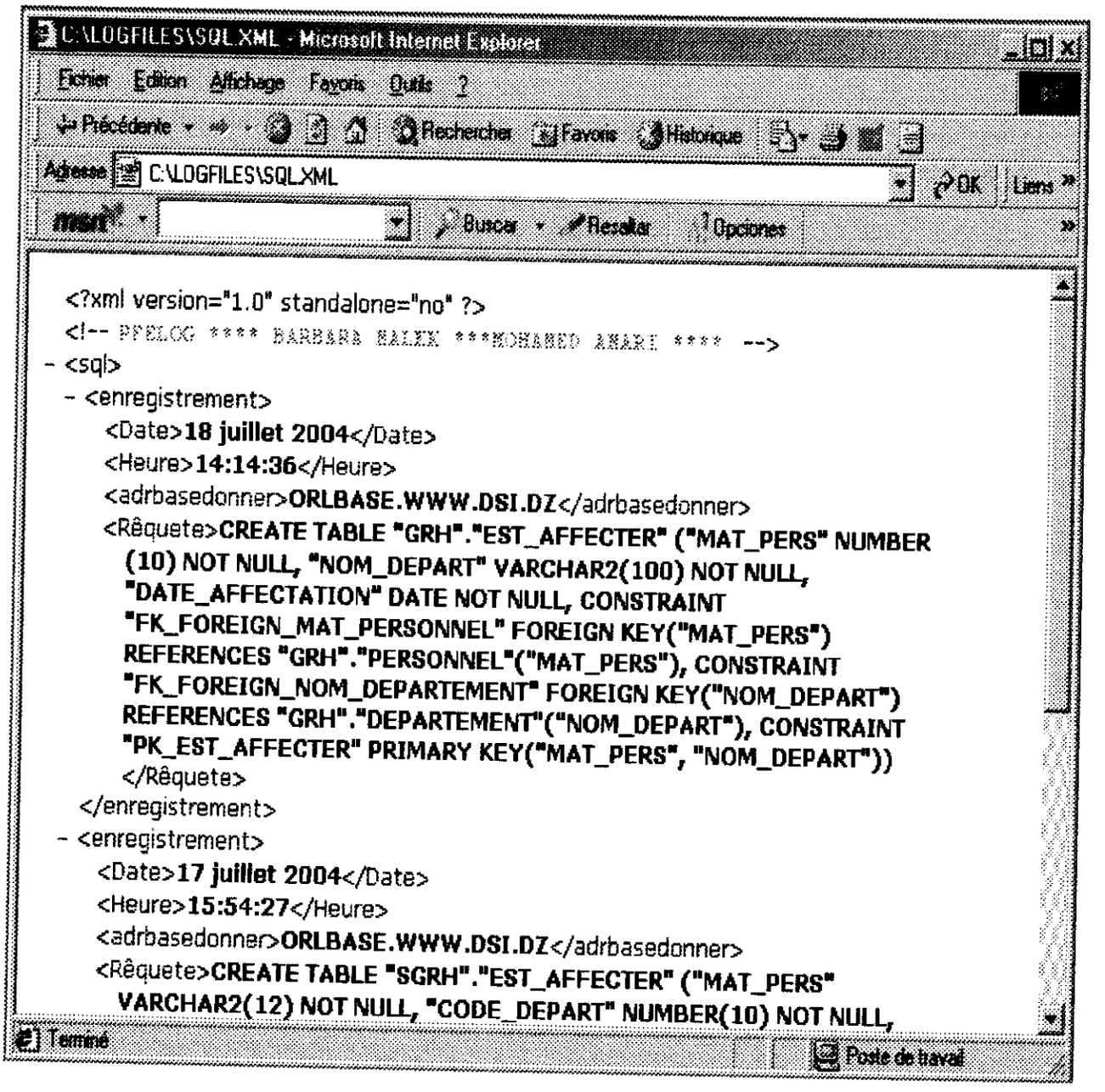


Fig-21 : Le code XML du fichier log sélectionné.

L'administrateur dispose de la possibilité d'effectuer des recherches selon plusieurs critères tel que : le type de fichier, le type d'informations à rechercher à savoir la date, l'heure, type d'événement, adresse IP, le port, l'utilisateur ainsi que le protocole. C'est-à-dire qu'il a le choix des critères d'exploiter ces informations comme la montre la figure suivante :

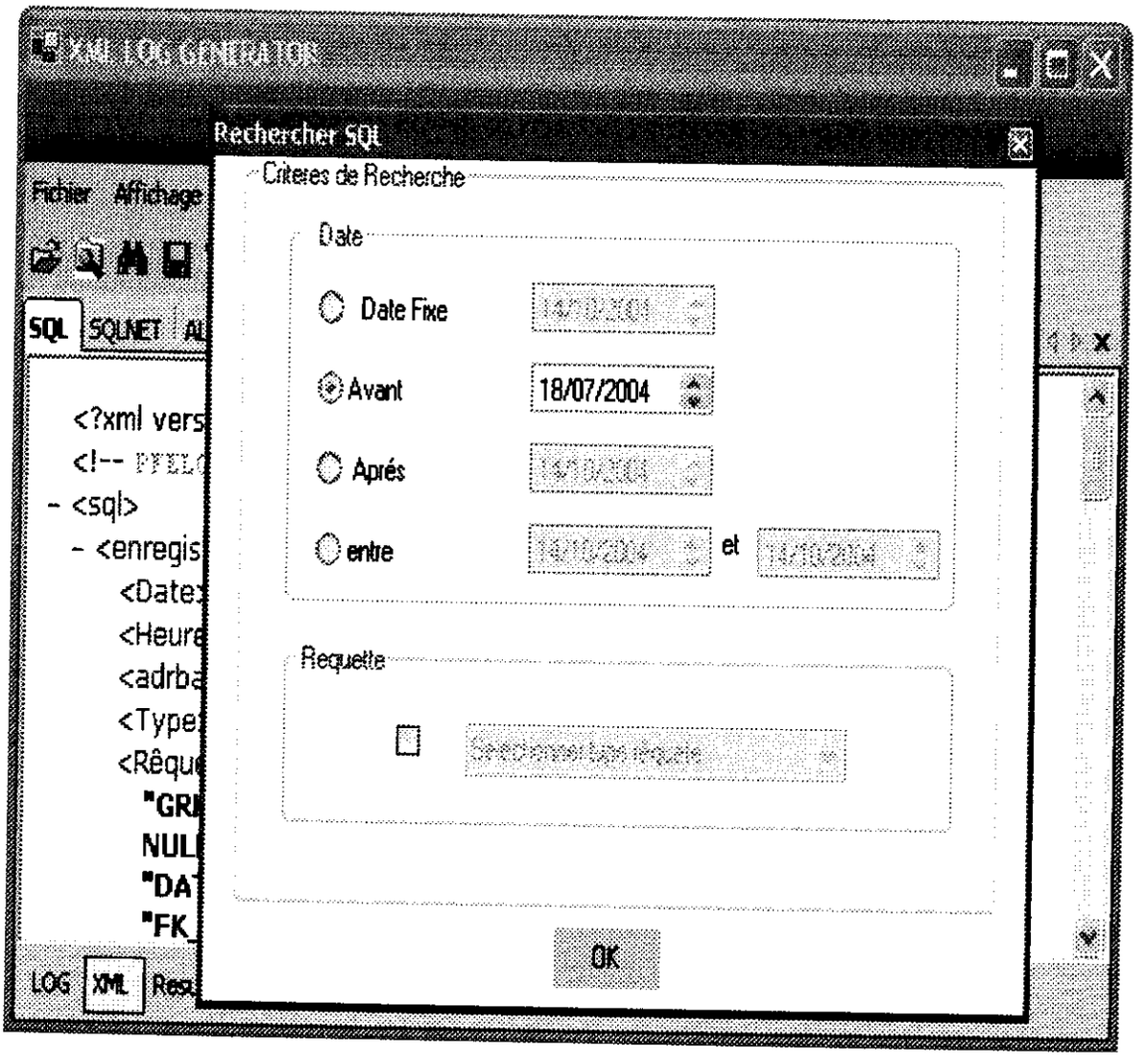


Fig-22 : Interface de recherche.

Le résultat de la recherche effectuée est affiché soit en XML ou bien sous forme tabulaire. La figure suivante nous montre l'affichage du résultat de la recherche en forme tabulaire :

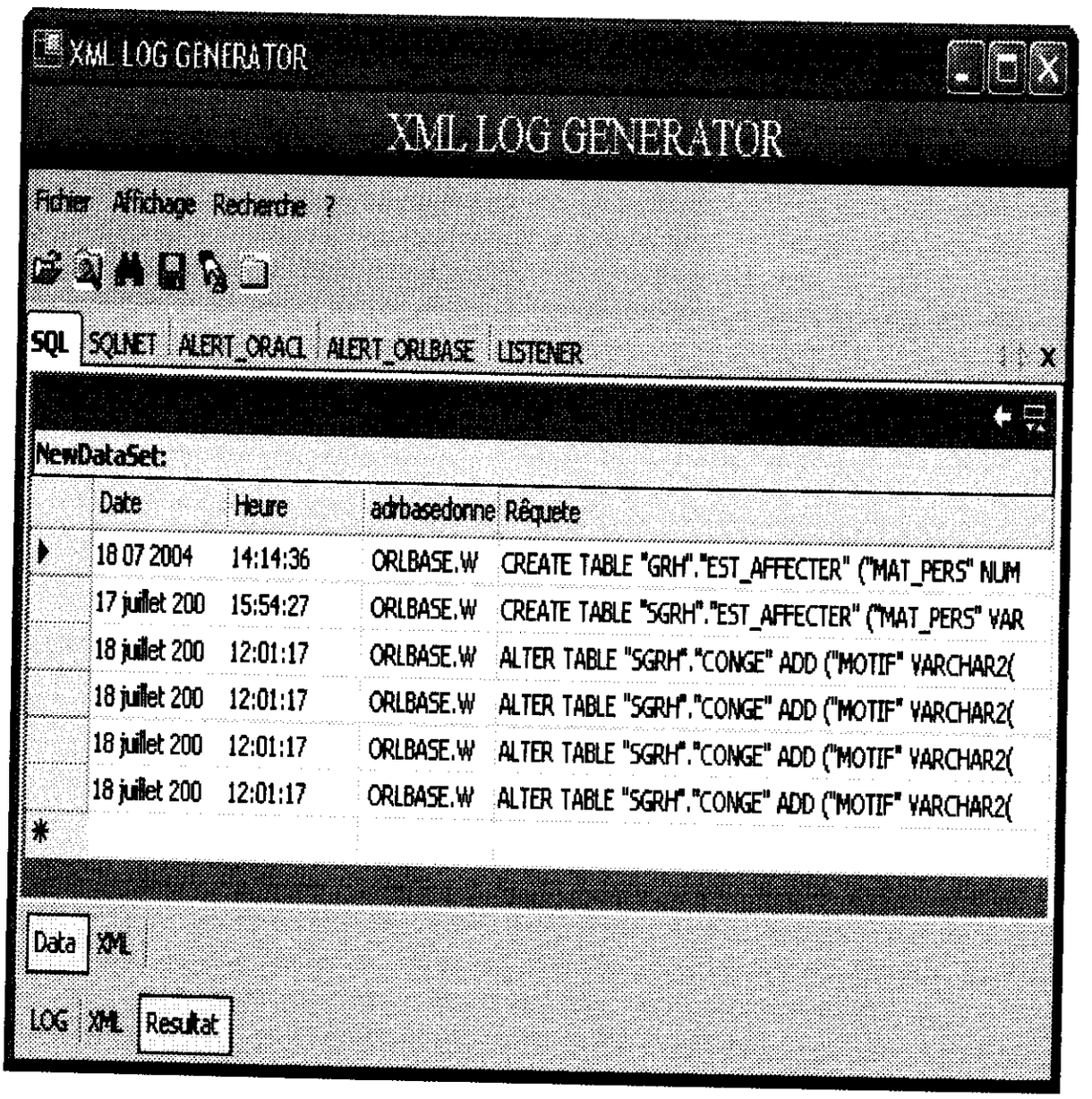


Fig-23 : Résultat de la recherche en forme Tabulaire.

La figure ci-après nous montre l'affichage du résultat de la recherche en XML :

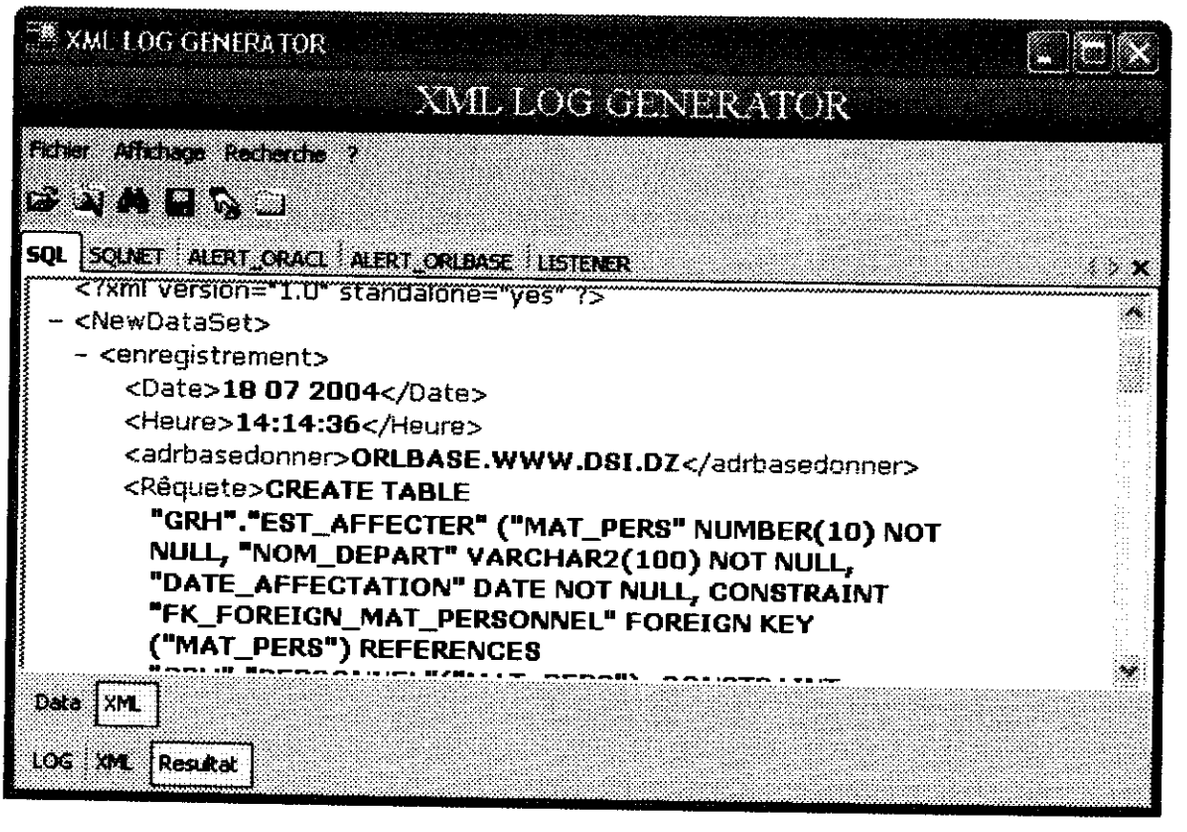


Fig-24 : Résultat de la recherche en XML.

CONCLUSION GENERALE

Conclusion Générale

Le point idéal de tout système est l'enregistrement des activités, événements, et toute autre action dans des fichiers logs qui constituent une base d'informations très puissante, car ils contiennent les empreintes des attaquants et indiquent les menaces et les attaques en cours.

Cependant, il est parfois difficile pour un administrateur d'interpréter les informations contenues dans un fichier log pour mener à bien ses recherches, établir des statistiques, et d'en déduire quelles sont les failles et erreurs en temps réels si possible. En effet, les fichiers logs sont sous formats différents et contiennent un grand nombre d'enregistrements.

Pour aider l'administrateur à prendre les décisions adéquates lors de l'investigation, il est impératif de le guider dans l'interprétation des fichiers logs. Afin de l'aider à prendre des décisions immédiates, nous avons présenté un modeste travail sur la génération d'un format standard à partir des fichiers logs, pour faciliter l'échange, l'interprétation, et l'exploitation de ces fichiers. Pour arriver à ce but, nous avons utilisé la norme XML.

Pour ce faire, notre application se base sur le choix des enregistrements les plus pertinents qui facilitent la tâche d'exploitation par l'administrateur, en minimisant le temps de consultations des fichiers XML générés à partir des fichiers log et en se basant sur des recherches effectuées sur ces fichiers selon des critères de recherche judicieusement choisis. Les résultats de ces recherches doivent être exploités par des experts qui ont un savoir sur les vulnérabilités des systèmes ainsi que sur les protocoles de communications et l'administration des bases de données.
Naturellement, la norme XML est plus souple à utiliser, car elle sert à stocker des données à long terme et permet d'extraire des documents lisibles et compréhensibles par l'homme.

Comme perspectives, nous proposons dans un premier temps de compléter le prototype développé et dans un deuxième temps de sécuriser les documents XML générés car ils sont

auto-descriptifs ce qui facilite leur lecture et leur modifications. Pour pallier ces difficultés, plusieurs propositions peuvent être envisagées :

- XML Signature : qui consiste à signer tout type de données. **[W3C 05]**
- XML Encryption : pour chiffrer les flux XML sans imposer des méthodes. **[W3C 05]**

Nous espérons que tout au long de cette étude, nos idées sur les différentes notions ont été suffisamment détaillées et seront utiles pour les futurs étudiants, et nous tenons à souligner qu'il reste une ouverture à des améliorations futures.

BIBLIOGRAPHIE

Bibliographie

[ARM 01] : Armée de terre -France-, Cours par correspondance préparatoire à l'EA2/FS/E5 du BSTAT [La sécurité de systèmes d'information] » Mars 2001

[DES 00] : Guillaume Desgeorge, « La sécurité des réseaux », 2000.

[MCC 01] : Stuart McClure, Joël Scamberay, George Kurtz, Halte aux Hackers « Sécurité réseau : secret et solution », Deuxièmes édition, Année 2001

[MAR 04]: Aurélien Marcon, Benjamin Fabrejon, La sécurité des réseaux : Le firewall

[MTA 05] : mesure de trafic et analyse d'audience pour l'audit de sites internet. http://www.stat-e-stik.com.

[TAS 02] Thèse Analyse statistique : pour faire de son site un hit ! Données, statistiques et analyse, 2002.

[WBT 04]: http://www.WebTrends.com, 2004

[AES 04] : Articles et études : L'utilisation des «logs files» en évaluation et en reconception : intérêts et limites, 2004

[DIC 05] : http://www.dicodunet.com/definitions/hebergement/fichier-log.htm, 2005

[SEC 01] :http://securis.info/securis/ker/index.html, 2001

[DEF 05] : http://Definition fichiers logs.htm, 2005

[JOU 05] : http://www.journaldunet.com/encyclopedie/definition/968/51/20/fichier_log.shtml, 2005.

[ADC 05] : http://www.adcom.fr/adcom/adcom.htm. 2005

[ENS 01] : http://www.enserb.u-bordeaux.fr, 2001

[ISS 96] : Introduction à la sécurité des systèmes Unix connectés à Internet, Pascal Brunox, Institut national polytechnique de Grenoble, 1996.

[SPY 04]: http://www.athena-gs.com/public/webspy,2004

[MBD 00] : Base de Données et Système de Gestion de Base de Données : Hervé MARTIN.

[SAM 04] : http://www.sam-mag.com/archives/cookies2.htm

[SQL 04] : Les petits papiers de SQLPro - SQL Server journal de transaction (log) - Club d'entraide des développeurs francophones.htm

[IMP 04] : Analyse du trafic, SQL Server, importation du fichier journal.htm

[ORA 09] : http://www.oracle.com.

[RDL 04] : http://ora7.free.fr/Les Redo Logs.htm

[ORL 04]: http://www.infini-fr.com/Sciences/Informatique/Bases_de_donnees/ Relationnel/ Oracle/index.htm

[RAH 03] : RAHIM.S et AFRIT.S, « APPORT XML POR SYSTEMES D'AIDES A L'INVESTIGATION POR LES FIREWALLS », 2003.

[W3C 05] : http://www.w3c.org/Xml, cours et articles, 2005

[ART 52] : Sylvain Bilodeau, http://www.a525g.com/index.html, 29-07-2003.

[ENO 05] : http://xquerydemo.enosyssoftware.com/.

[HIV 05] : http://support.x-hive.com/xquery/index.html

[XQL 05] : Jacques Le Maitre, XQuery, le langage d'interrogation de données XML, SIS, Université de Toulon et du Var

ANNEXE A

1 - Présentation de la plate-forme Microsoft .NET :

1.1- La plate-forme .NET :

Cette section présente les grandes lignes de l'architecture de la plate-forme .NET. La plate forme .NET se compose de plusieurs fonctionnalités et services de la base, comme l'illustre la figure A.1, l'un des objectifs de cette nouvelle plate-forme est de simplifier le développement Web.

1.2- Technologie de base de la plate-forme .NET :

Les technologies de base qui composent la plate-forme .NET sont les suivantes :

- **.NET** Framework :

  Cette technologie se fonde sur un nouveau Common Language Runtime. Celui-ci fournit un ensemble commun de services pour les projets crées avec Visuel Studio.NET, indépendamment du langage. Ces services fournissent des blocs modulaires de base pour les applications de tous types, utilisables à tous les niveaux des applications.
  Microsoft Visuel Basic, Microsoft Visuel C++, Microsoft Visuel C# et d'autres langage de programmation Microsoft ont été améliorés pour tirer profit de ces services.

- **.NET** Building Block Services :

  C'est un ensemble de services programmables distribués disponibles à le fois en ligne et hors connexion. Un service peut être appelé sur un ordinateur autonome non connecté à Internet ; il peut également être fourni par un serveur local fonctionnant au sein d'une entreprise. .NET Building Block Services peut être utilisé à partir de n'importe quelle plate-forme prenant en charge SOAP. Au nombre de ces services, citons : les calendrier, les annuaire, la notification et la messagerie...etc.

- Visuel Studio **.NET** :

  Constitue un environnement de développement de haut niveau, destiné à la création d'application sur le .NET Framwork. IL fournit des technologies clés afin de simplifier la création.

- .NET Entreprise Server :

  Les produits .NET Entreprise Server permettent une évolutivité, une fiabilité, une gestion de l'intégration. Ils offrent en outre un grand nombre de fonctionnalité par exemple : Microsoft SQL Server 2000, Microsoft Commerce Server 2000.

La figure suivante décrit les technologies de base de la plate forme Microsoft.NET

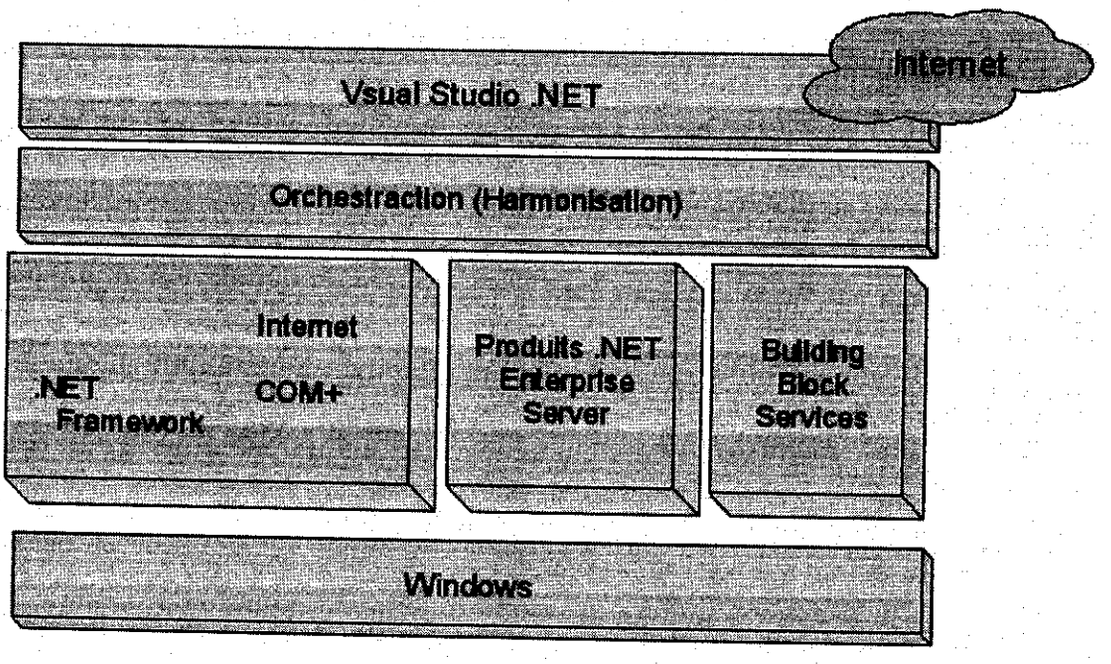


Figure A.1 Plate forme Microsoft .NET

1.3- Les avantages de la nouvelle technologie :

La plate-forme .NET offre les avantages suivants :

- Modèle de programmation cohérent et indépendant du langage, utilisable à tous les niveaux d'une application ;
- Interobabilité parfaite entre technologies ;
- Migration aisée à partir des technologies existantes ;
- Prise en charge totale des technologies Internet fondées sur des standards et indépendantes des plate-forme, telles que http, XML et SOAP.
- Grâce au Common Language Runtime, tous les langages compatibles avec la plate-forme .NET vont utiliser les mêmes fichiers d'exécution. Il n'est donc plus nécessaire de distribuer des bibliothèques d'exécution spécifiques à un seul langage, parce que les fichiers d'exécution .NET seront installés automatiquement dans les versions de Microsoft.

2 -Présentation du .NET Framework :

Le .NET Framework fournit tous les services communs nécessaire pour l'exécution de nos applications ; Ces services sont disponibles dans tous les langages compatibles avec .NET grâce à la spécification CLS (Common Language Specification)
Cette figure décrit ces services :

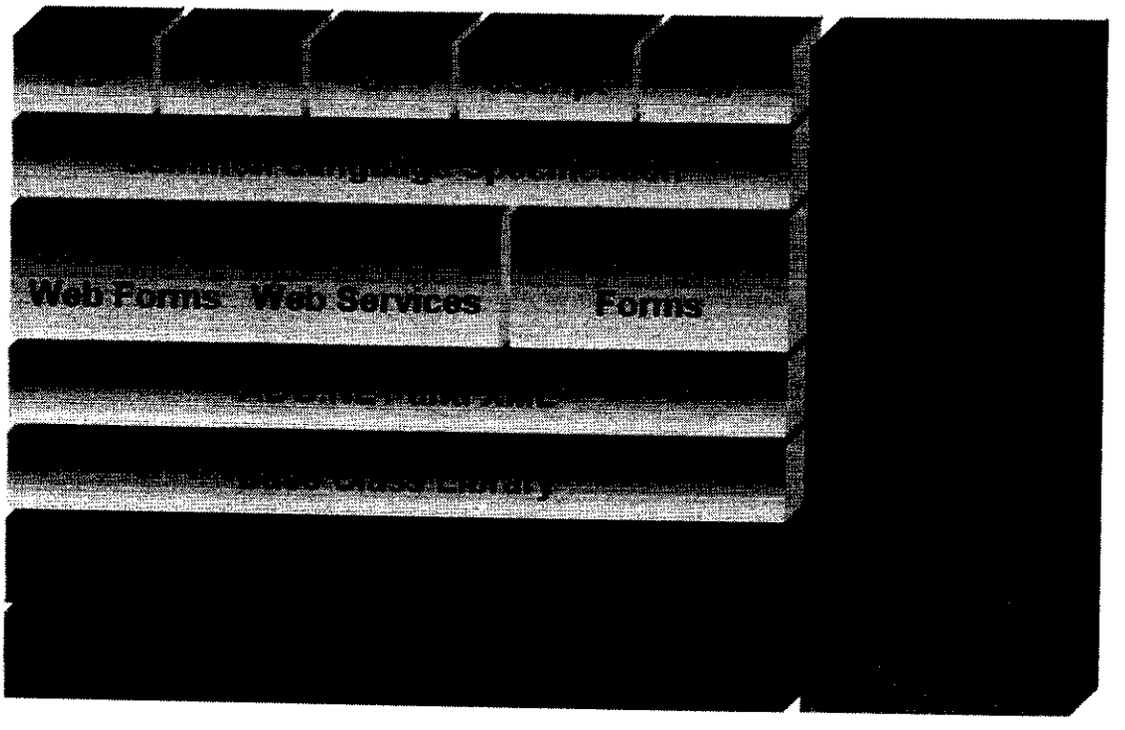


Figure A.2 Services communs pour l'exécution des applications

2.1 - Création de composants dans le .NET Framework :

Avant l'avènement de COM, les applications étaient des entités totalement séparées, sans aucune intégration ou très peu. Grâce à COM, nous pouvons intégrer des composants au sein d'une même application et à plusieurs, en exposant des interfaces communes. Dans le .NET Framework, les composants possèdent une base commune. Il n'est plus nécessaire d'écrire le code visant à permettre aux objets d'interagir directement les uns avec les autres.
Dans cet environnement le .NET Framework prend totalement en charge les classes, l'héritage, les méthodes, les propriétés, le polymorphisme, les constructeur et d'autres construction orientées objet.

2.2- Spécification CLS (Common Language Specification) :

La spécification CLS définit les standards communs que doivent respecter les langage et les développeurs pour que leurs composants et applications puissent être largement utilisés par d'autres langages compatibles avec le modèle .NET . Elle permet aux développeurs Visuel Basic .NET, Visuel C++ ou d'autres langages de créer des applications dans le cadre d'une équipe multi-langage, avec l'assurance que l'intégration des différents langages s'effectuera sans problème. CLS permet même aux développeurs Visuel Basic .NET ou Visuel C++ ...etc d'hériter de classes définies dans des langage différents.

2.3 Visuel Studio .NET:

Dans le .NET Framework, Visuel Studio .NET fournit les outils servant au développement rapide d'applications.

2.4 Les Langages du .NET Framework :

Cette section présente les langages que Microsoft fournit avec Visuel Studio .NET pour le .NET Framework.

- **Visuel Basic .NET** : Nouvelle version de Visuel Basic avec des innovations substantielles en terme de langage.
- **C#:** Il s'agit du premier langage moderne orienté composant.
- **Extensions C++:** offre plus de puissance et de contrôle.
- **J# .NET** : est un langage pour les développeurs java qui souhaitent créer des applications et des services pour le .NET .
- **Langages tiers :** divers langage tiers prennent en charge le .NET : COBOL, Pascal, SmallTalk... etc.

3 Présentation des composants .NET Framework :

Les composants du .NET Framework sont les suivants :

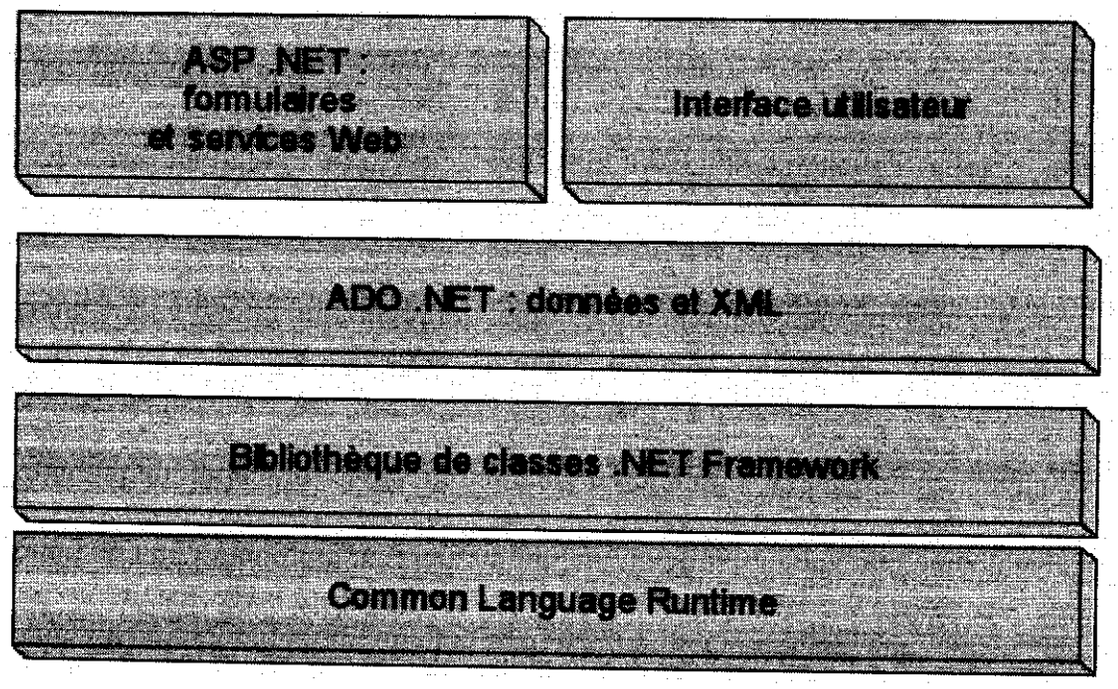


Figure A.3 Composants du .NET Framework

3.1- Common Langage Runtime :

Il simplifie le développement d'application, fournit un environnement d'exécution robuste et sécurisé, prend en charge plusieurs langages, simplifie le déploiement et la gestion des applications et offre un environnement géré

3.1.1- Composants du Common Langage Runtime :

Les fonctionnalités du Common Langage Runtime sont décrites dans la figure suivante :

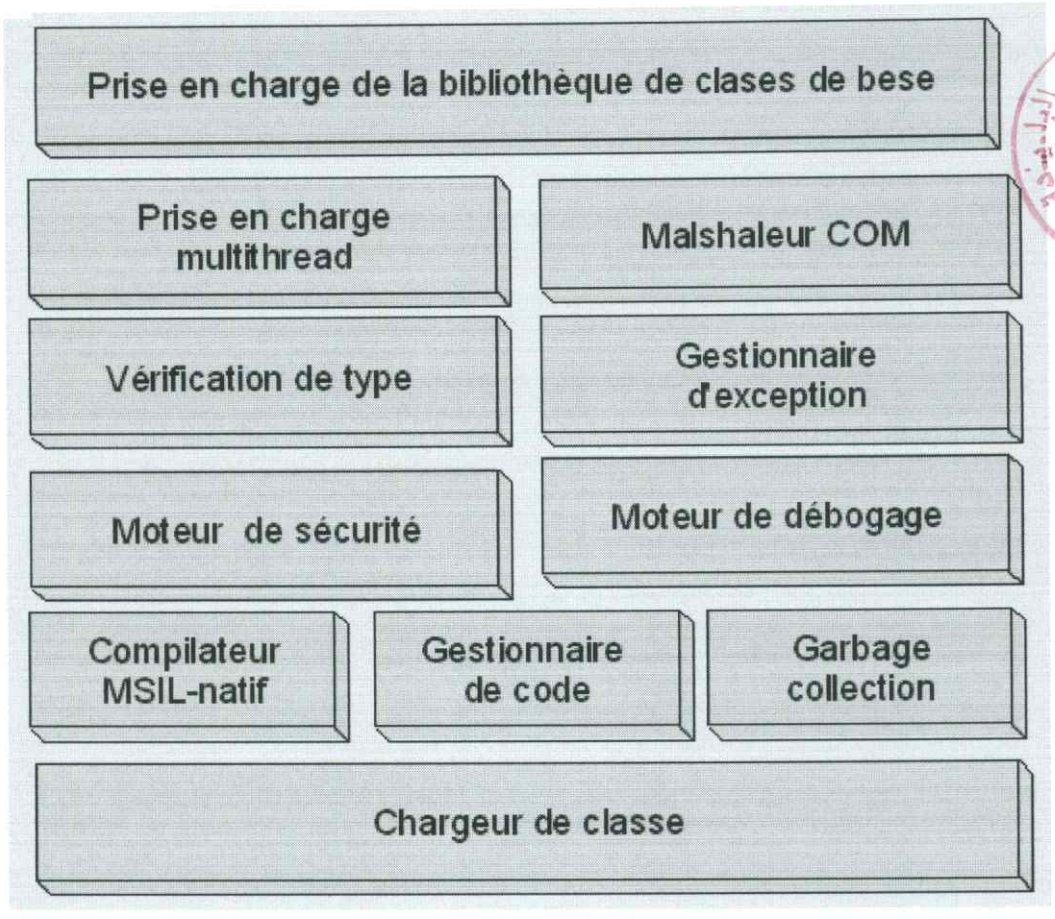


Figure A.4 Composants du Common Langage Runtime.

- **Chargeur de classe :** charge en mémoire l'implémentation d'un type chargeable et le répare à l'exécution.
- **Compilateur MSIL(**Microsoft Intermediate Language **)- natif :** convertit MSIL en un code natif.
- **Gestionnaire de code :** gère l'exécution du code.
- **Garbage collection :** fournit une gestion automatique de la durée de vie de tous nos objets.
- **Moteur de sécurité :** fournit une sécurité par preuve, fondée sur l'origine du code en plus de l'utilisateur.
- **Moteur de débogage :** permet de débogue l'application et de tracer l'exécution du code.
- **Vérification de type :** n'autorisera pas les conversions non sécurisées ou les variables non initialisées.

- **Gestionnaire d'exceptions :** fournit un traitement structuré des exception.
- **Prise en charge multithread :** fournit des classes et des interface qui permettent la programmation multithread.
- **Marchaleur COM :** fournit le marshaling à partir et à destination de COM.
- **Prise en charge de la bibliothèque de classes.NET Framework :** intègre du code au runtime qui prend en charge la bibliothèque de classes.

3.2-Bibliothèque de classes .NET Framework :

Elle fournit de nombreuses nouvelles fonctionnalités puissantes du runtime et d'autres services essentiels de haut niveau via une hiérarchie d'objets qui s'appelle un espace de nom.

La figure suivante décrit ces espaces de nom :

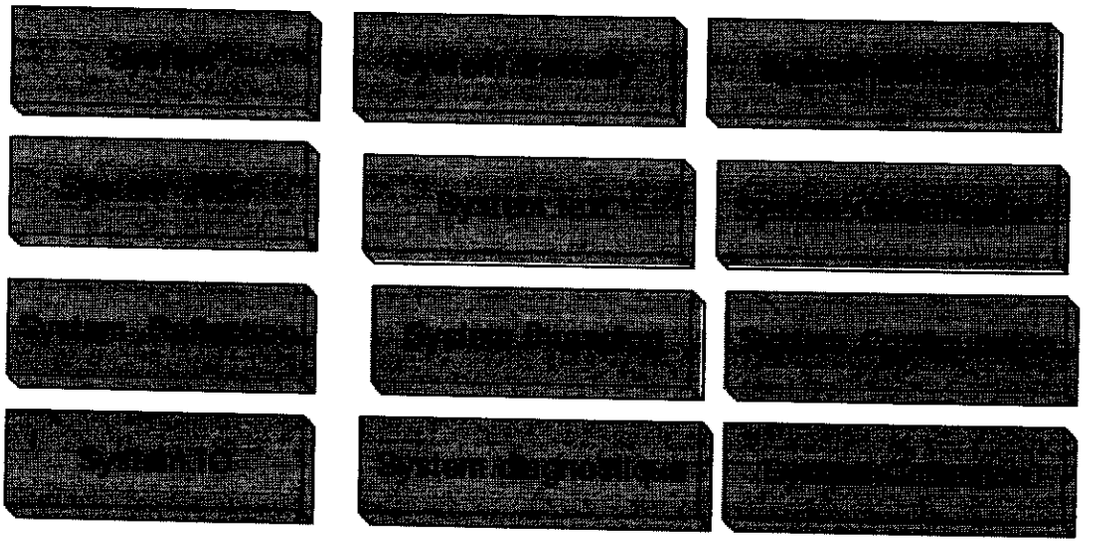


Figure A.5 Bibliothèque de classes .NET

- **System :** contient des classes fondamentales et de base qui définissent les types de données, les événements, les interfaces, les attributs…etc.
- **System.Collection :** fournit des listes, des tables et d'autres méthodes de regroupement de données.
- **System.IO :** il s'agit d'entrée /sortie et flux de fichiers.
- **System.NET :** fournit une prise en charge des sockets et de TCP/IP.

Pour plus d'information consulter la documentation du SDK Microsoft .NET Framework.

3.3- ADO .NET données et XML :

ADO.NET est la nouvelles génération de la technologie ADO (ActiveX Data Object) Son but est l'amélioration du modèle de programmation déconnecté, ainsi elle est riche de XML.

- **System.Data :** comprend la classe **DataSet** qui représente des tables mutiples et leur relations.
- **System.XML :** il comprend un outil d'écriture et un analyseurXML.

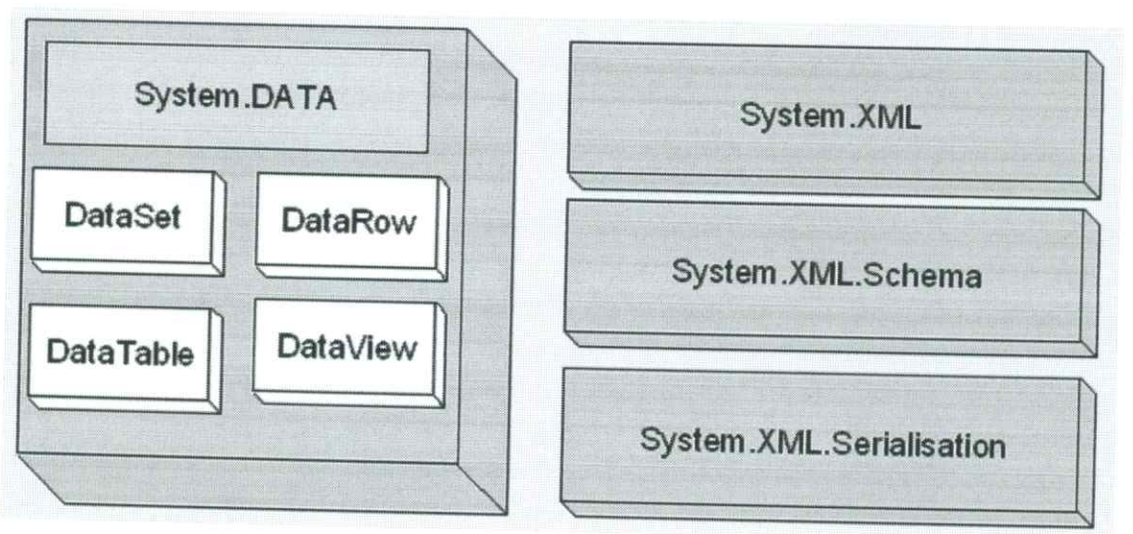


Figure A.6 ADO.NET

3.4- ASP .NET formulaires et services Web (Active Server Pages) :

ASP.NET est un cadre de programmation élaboré sur la base du Common Language Runtime et qui peut être employé sur un serveur pour créer des applications Web puissantes. Les formulaires ASP.NET sont des outils d'emploi pour la création d'interface utilisateur Web dynamiques.

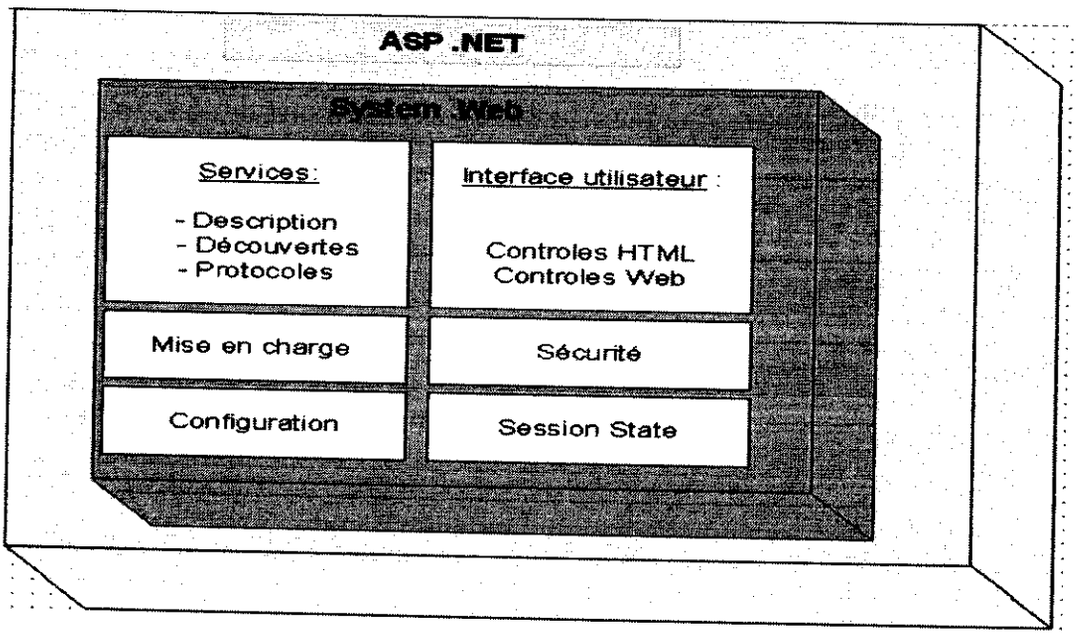


Figure A.7 Présentation des formulaire et services Web : ASP .NET

Dans **System.Web,** certais services tels que la mise en charge, la sécurité ou la configuration sont partagés par les services Web et l'interface.

3.5 Interface utilisateur :

La figure suivante explique comment le .NET gère l'interface Framework des applications Windows traditionnelles :

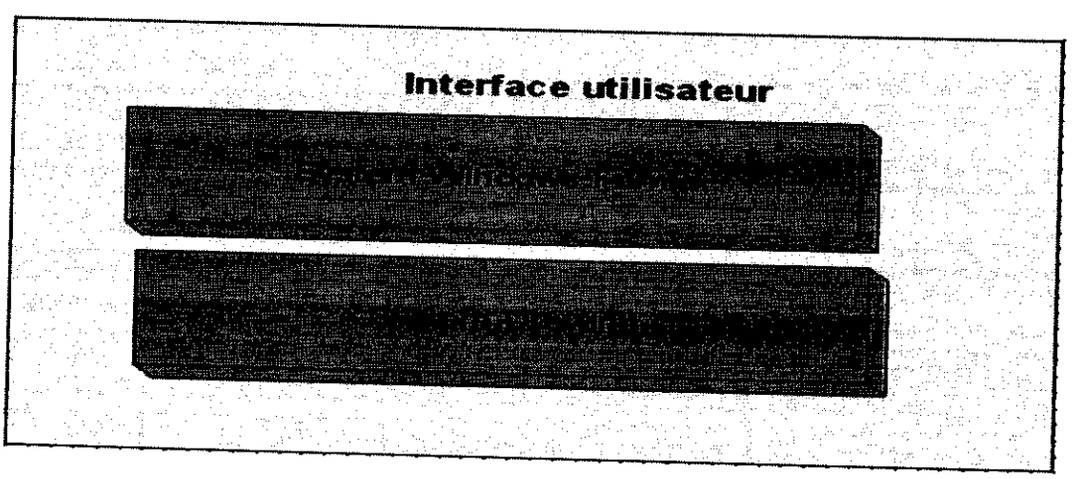


Figure A.8 Interface utilisateur.

- **System.windows.Forms** représente l'interface utilisateur coté client ;tandis que **System.Drawing** représente la nouvelle génération de services GDI + (Graphic Device Interface Plus).

Résumé :

L'objectif de ce travail est de proposer un format standard pour les fichiers logs, il consiste dans un premier temps à étudier les principaux fichiers logs générés par les Firewalls, les Sites Web et les SGBD afin de pouvoir évaluer l'opportunité de mettre en place un format standard. Cette étude nous a permis de constater qu'il était intéressant de concevoir un format pour les fichiers logs des SGBD. Ensuite nous proposerons le formats standard basé sur la norme XML et enfin, nous développons un logiciel pour l'exploitation de ce format afin d'aboutir à des conclusion relatives à la sécurité du système.

Mots clés : Sécurité, LOG, audit de la sécurité, XML, Firewall, Site Web, SGBD.

Abstract:

The main goal of our work is to propose a standard format for log files. First, it consists of studying the most important log files :(those generated by Firewall, Web Sites and DBMS) to evaluate the opportunity of developing standard format. These studying help us that it was important to conceive a format for DBMS log files. Then, we propose the standard format based on XML norm. Finally, we develop software for operating this format in order to lead to system security conclusions.

Key words: Security, LOG, security audit, XML, Firewall, Web sites, DBMS.