Mesure d'audience d'un site Web : Utilisation du formalisme XML

HANOUNE Mostafa *, BENABBOU Faouzia, A. Marzak

Laboratoire des technologies de l'information et modélisation (TIM), Faculté des sciences Ben M'Sik,
Université Hassan II, Mohammedia, Casablanca, Maroc
mhanoune@gmail.com, hgfbenabbou@menara.ma
* mhanoune@gmail.com, m_hanoune@yahoo.fr

1. Introduction :

Le comportement de l'utilisateur sur un site Web se représente par une suite de clics de souris et de saisies sur un clavier. Ces informations déclenchent des requêtes qui ont pour résultat l'affichage de certaines pages du site **[1]**. Ces requêtes sont enregistrées dans un fichier, à mesure qu'elles sont déclenchées par l'utilisateur, de manière standardisée, de façon à ce qu'il soit possible de procéder à des analyses par la suite. Cette base de données, constituée, est communément appelée **fichier log**. Son analyse et exploitation permettent, en principe, de savoir par exemple, quelles sont les requêtes qui n'aboutissent pas (page manquante, lien erroné…) ou encore quelle est la fréquentation d'une page spécifique. Cependant cela n'est possible que si la structure et le contenu de ce fichier correspondent à un formalisme qui s'y prête, le formalisme le plus répandu de fichier log est **« l'ELF »** (Extended Log File Format) **[2]**. Chaque ligne de ce fichier donne une information sur l'utilisateur, son matériel, la date et l'heure de la requête, la page requise, le statut de la page requise, la page de référence ainsi que quelques informations liées au protocole d'échange de données.

Notre objectif est de développé une application qui peut analyser les données du fichier log et fournir des statistiques de différents types en respectant le processus ECD, qui est un processus logiciel assez complexe permettant l'extraction d'informations originales, auparavant inconnues et potentiellement utiles, à partir de données, et ce de façon automatique **[3]**. Ce processus commence par le nettoyage et la récupération des données sous un format adapté aux étapes qui suivront **[4]**. L'ensemble des outils logiciels assurant ces fonctionnalités sont appelés outils E.T.L (pour **E**xtraction, **T**ransformation and **L**oading). Le processus se poursuit alors par l´étape de fouille proprement dite. Son déroulement dépend très largement de la technique de fouille employée. L'application E.C.D. doit ensuite permettre la visualisation des résultats sous forme de graphiques ou de tableaux de bord. Ces fonctionnalités sont appelées outils de visualisation et outils de reporting **[5]**.

Notre base de départ sera donc le fichier log, sur lequel on fera plusieurs traitements pour aboutir à des fichiers XML, contenant des informations plus facilement analysables et exploitables.

Les trois phases du processus de traitement du fichier log seront: **Prétraitement, Fouille et Déploiement.**

Exemple d'un fichier log :

```
161.31.132.116 - - [21/Dec/2001:08:42:55 -0500] "GET /home.htm HTTP/1.0" 200 4392 "http://fr.search.yahoo.com/fr?p=peinture"
"Mozilla/4.7 [en] (Win98)"
161.31.132.116 - - [21/Dec/2001:08:43:59 -0500] "GET /images/flagfr.jpg HTTP/1.0" 304 - "-" "Mozilla/4.7 [en] (Win98)"
209.130.181.212 - - [21/Dec/2001:08:44:02 -0500] "GET /cs HTTP/1.1" 301 236 "-" "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)"
209.130.181.212 - - [21/Dec/2001:08:44:03 -0500] "GET /cs/ HTTP/1.1" 200 1643 "-" "Mozilla/4.0 (compatible; MSIE 5.5; Windows
98)"
209.130.181.212 - - [21/Dec/2001:08:44:05 -0500] "GET /cs/frameh.htm HTTP/1.1" 200 7363 "/cs/" "Mozilla/4.0 (compatible; MSIE 5.5;
Windows 98)"
161.31.132.116 - - [21/Dec/2001:08:44:09 -0500] "GET /bienvenue.htm HTTP/1.0" 304 - "/home.htm" "Mozilla/4.7 [en] (Win98)"
```

2. Analyse et conception UML :

2.1 Diagramme de cas d'utilisation :

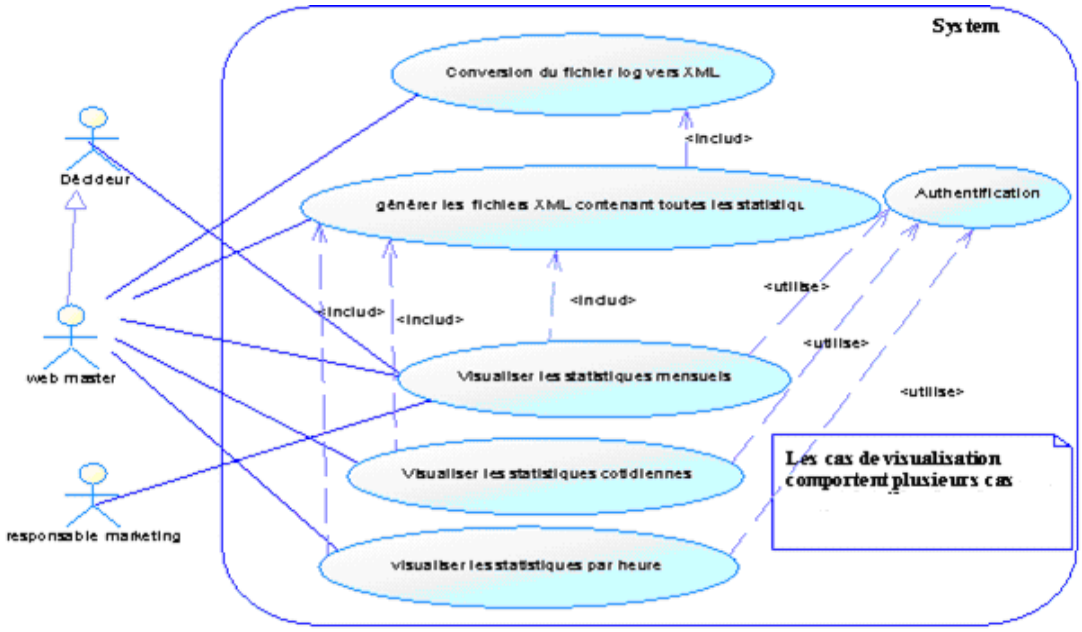


Figure 1 : Cas d'utilisation

Le diagramme cas d'utilisation permet de visualiser les besoins des utilisateurs et les objectifs correspondants.

Les différents cas d'utilisation

- Transformer le fichier log en plusieurs fichiers XML (web master)
- Transformer le fichier XML en plusieurs fichiers XML de statistique (web master)
- Visualisation des statistiques sur les erreurs par page (web master)
- Visualisation des statistiques sur les pays d'origine des visiteurs (web master)
- Visualisation des statistiques sur les robots (web master)
- Visualisation des statistiques concernant le taux de retour des visiteurs par jour (Décideurs).
- Visualisation des statistiques sur le taux de fidélité (Décideurs).
- Visualisation des statistiques sur le nombre de visite, de visiteurs, de consultation et de trafic (Décideurs)
- Visualisation des statistiques sur les sites qui nous sert de référant (responsable marketing)
- Visualisation de toutes les connaissances fixées dans le cahier de charge

2.2 Diagrammes d'activité :

2.2.1 Diagrammes d'activité du prétraitement du fichier log :

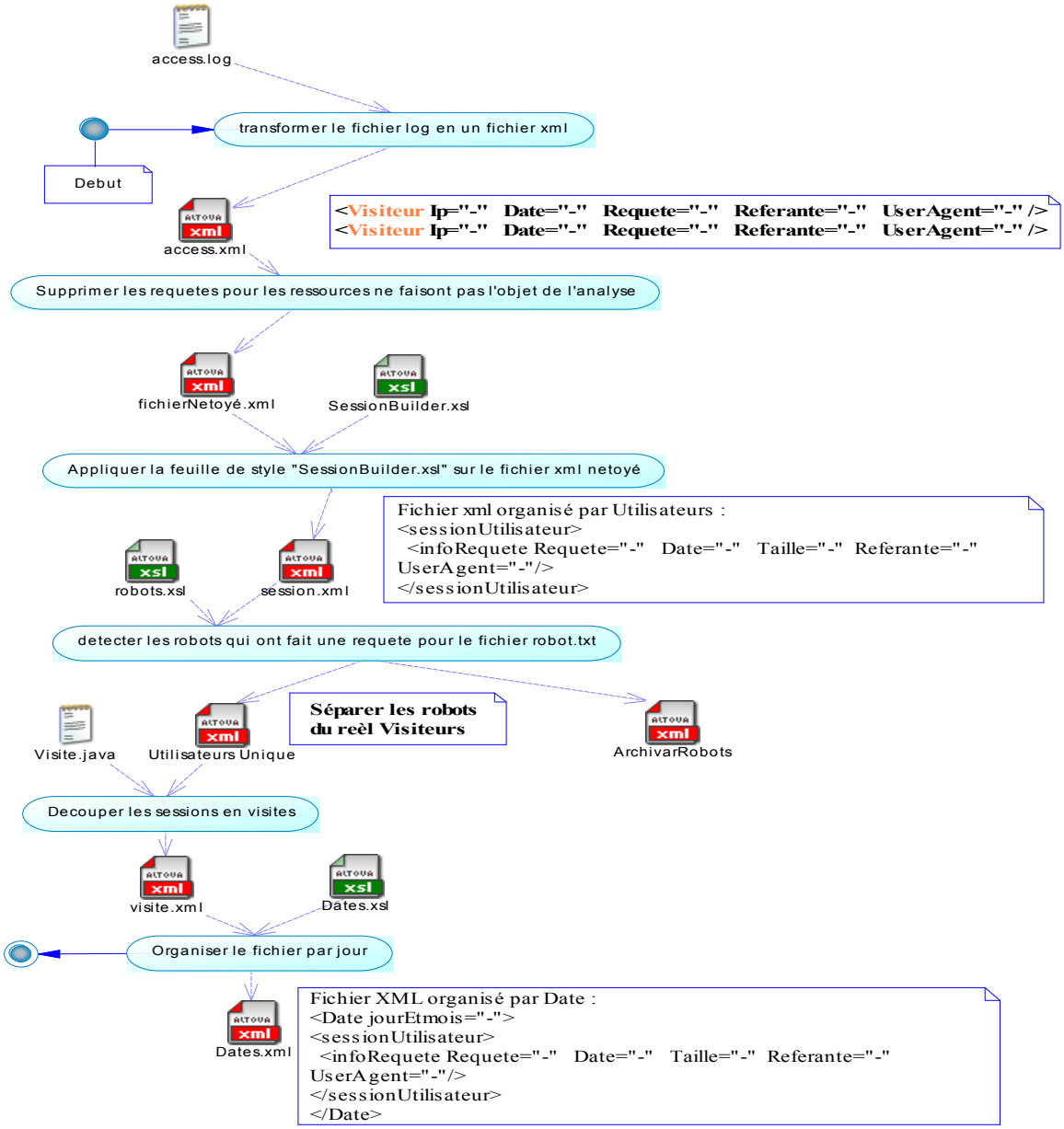


Figure 2 : Prétraitement du fichier log

Dans cette étape on montre toute les phases par les quelles passe le fichier log avant de commencer son exploitation.

2.2.2 Génération des Fichiers de calcule :

2.2.2.1 Les Statistique par jour et par heure :

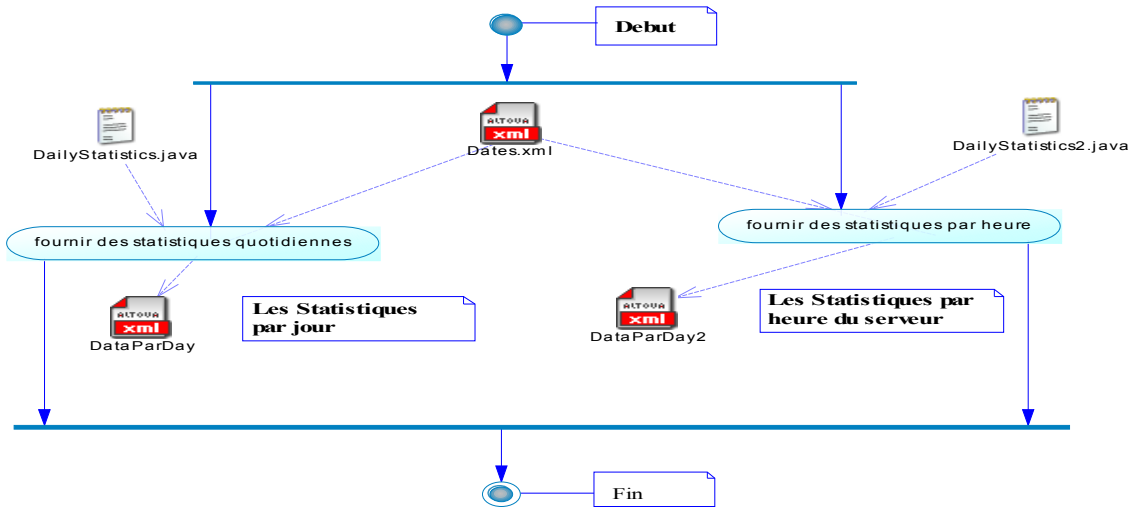


Figure 3 : Diagramme d'activité des Statistiques par jour et par heure.

Dans cette étape on s'intéresse aux statistiques relatives au Jour et à l'Heure

2.2.2.2 Les Statistiques par mois :

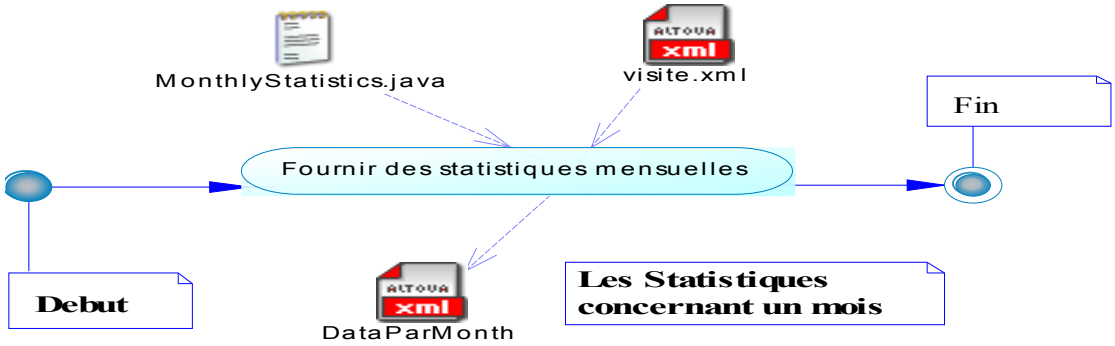


Figure 4 : Diagramme d'activité des Statistiques par mois

Dans cette étape on s'intéresse aux statistiques relatives au mois

2.2.2.3 Les Pages référencées :

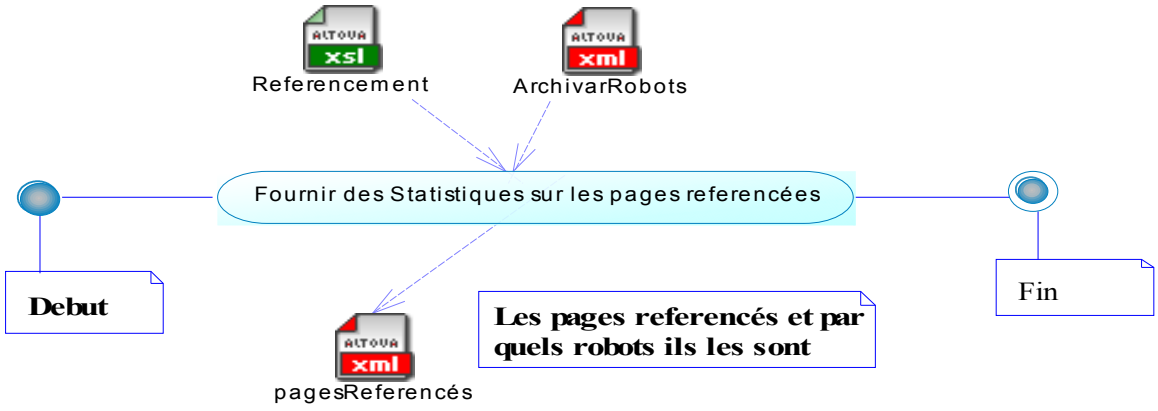


Figure 5 : Diagramme d'activité des pages référencées

Dans cette étape on s'intéresse au référencement des pages et au robots qui les ont référencés.

2.2.3 Transformation des fichiers XML en histogrammes et tables :

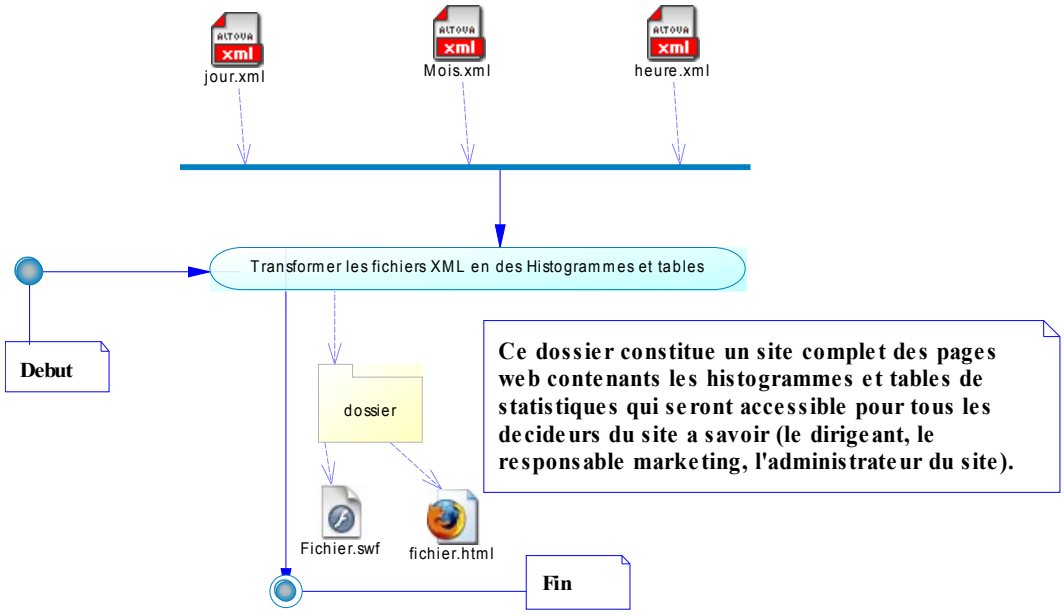


Figure 6 : Diagramme d'activité du Transformation des fichiers XML en histogrammes et tables

2.3 Diagramme de classes :

A partir du diagramme de classes ci-dessous nous avons généré les fichiers XML qui nous ont servi à tracer les histogrammes et à créer les tables de statistiques.

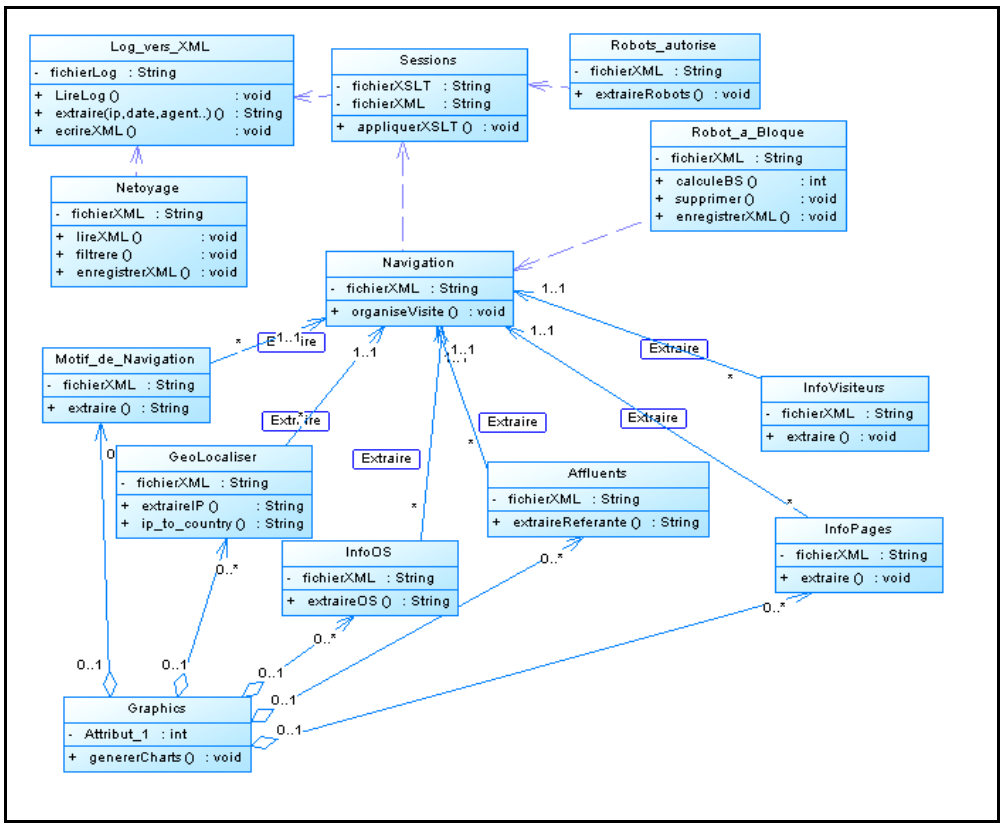


Figure 7 : Diagramme de classe

2.3 *Diagramme de séquences*

Dans cette partie on s'intéresse à la représentation des messages passés entre les instances d'objets. Ces diagrammes indiquent le comportement public dont les objets ont besoin afin de pouvoir travailler et coopérer correctement, ils seront utilisés pour montrer les responsabilités et opérations effectives assignées à chaque classe.

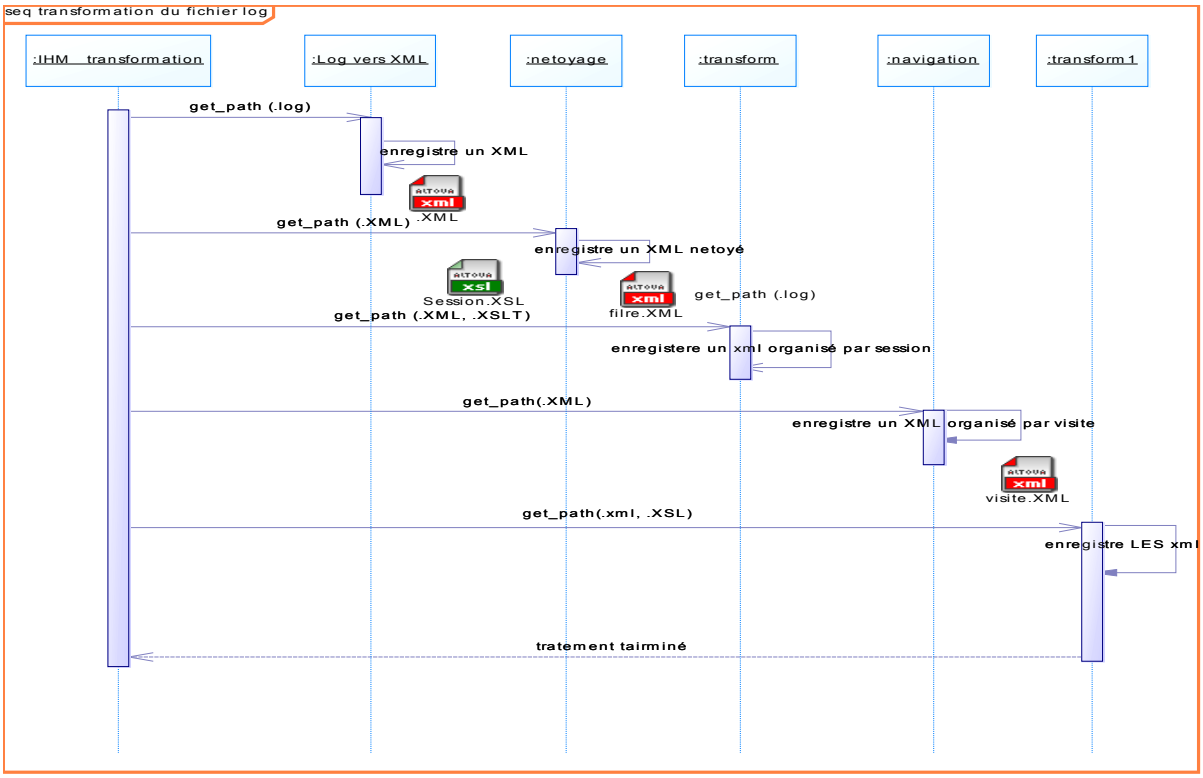


Figure 8 : Séquences de transformation du Fichier Log

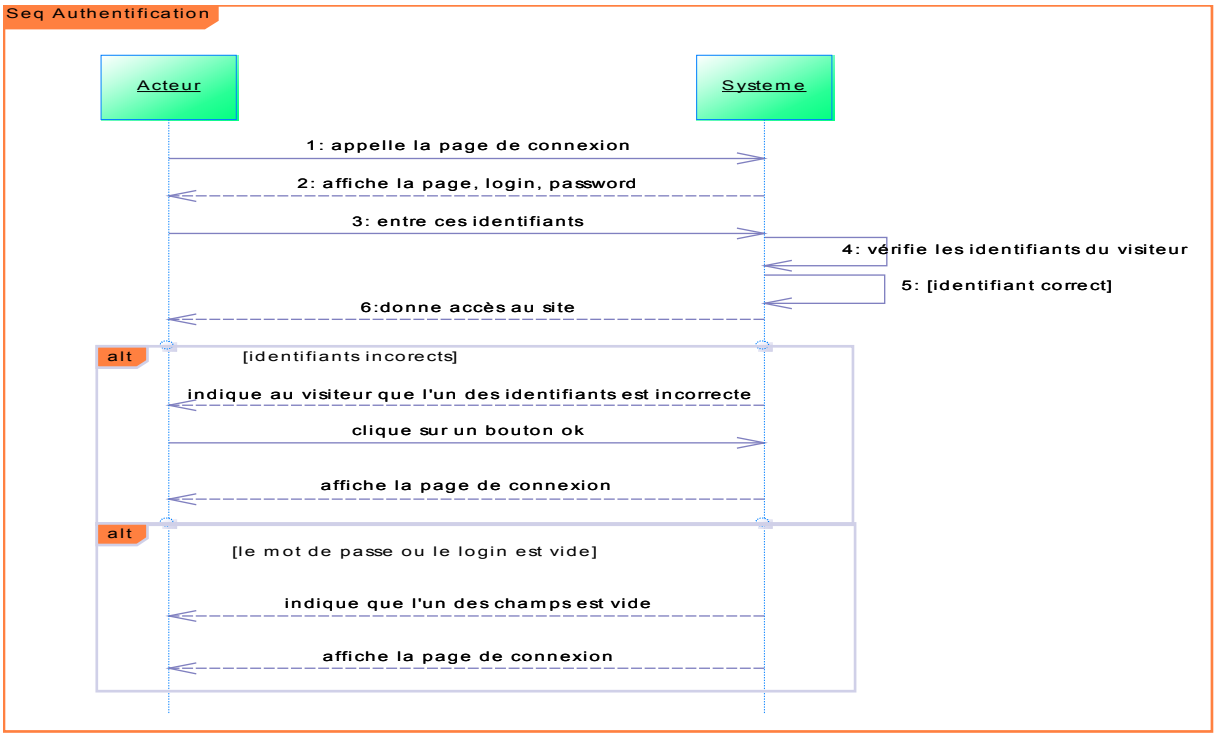


Figure 9: Séquences d'authentification

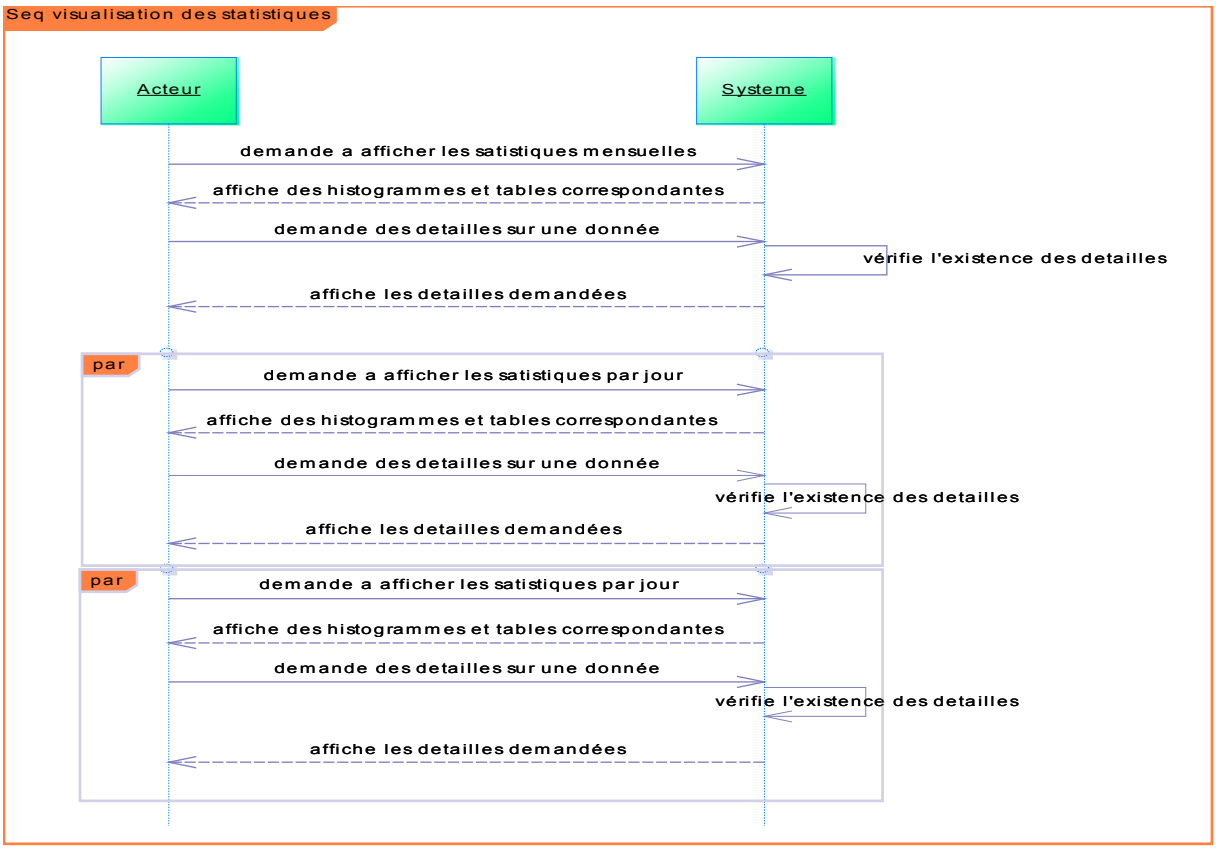


Figure 10: Séquences de visualisation des statistiques

2.4 *Diagramme d'états de transition :*

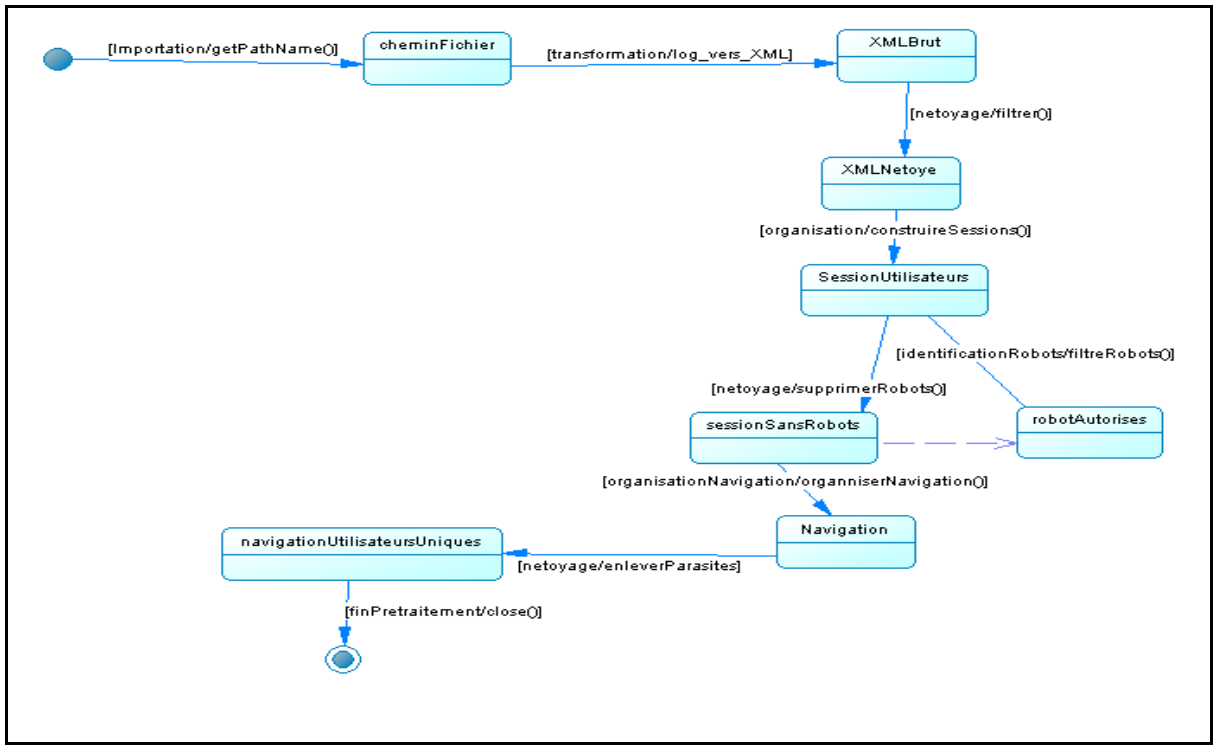


Figure11 : Etats de transition du fichier log.

Les diagrammes d'état sont utilisés pour saisir les transformations du système à travers le temps, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

Au moment de l'exécution, chaque objet possédant des attributs non constants pourra avoir potentiellement un certain nombre d'états.

3. Prétraitement du fichier log

Le prétraitement du fichier log se fait en 6 étapes :

Etape 1 : Convertir le fichier log vers un format manipulable : XML.

Etape 2 : Nettoyer le fichier XML obtenu, des données ne faisons pas l'objet de l'analyse (Ces données sont en générale les fichiers avec les extensions suivantes : .jpeg, .css, .class, .gif, .ico…)

Etape 3 : Construction des sessions utilisateurs.

Etape 4 : Séparer les robots des visiteurs physiques.

Etape 5 : Construction des visites. (Le fichier navigation.xml).

Etape 6 : suppression des requêtes en provenance des robots, autre que ceux d'indexation.

Ces étapes sont illustrées par la figure 12 ci-après

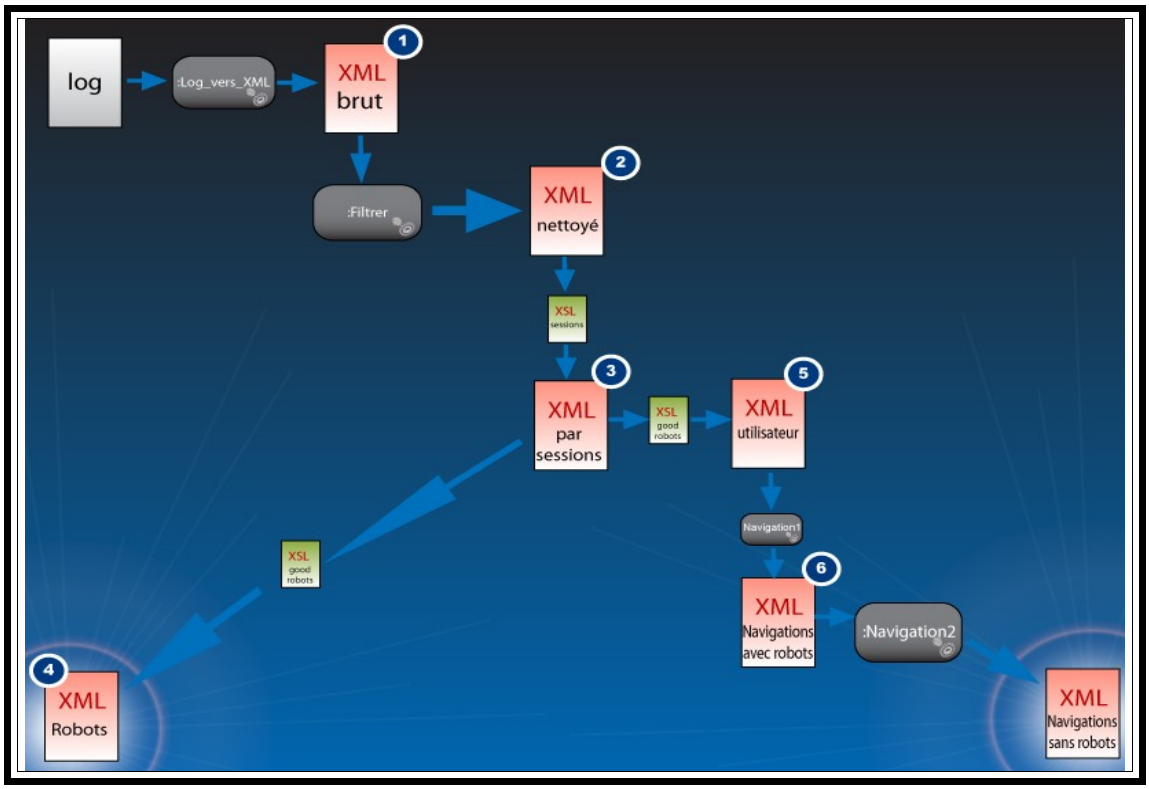


Figure 12 : La phase de prétraitement

3.1 Passage au format XML

Notre premier travail de prétraitement sera la transformation du fichier Log vers le Formalisme XML.

Génération du fichier XML à partir du fichier log.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<LogDuServeur>
<Visiteur IP="66.249.72.138" NomUtilisateur="-" MotPass="-"
date="30/04/2007:13:12:18" requete="/bb/viewforum.php"
Protocole="HTTP/1.1" statutCode="200" tailleFichier="4231" referante="-"
agent="Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)" />
</LogDuServeur>
```

3.2 Nettoyage des données

La phase de nettoyage des données, consiste à supprimer les requêtes inutiles de fichiers log. Ces requêtes concernent souvent les images et les fichiers multimédia, ou toute autre requête pour une ressource web ne faisant pas l'objet de l'analyse.

Prenons par exemple une page web composée du texte, d'une image, une applet java et probablement une feuille de style. Si un visiteur demande à afficher cette page nous aurons 3 voir 4 nouvelles entrées dans le fichier log, ce qui pourra être interprété par 4 consultations, ce qui n'est pas le cas.

Pour ce nettoyage nous avons utilisé l'API **JDOM** (un parseur XML en java), ce qui a permis la suppression de toutes les requêtes, concernant un fichier de type : **.jpg, .gif, .png, .ico, .css, .jpeg, .class, .jsc.**

Code java permettant la suppression des éléments XML :

```
If (element.getAttributeValue ("requete").contains(".gif") ||
element.getAttributeValue("requete").contains(".jpeg")||
element.getAttributeValue("requete").contains(".css")||element.getAttributeValue
("requete").contains(".class")||element.getAttributeValue("requete").contains(".jpg")||
element.getAttributeValue("requete").contains(".ico"))
{remove (content) ;}
```

3.3 Construction des sessions utilisateur

Il s'agit dans cette partie de regrouper toutes les requêtes d'un même utilisateur.

Difficultés d'identification d'utilisateur :

Pour regrouper les requêtes, il est nécessaire de savoir quels utilisateurs les ont émises. Si l'utilisateur a accepté de s'enregistrer et s'identifie avec un login, alors le repérage est immédiat, mais cela ne concerne qu'une très faible minorité des visites. Une autre méthode répandue mais nécessitant l'acceptation de l'utilisateur, consiste à écrire dans la mémoire du navigateur. C'est-à-dire sur le poste client, un fichier d'identification nommée *cookie* qui sera réutilisé dans chacune des requêtes et permettra au serveur d'en identifier la provenance. En fait on ne dispose que de l'adresse IP qui est dynamique et qui est, en plus, identique pour tous les utilisateurs partageant un même routeur ou accédant à l'Internet via un même serveur proxy. Dans ce cas il est difficile de parler d'identification d'utilisateur.

En pratique nous avons considéré le couple (IP ; navigateur) comme étant un utilisateur et nous avons écrit une feuille de style **XSLT** qui regroupe les requêtes de chaque utilisateur sous le nom d'une même balise (voir l'exemple 1 et 2 ci dessous).

Exemple1 : La feuille de style permettant de construire les sessions utilisateur

```
<?xml version="1.0" encoding="UTF-8" ?>
-<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
   <xsl:text />
     <totalSessionUtilisateur>
       <xsl:for-each select="LogDuServeur/Visiteur">
   <xsl:variable name="adresseIP" select="@IP" />
   <xsl:variable name="infoOs" select="@agent" />
          <xsl:if test="count(preceding::Visiteur[@IP=$adresseIP and
 @agent=$infoOs]) = 0">
             <xsl:element name="sessionUtilisateur">
               <xsl:attribute name="AdresseIP">
```

```
        <xsl:value-of select="@IP" />
      </xsl:attribute>
            <xsl:attribute name="OS">
       <xsl:value-of select="@agent" />
      </xsl:attribute>
- <xsl:for-each select="/LogDuServeur/Visiteur[@IP=$adresseIP and
  @agent=$infoOs]">
- <xsl:element name="infoRequete">
- <xsl:attribute name="requete">
 <xsl:value-of select="@requete" />
</xsl:attribute>
</xsl:element>
</xsl:for-each>
</xsl:element>
</xsl:if>
</xsl:for-each>
</totalSessionUtilisateur>
</xsl:template>
</xsl:stylesheet>
```

En appliquant la feuille de style dans l'exemple 1 nous avons eu en résultat un fichier XML décrit dans l'exemple 2 ci-dessous.

Exemple 2 : Le fichier XML organisé par sessions utilisateurs

```
<?xml version="1.0" encoding="UTF-8" ?>
<totalSessionUtilisateur>
 <sessionUtilisateur AdresseIP="74.6.68.213" OS="Mozilla/5.0 (compatible;
Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)">
<i  nfoRequete requete="/forum/index.php"
dateRequete="30/04/2007:13:47:05" referante="-" codeRetour="404"
taille="171" /> </sessionUtilisateur></totalSessionUtilisateur>
```

3.4 Identification des robots

Deux sortes de robots parcours continuellement la toile :

- Les robots conformes aux normes dits robots d'indexation.
- Les robots non conformes aux normes. Leurs comportements sur les sites, présente une éventuelle atteinte à la sécurité et à la confidentialité des données de ces derniers.

Les robots d'indexation consultent toujours le fichier robots.txt avant de parcourir un site donné. Le fichier robots.txt définit les parties du site que l'administrateur du site veut référencer et quel est le moteur de recherche qui y sera autorisé.

En pratique nous avons utilisé trois heuristiques pour identifier les requêtes ou visites issues des robots:

a. Identifier les IP qui ont sollicité la page '\robots.txt' dans une requête.

b. Utiliser des listes de « User agents » connus comme étant des robots.

c. Utiliser un seuil pour « la vitesse de navigation » *BS* (_Browsing Speed_), qui est égale au rapport: *BS = (nombre de page) / (Durée de la visite*). Si *BS* > 2 pages/seconde, alors la visite provient d'un robot.

3.5 Construction des visites

Une visite est représentée par un ensemble de requête en provenance d'un même utilisateur dans une durée déterminée dés sa connexion au site jusqu'à ce qu'il ferme toutes les pages du site.

En pratique dans un fichier log on n'a aucun indice de la de connexion d'un utilisateur au site. Alors nous nous somme proposé de calculer, pour chacune de deux requêtes adjacentes la différence de temps qui les sépare, et nous avons coupé les visites dés que l'écart de temps entre deux requêtes est plus grand que 30mn. Autrement dit, une durée d'inactivité sur le site supérieure à 30 minutes sera considérée comme étant une déconnexion de cet utilisateur, et la requête qui suivra appartiendra à une nouvelle session.

4. Fouille des données / Extraction des connaissances

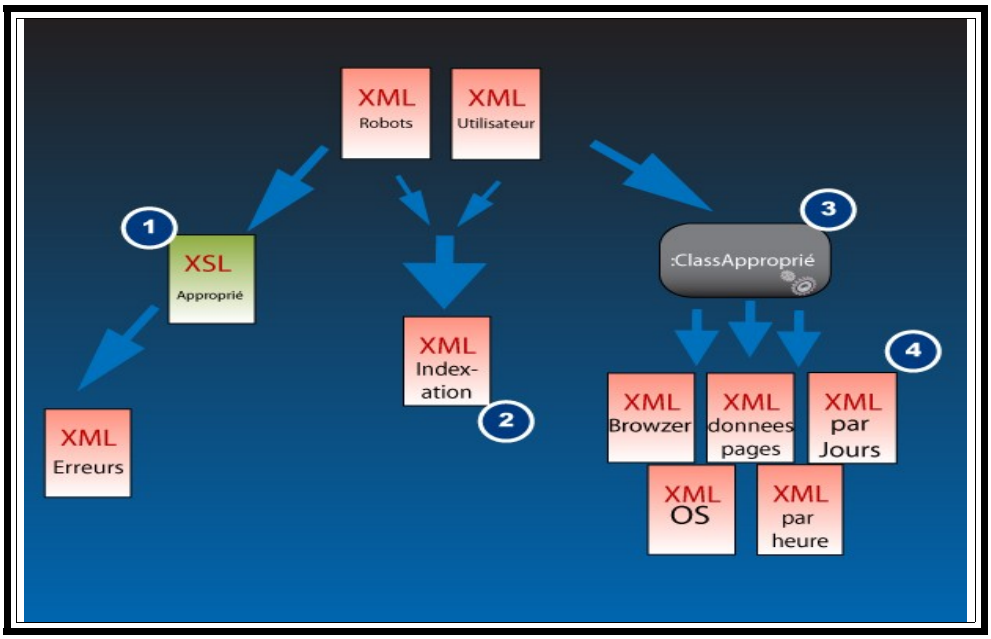


Figure 13 : La phase de fouille des données

Dans la phase de fouille, nous partons de deux fichiers XML : **Fichier des utilisateurs physiques** et **Fichiers des robots**.
Les étapes de fouille de données sont :
Etape 1 : Génération d'une feuille de style XSLT, qui prend en entrée les fichiers XML organisant les visiteurs, et fournie en sortie un fichier XML qui contient les erreurs.
Etape 2 : Génération d'un fichier XML contenant les statistiques de référencement.
Etape 3 : Création d'un ensemble de classe java qui génèrent les fichiers XML de statistiques.
Etape 4 : Génération de fichiers statistiques après l'exécution des classes java en Etape 3.

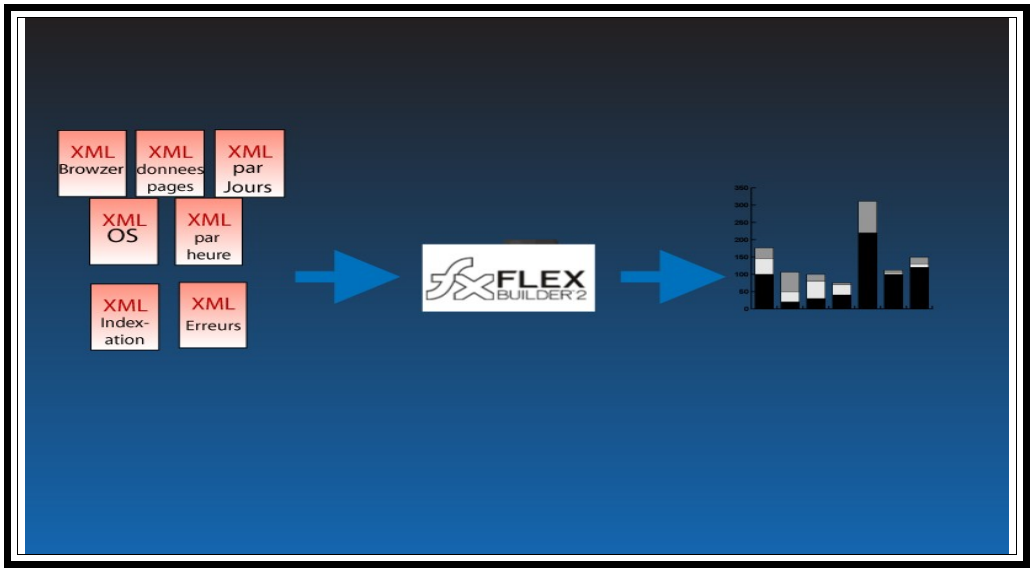


Figure 14 : La phase de déploiement

Par la suite, on a exporté les données des fichiers XML vers des objets Flex, de dessin de graphiques, ce qui a permis l'affichage des données en tables et histogrammes, facilement exploitable.

Après avoir prétraité et organisé, en session et puis en visite, le fichier XML ; nous pouvons maintenant faire les statistiques adéquates sur ce fichier.

Les connaissances sur le référencement ont été extraites du fichier XML, contenant les sessions des robots d'indexation. Pour ceux concernant les pays de provenances des visiteurs nous les avons obtenus en se connectant à une base de données Access (ip-to-contry). pour les statistiques par jour nous les avons obtenus a partir d'un fichier XML organisé par jour.

Dans les exemples ci-dessous nous montrons à quoi ressemblent les fichiers XML qui contiennent les statistiques.

```
<?xml version="1.0" encoding="UTF-8" ?
 >
- <result>
- <heure valeur="0">
  <visiteur totale="11" avg="1,73" />
   <visite totale="14" avg="1,97" />
  <consultation totale="33" avg="0,82" /
 >
  <Kbyte Kbyte="234907" avg="1,08" /
```

Exemple 1 Les statistiques par heure

```
<?xml version="1.0" encoding="UTF-8" ?
 >
- <result>
   <Date jourEtMois="30/04"
visiteur="9"   visite="9"
consultation="11" Kbyte="39970"   />
   <Date jourEtMois="01/05"
visiteur="15"  visite="16"
consultation="30" Kbyte="172538" />
```

Exemple 2 Les statistiques par jour

```
<?xml version="1.0" encoding="UTF-8" ?
 >
- <result>
 <pageEntree nom="/" valeur="78"
 pourcentage="10,99" />
</result>
```

Exemple 3 Les pages d'entrée au site

```
<?xml version="1.0" encoding="UTF-8" ?
 >
- <result>
 <pageSortie nom="/forum/index.php"
 valeur="1" pourcentage="0,14" />
 </result>
```

Exemple 4 Les pages de sortie du site

```
<?xml version="1.0" encoding="UTF-8"?>
- <result>
  <motif
  combainaison="/bb/viewtopic.php
  /bb/viewforum.php " valeur="2"
```

```
<?xml version="1.0" encoding="UTF-8" ?
  >
- <result>
  <site
   nom="http://www.bramjnet.com"
   nombre="254" /> </result>
```

Exemple 7 La fidélité des utilisateurs

```
<?xml version="1.0" encoding="UTF-8" ?
  >
- <result>
  <Date jourEtMois="30/04" connus="9"
  nonConnus="0" />
</result>
```

```
<?xml version="1.0" encoding="UTF-8" ?
  >
- <result>
  <Date jourEtMois="01/05" totale="15"
  tauxDeVariation="66,67" />
</result>
```

Exemple 8 Le taux de fréquentation par jour

```
<?xml version="1.0" encoding="UTF-8" ?
  >
- <result>
  <searshWord words="popuniversity"
  nombre="1" />
  </result>
```

Exemple 9 Les mots clés amenant au site

```
<?xml version="1.0" encoding="UTF-8" ?
  >
- <result>
<RobotARejeté
  adresseIP="24.191.97.135" />
  </result>
```

Exemple 10 Les robots à bloquer

```
<?xml version="1.0" encoding="UTF-8" ?>
- <result>
<navigateur nom="Gecko" valeur="67" />
<navigateur nom="IE" valeur="498" />
<navigateur nom="Khtml" valeur="16" />
<navigateur nom="Opera" valeur="8" />
</result>
```

Exemple 11 Les navigateurs des visiteurs

```
<?xml version="1.0" encoding="utf-8"
  standalone="no" ?>
- <result>
  <page nom="/forum/index.php"
  totalErreurs="1" vu="1" taux="0">
<erreur Type="Introuvable (404)"
  Coté="Client" nombre="1" />
  </page>  </result>
```

Exemple 12 Les erreurs par page

Ces exemples de fichiers XML sont générés dans la partie serveur de l'application (celle écrite en java sous Eclipse). Ces fichiers vont être utilisés, par la suite, avec la partie client de l'application (celle écrite en ActionScript sous Flex).

Le choix d'une application deux tiers est argumenté par le fait que ; les utilisateurs de l'application sont nombreux ; et pour éviter le chargement et le traitement du fichier log par tous les utilisateurs du système (seul le web master peu charger le fichier log, et la partie serveur de l'application. Cependant les exploitants de l'application vont pouvoir visualiser les statistiques à partir de leur bureau, partie client). Ainsi nous avons réussi à garder la confidentialité des données de site et en même temps délocalisé les informations des statistiques.

Ecrans réalisés

Dans ce chapitre nous allons présenter les résultats obtenus en analysant le fichier log de notre site dont le nom de domaine est le suivant : WWW.POPUNIVERSITY.COM.

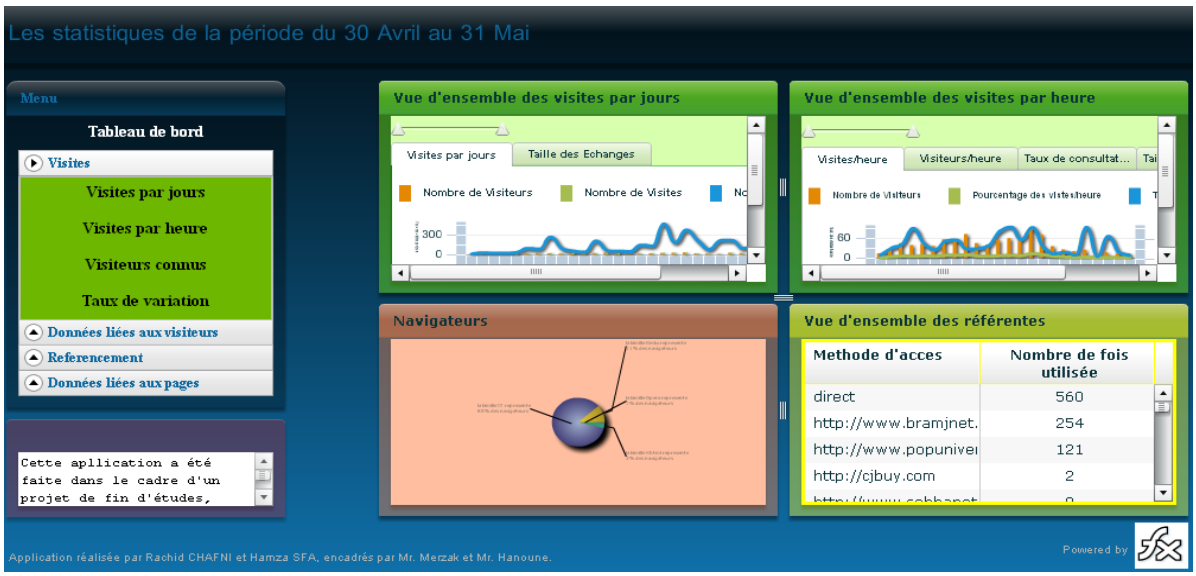


Figure 15 : La page de démarrage du site

En cliquant sur un item du menu à gauche, l'histogramme ou le tableau associé s'affiche au centre de la page.

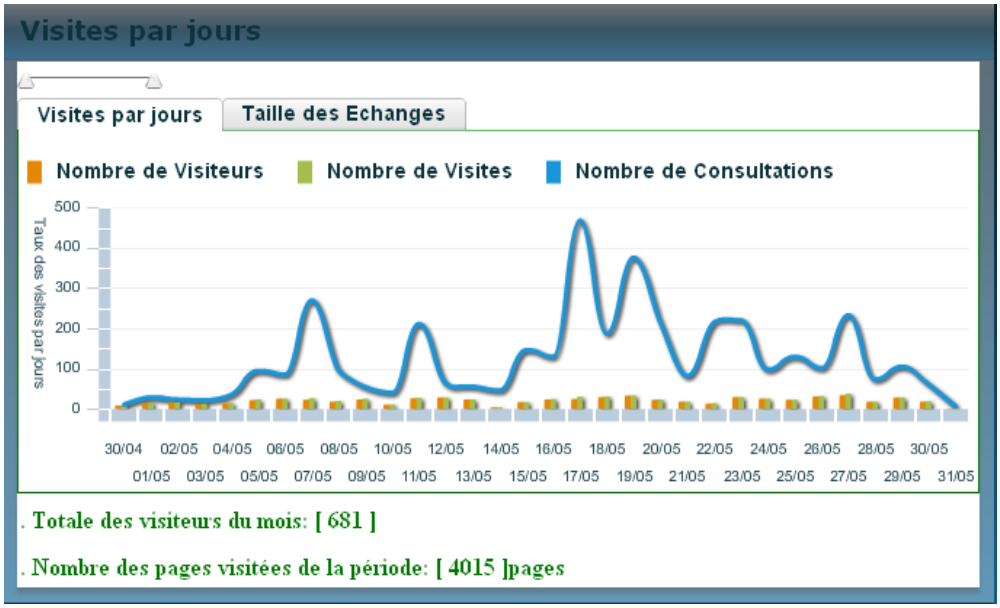


Figure 16 : Fenêtre des visites par jours

La figure 16 montre la page qu'on obtient en cliquant sur le lien, visites par jour, du menu à gauche. Cet histogramme nous renseigne sur le nombre de visiteurs, le nombre de visites et le nombre de consultations du trafic par jour. On peut restreindre les statistiques à un nombre de jours déterminé, on déplaçant la barre en haut de « la figure 16 » a l'aide de la souri.

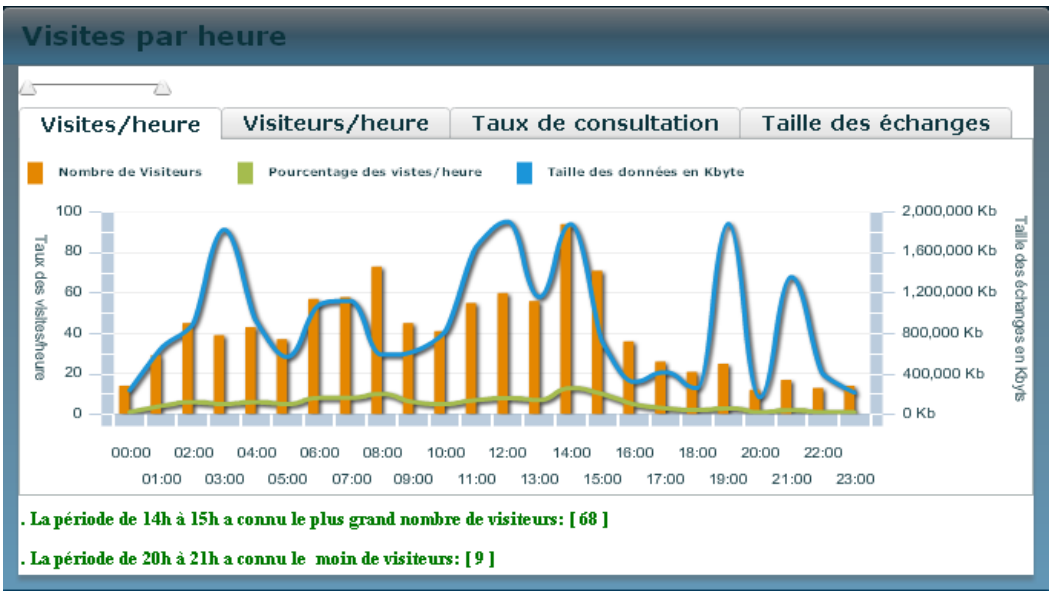


Figure 17 : Fenêtre de visites par heure

La figure 17, montre les caractéristiques des visites par heure de serveur. Les informations issues de ces histogrammes intéressent particulièrement l'administrateur du site qui peut à travers ces informations décider d'augmenter sa bande passante dans une heure précise par exemple et bien d'autres décisions.

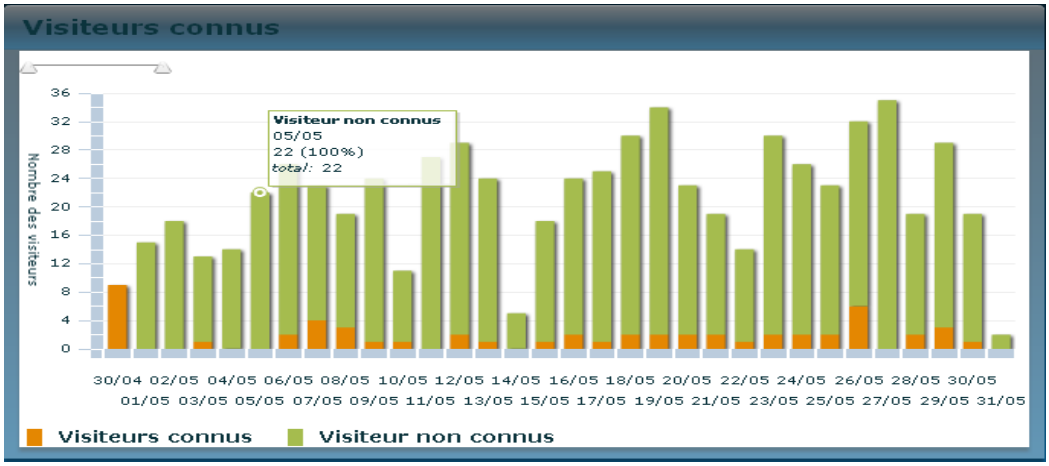


Figure 18 : Fenêtre de taux de fidélité

La figure 18 représente le taux de fidélité des visiteurs, et par la suite on pourra adapter les techniques de fidélisation des visiteurs s'ils ne le sont pas déjà, ou s'ils ne le sont pas suffisamment.

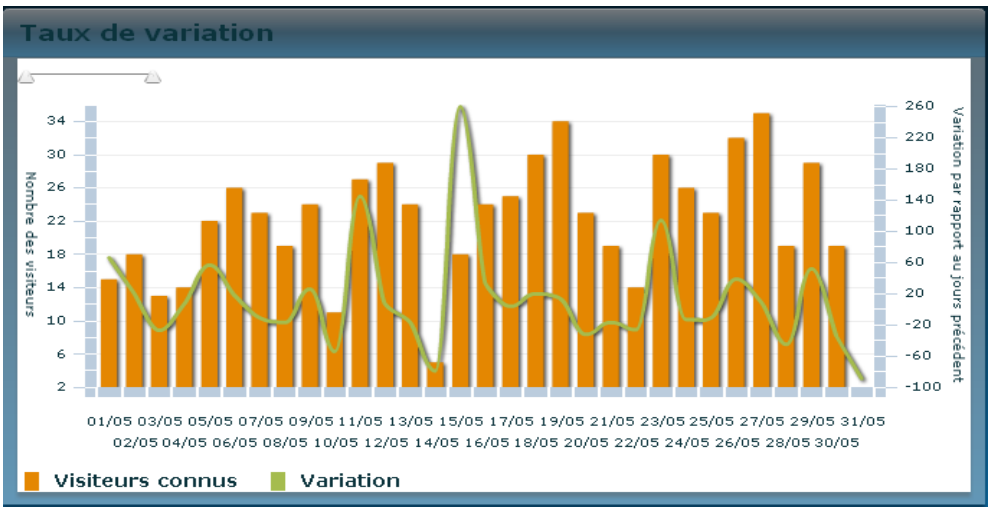


Figure 19 : Fenêtre de la variation de nombre de visiteur par jours

La figure 19 montre un histogramme qui renseigne les décideurs sur les jours où il y a une baisse de nombre de visiteurs et les jours ou il y a une augmentation de ce nombre.

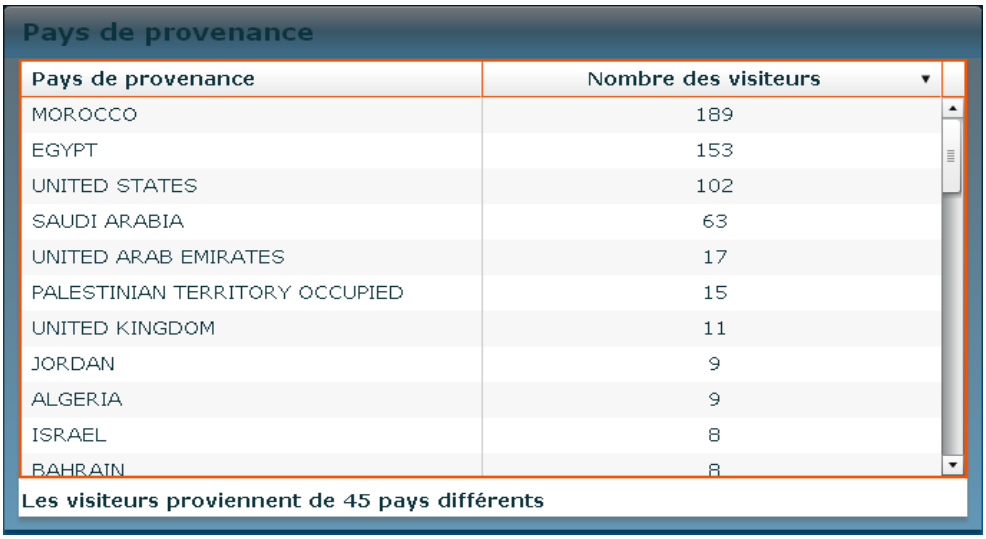


Figure 20 : Fenêtre table de pays de provenance

La figure 20 montre les pays de provenance des utilisateurs. C'est une information très importante pour savoir la popularité du site et sa puissance en ligne.

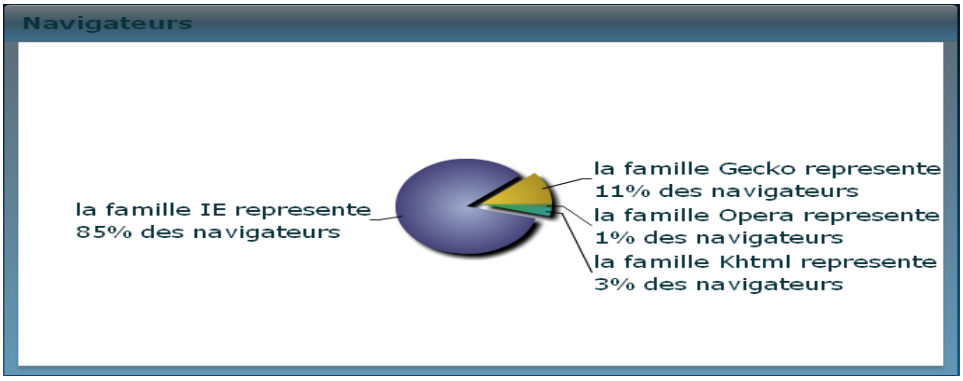


Figure 21 : Fenêtre des types de navigateurs

La figure 21 informe l'administrateur du site sur le type de navigateur le plus utilisé par ces visiteurs. Cette information pourra lui être utile au moment de la conception de son site.

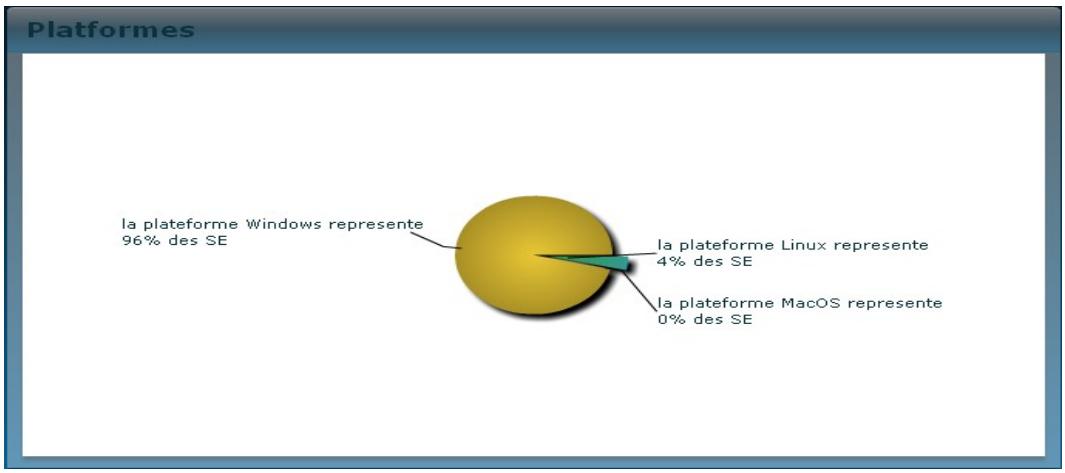


Figure 22 : Fenêtre des types des plateformes

La figure 22 informe le web master sur les types de systèmes d'exploitation utilisés par les visiteurs de son site.

Figure 23 : Fenêtre des différents affluents

La figure 23 est une table qui renseigne sur les sites, à partir desquels, les visiteurs accèdent au site étudié. C'est une information très pertinente pour le responsable marketing plus que d'autres.

Figure 24 : Fenêtre des mots de recherches amenant au site

La figure 24 donne aux web master une idée sur les mots de recherche que les visiteurs de son site utilisent dans les moteurs de recherche avant d'accéder au site.

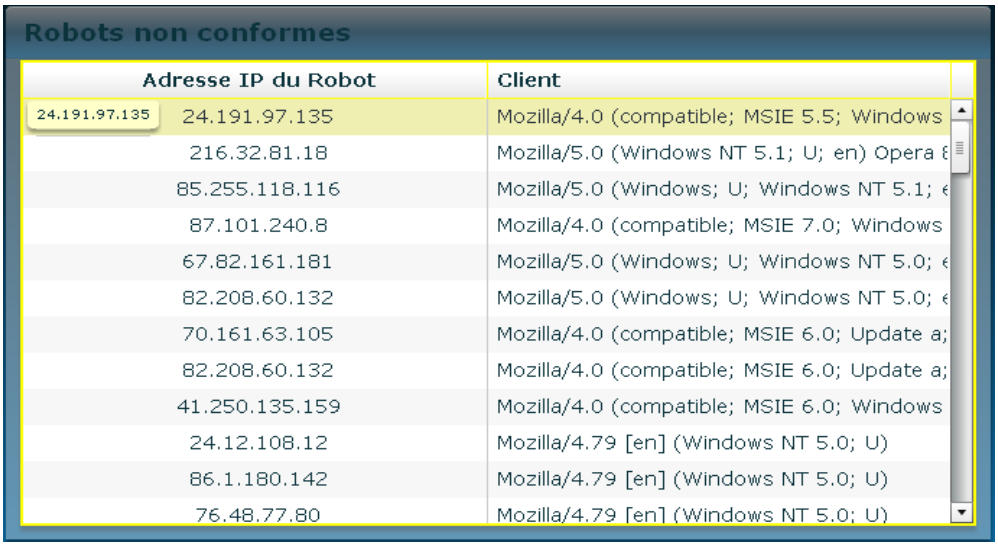


Figure 25 : Fenêtre d'une liste de robots

La figure 25 est une liste de robots, qui parcourt le site et qui ne consultent pas le fichier robots.txt pour voir leurs droits. Ces robots peuvent être bloqués carrément par le web master en indiquant leur adresse IP et leur navigateur dans un fichier dédié pour ça, ce fichier est souvent nommé *htAccess* ou *allowAccess.*

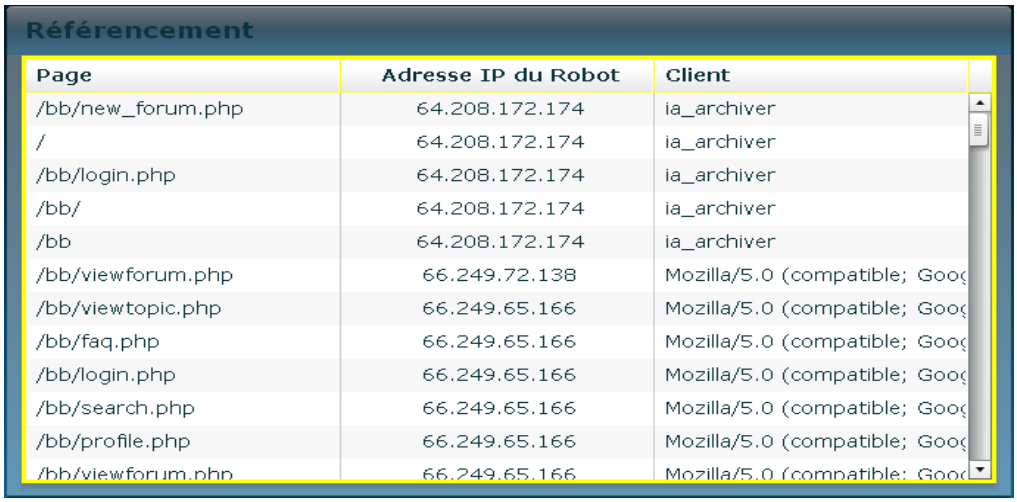


Figure 26 : Fenêtre des pages indexées

La figure 26 indique au web master les pages qui sont indexées et par quel moteur elles le sont. Comme ça si une page confidentielle est connue par un robot, il le saura et prendra les décisions nécessaires.

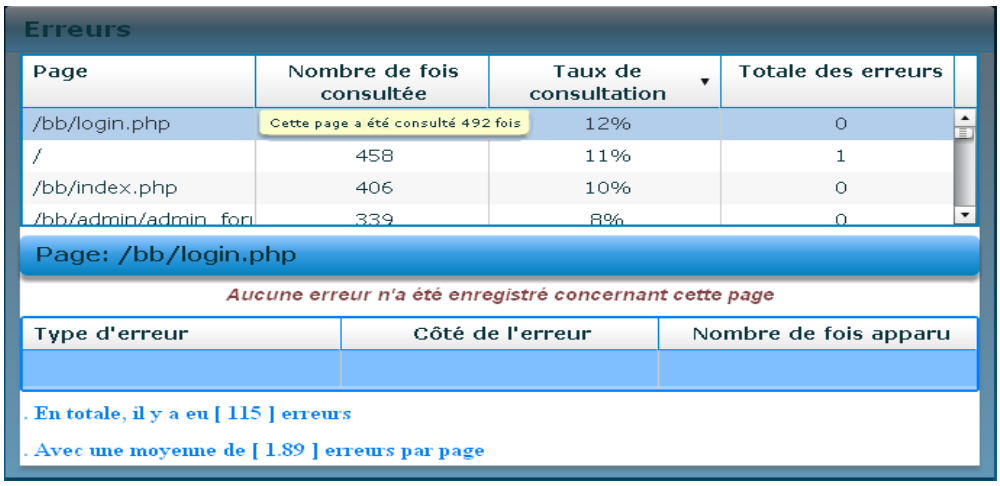


Figure 27 : Fenêtre des erreurs

La figure 27 informe l'administrateur du site sur le nombre de fois qu'une page a été vue par les visiteurs, et combien de fois son statut était un code d'erreur, dans ce dernier cas l'administrateur du site verra en bas de la table une description sur l'erreur ou les erreurs .

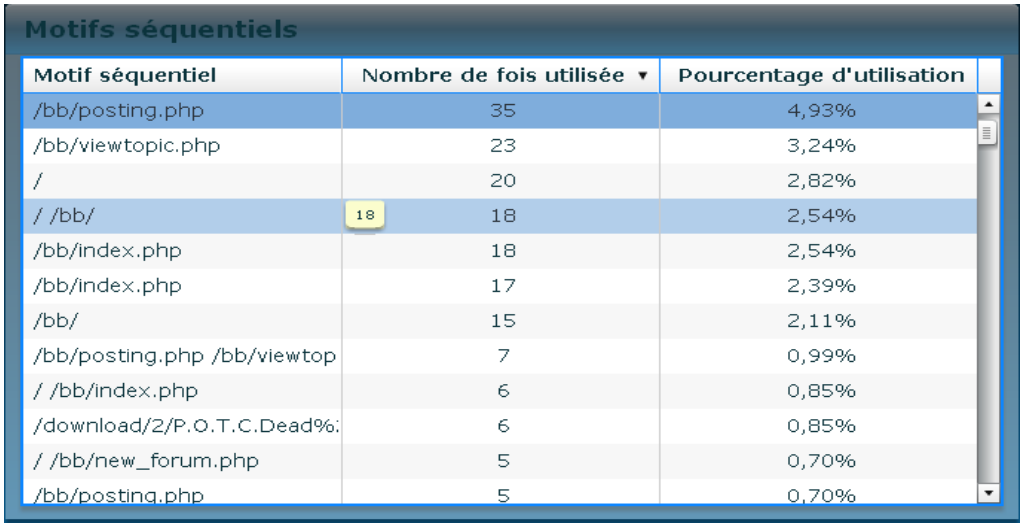
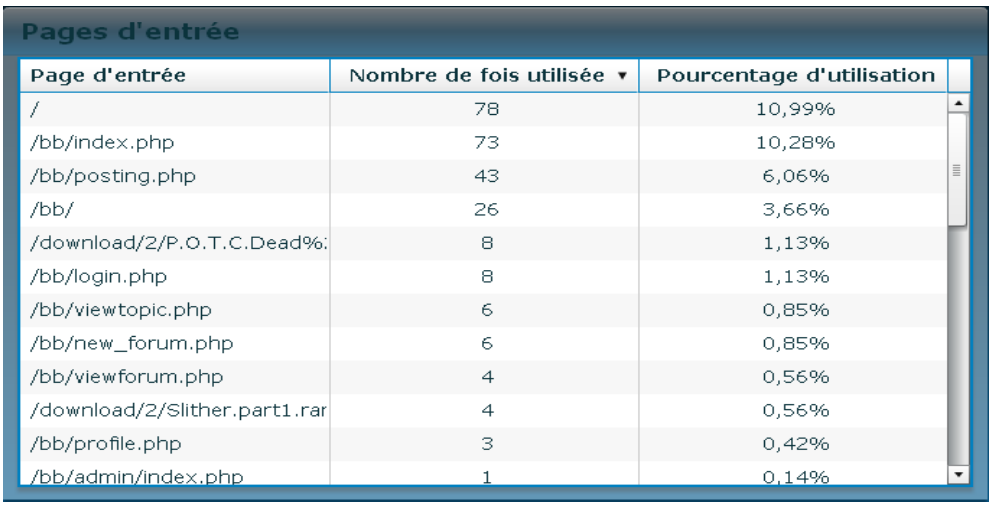


Figure 28 : Fenêtre des motifs séquentiels des visites

La figure 28 est une table dont chaque ligne est une suite de page représentant un nombre précis de visite. Dans la ligne sélectionnée par exemple la règle est la suivante 2,54% des visiteurs qui visite la page « / » vont visiter également et dans l'ordre la page **« /bb ».**

La page « / » est la page principale du site c'est-à-dire la page dont le nom et celui du nom de domaine du site.

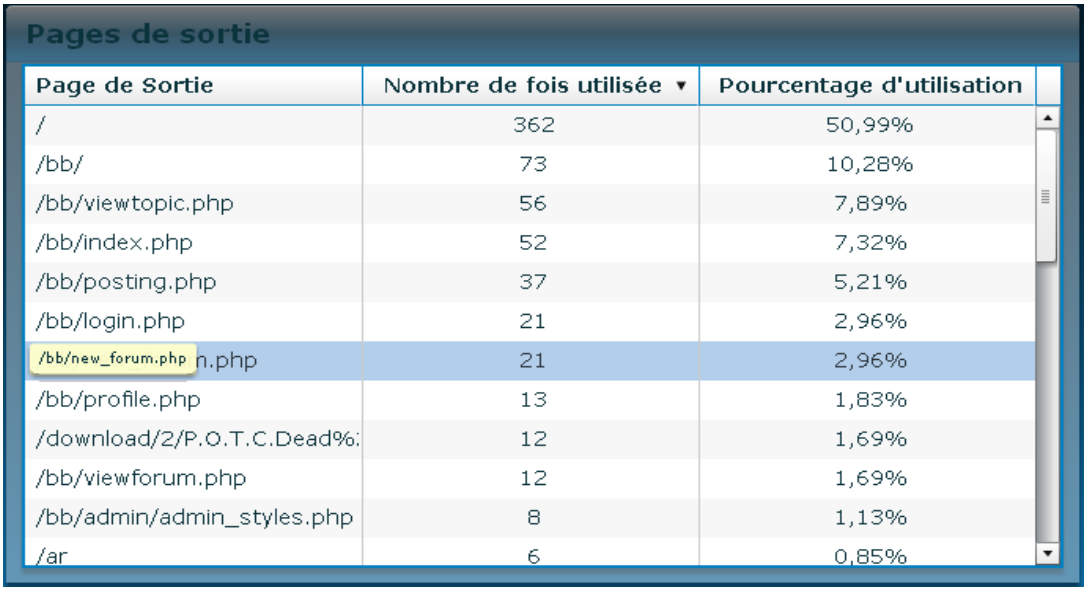


Pages d'entrée

Page d'entrée	Nombre de fois utilisée	Pourcentage d'utilisation
/	78	10,99%
/bb/index.php	73	10,28%
/bb/posting.php	43	6,06%
/bb/	26	3,66%
/download/2/P.O.T.C.Dead%:	8	1,13%
/bb/login.php	8	1,13%
/bb/viewtopic.php	6	0,85%
/bb/new_forum.php	6	0,85%
/bb/viewforum.php	4	0,56%
/download/2/Slither.part1.rar	4	0,56%
/bb/profile.php	3	0,42%
/bb/admin/index.php	1	0,14%

Figure 29 : Fenêtre des pages d'entrées au site

La figure 29 est une table qui reprend toutes les pages du site qui ont été une page d'entrée pour un ou plusieurs visiteurs. Cette information indique au web master la page ou les pages par lesquelles un certains nombres de visiteurs se connectent au site **[6]**.



Pages de sortie

Page de Sortie	Nombre de fois utilisée	Pourcentage d'utilisation
/	362	50,99%
/bb/	73	10,28%
/bb/viewtopic.php	56	7,89%
/bb/index.php	52	7,32%
/bb/posting.php	37	5,21%
/bb/login.php	21	2,96%
/bb/new_forum.php	21	2,96%
/bb/profile.php	13	1,83%
/download/2/P.O.T.C.Dead%:	12	1,69%
/bb/viewforum.php	12	1,69%
/bb/admin/admin_styles.php	8	1,13%
/ar	6	0,85%

Figure 30 : Fenêtre des pages de déconnexion du site

La figure 30 est une table qui reprend toutes les pages du site qui ont été une page de sortie pour un ou plusieurs visiteurs **[7]**. Cette information indique au web master la page ou les pages sur lesquelles un certains nombres de visiteurs se déconnectent du site.

5 Conclusion

Le Web Usage Mining contribue toujours à l'amélioration et au diagnostic de site Web, déployant différentes méthodes et techniques comme l'analyse des fichiers log. Cependant celle-ci commence à être limitée et présente quelques inconvénients :

- Si le site est distribué sur plusieurs serveurs, Il faut mettre en œuvre un outil pour la concaténation des fichiers log.
- impossible de calculer le nombre de visiteurs : adresses IP dynamiques
- Aucun indice ou information, quand un visiteur quitte le site.
- Impossible de calculer le temps passé sur chaque page ou le temps d'une navigation.
- Frame HTML (si une page est constituée de plusieurs frames alors au lieu de calculer « une page vue » on calcule un « multiple de nombre de pages vues »).
- Trafic artificiel généré par les robots et les outils de monitoring.
- Format des informations statistiques difficilement lisible.
- Pour les sites qui ont beaucoup de trafic, les fichiers Log sont plus lourds et plus compliqués à manipuler.
- Possibilité de réécriture du fichier log par un internaute ou par les robots.

Cependant une autre approche est plus intéressante, qui présente des améliorations par rapport à la méthode des fichiers log : « **L'analyse des Tags » [8]**.

L'approche utilisée par la démarche « **L'analyse des Tags »** est différente car elle ne se concentre pas sur le serveur mais sur l'utilisateur. Un code à insérer sur toutes les pages du site va servir de marqueur (il s'agit d'un script écrit souvent en java script). Lorsque la page sera chargée, elle envoie des informations sur la navigation de l'internaute.

L'analyse des Tags se base essentiellement sur l'analyse des adresses IP ainsi que sur l'utilisation de cookies, installés dans le navigateur de chaque ordinateur qui se connecte à un site. Elle ne nécessite aucune installation matérielle (un simple script à insérer dans les pages qu'on veut auditer), on peu aussi démembrer un certain nombre d'avantage **[9]**:

- Pas de problèmes de Proxy : reconstitution des chemins de visiteurs immédiat
- Informations sur la configuration matérielle des internautes
- Suivi du tracking en quasi temps réel.
- Utilisation de cookies permettant d'analyser les visiteurs uniques sur l'ensemble des sites : « audience transversale » (dans le cas d'un site distribué géographiquement).
- Les tags ne peuvent pas être exécutés par les robots
- Information sur le temps de consultation de chaque page.

Références

[1] R. Kimball et R. Merz « Le data webhouse. Analyse des comportements clients sur le Web ». Editions Eyrolles, Paris (2000).

[2] Phillip M. Hallam-Baker**,** Brian Behlendorf, « Extended Log File Format", World Wide Web Consortium, Working Draft WD-logfile-960323, March 1996.

[3] Berry and G. Linoff. « Data Mining Techniques for Marketing, Sales and Customer Support ». New York: Wiley, 1997.

[4] Cooley, R., B. Mobasher, et J. Srivastava (1999). « Data Preparation for Mining World Wide Web Browsing Patterns ». Journal of Knowledge and Information Systems.

[5] Malika Charrad, Mohamed Ben Ahmed, Yves Lechevallier (2005), Extraction des connaissances à partir des fichiers Logs, Actes de l'atelier Fouille du Web des 6èmes journées francophones «Extraction et Gestion des Connaissances ».

[6] Y. Lechevallier, D. Tonasa, B. Trousse, R. Verde. « Classification automatique : Applications au Web Mining ». in Proceeding of SFC 2003, Neuchatel, Swiss. Septembre (2003) pages 10-12.

[7] M. Charrad, M. Ben Ahmed et Y. Lechevallier « Web Usage Mining: WWW pages classification from log files ». Actes de l'atelier Fouille du Web des 6èmes journées francophones, « Extraction et Gestion des Connaissances EGC 2006 » à ENIC Telecom Lille1, Cité Scientifique Lille, France. 17-20 Janvier (2006) pages 41-52.

[8] P. Buche , J. Dibie-Barthélemy , O. Haemmerlé et G. Hignette. « Fuzzy semantic tagging and flexible querying of XML documents extracted from the Web». Journal of Intelligent Information Systems , numéro 26 , pages 25-40. Springer, 2006.

[9] « Quelle technologie de mesure de statistiques choisir », Dossier sur le site : www.web-analytique.com/les-dossiers/quelle-technologie-de-mesure-de-statistiques-choisir-.html