REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire

En vue de l'obtention d'un Master en informatique spécialité Systèmes Informatiques

Thème :

Détection de parcours à travers les mesures de similarité sémantique de visite

Proposé et dirigé par :

Mlle S.AIT-ADDA

Réaliser par :

Mr EL-KECHAI Nadir
Mr HAMEG Mohamed

Promotion : 2011/2012

Sommaire

Introduction générale :

Pour notre mémoire de fin d'étude, nous nous sommes intéressé aux environnements d'apprentissage et plus précisément aux apprenants et a leur donnée d'usage du web. Notre travail portera sur l'étude de ses donnée et leur interprétation en vue d'identifier le type de parcours et les options que le tuteur pourrai avoir pour palier soit a une défaillance du système d'apprentissage, soit a des lacune d'un ou plusieurs apprenants ou encore a une manque d'information ou a une mauvaise structuration de ces dernières.

A cet effet nous seront appelés a utiliser plusieurs technologie de l'information, et quelques outils pour l'obtention des données et pour formaliser les solutions. Dans le premier chapitre, nous parlerons des traces, qui repesent dans notre travail l'essentiel des donner a traiter, qu'il faudra d'abord récupérer.

Dans le second chapitre, nous introduirons des notions de base des métadonnées et les ontologies, qui constituent l'outil principal pour rendre les données récupérées utilisable et exploitable. En effet, les métadonnées serviront de point de repère pour extraire les données utiles et ne pas s'encombrer des données superflues dont on n'a pas besoin. Les ontologies nous permettrons de décrire les concepts d'un cours proposé au niveau d'un environnement d'apprentissage, au quel seront comparé les résultats du traçage d'un apprenant pour en déduire le type de parcours.

Dans le troisième chapitre, nous décrirons notre démarche de travail pour la récupération des trace d'usage du web pour un apprenant, le traitement de ces données, leur structuration, ainsi que le traitement appliquées a ces données pour en extraire le type de parcours.

Dans le dernier chapitre nous proposons une description de la conception notre application qui permet de définir le parcours de visite d'un apprenant et de reconnaitre les difficultés rencontré et de marquer les concepts non bien assimilés pendant son apprentissage et ceci à travers le recueil et l'interprétation des traces d'interactions durant une session d'apprentissage

Chapitre I – Les traces

1. Introduction :

Les traces sont les informations résignant sur les diverses activités qui se déroule sur un terminal. Plus concrètement, elles se présentent sous forme de fichier journal, ou est enregistré l'activité de l'utilisateur, cela est régit généralement par le système d'exploitation, mais aussi par des logiciels dédiés a cet effet. Ces journaux (aussi appelé « *fichier journal »* ou « *trace brut*» ou en anglais « *log file »*) contiennent des dates, des heures, des durée, des actions (copier, coller, recherche …) aussi des liens et des chemins de fichier. Ces fichier s'avère être très utiles dans les environnements d'apprentissage qui seront bien évidement définis dans le

Depuis quelques années, les systèmes d'apprentissage ont beaucoup évolué et le nombre d'utilisateurs de ces systèmes ne cesse de croitre.

Toutefois, dans les formations ouvertes et à distance, l'enseignant ne se trouve pas face--face avec ses apprenants ainsi le suivi et l'adaptation de la présentation du contenu devient de plus en plus difficile. Car il ne peu déterminer, s'ils ont bien assimilé les connaissances présentées. Il n'est pas non plus en mesure de savoir si la démarche entreprise dans la construction du cours s'adapte au niveau des apprenants.

De ce fait, l'analyse des activités des apprenants dans les environnements d'apprentissages est devenue un thème de recherche dynamique. Cette analyse vise plusieurs objectifs : comprendre et suivre les apprentissages d'un apprenant ou d'un groupe d'apprenants, qualifier l'utilisation, l'utilisabilité et l'acceptabilité du système pour le rendre plus adaptatif.

Pour atteindre ces objectifs, de nombreuses recherches sur l'analyse et l'interprétation des activités réalisées par les apprenants durant leurs interactions avec l'environnement de formation ont été menées. Ces travaux portent sur le recueil et l'interprétation en cours de session, d'informations appelées traces. Ces traces seront filtrées puis structurées en données de plus haut niveau.

L'élément de complexité de cette méthode est la collecte, le filtrage et la structuration de trace brute, pour une interprétation pertinente, qui permet soit à l'être humain ou bien à un agent artificiel de suivre, d'évaluer ou de réguler la situation d'apprentissage. Nous commençons par définir dans ce chapitre le concept de trace, sa classification et les objectifs qui l'entourent.

2. Contexte de travail :

2.1. Définition des EIAH :

Notre travail ‘articule autours des environnements d’apprentissage, le terme le plus utilisé est Environnement Informatique d’Apprentissage Humain (EIAH),

Les environnements informatiques pour l'apprentissage humain (EIAH) sont des environnements informatiques qui ont pour objectifs de favoriser ou susciter des apprentissages, de les accompagner et de les valider.

La recherche dans ce domaine est née avec l'informatique mais se sont surtout développées dans le sillage de l'intelligence artificielle.

Le terme EIAH est né dans les années 90, avec le souhait de souligner l'interaction entre les deux pôles source de la complexité du projet technologique et scientifique : l'informatique (avec la modélisation computationnelle qu'elle exige et son inscription matérielle) et l'apprentissage humain (pour lequel on ne dispose encore que de modèles très partiels). La recherche sur les EIAH est fondamentalement pluridisciplinaire, en appelant à la coopération de différents secteurs de l'informatique (génie logiciel, réseau, la modélisation des connaissances et des interactions, etc.),

2.2. Problématique :

De nombreux Environnements Informatiques d’Apprentissage Humain (EIAH) se sont appuyés principalement sur la détection des caractéristiques relatives aux connaissances, aux intérêts, aux objectifs, aux pré-requis et aux traits individuels pour le suivi et l’adaptation des contenus. Cependant, l’identification de ces caractéristiques est un problème difficile dans le domaine de l’enseignement à distance. En effet, l’observation de l’apprenant, est rendue difficile par l’absence du contact face-à-face. Par conséquent, les recherches se sont orientées vers l’analyse du comportement de l’apprenant dans l’environnement d’apprentissage pour remplacer, en partie, l’observation de son activité. Cette analyse est basée sur l’interprétation d’informations recueillies pendant la session d’apprentissage, appelées traces.

Durant son apprentissage, une bonne compréhension du comportement de l’apprenant peut apporter plusieurs avantages à tous les acteurs qui participent au processus d’apprentissage. Du coté concepteur du cours, le retour obtenu par l’analyse peut conduire à la remise en cause et à l'adaptation d'un scénario d’apprentissage par une révision ou enrichissement. Du coté apprenant, il est possible d’obtenir plus d’aide pour l’activité en cours par l’amélioration des approches d’assistance en ligne. Enfin, du coté tuteur, cette approche permet de lui apporter plus d’aide en détectant les apprenants en difficulté au bon moment et sur les concepts non ou pas bien maitrisés.

3. Trace :

3.1. Définition

La définition de la notion de trace diffère selon son rôle et son utilisation dans chaque domaine de recherche.

[1] définissent une « trace d'interactions » comme « tout objet informatique dans lequel s'accumulent des données à propos des interactions entre un système informatique et son utilisateur ».

Dans le cadre spécifique des environnements d'apprentissage, [2] définissent la trace d'apprentissage comme « une observation ou un enregistrement de l'interaction de l'apprenant avec un système en vue d'une analyse ».

J-P. Pernin [3] définit une trace d'apprentissage comme un indice de l'activité des acteurs d'une situation d'apprentissage, qu'elle soit ou non instrumentée. Il complète, par ailleurs, sa définition en précisant qu'il s'agit d'un résultat obtenu au cours ou au terme d'une activité, d'un événement ou d'un ensemble d'évènements relatifs au déroulement de la situation d'apprentissage. Dans une optique légèrement différente, P-A. Champin [4] parle d'une séquence d'états et de transitions représentant l'activité de l'utilisateur : «la séquence temporelle des objets et opérations mobilisés par l'utilisateur lorsqu'il utilise le système est appelée trace d'utilisation ».

Dans ces deux définitions, une trace est une trace d'activité, d'utilisation, d'interaction. Il ne lui est pas associé d'interprétation sur la situation d'apprentissage. On parle alors de traces primaires, brutes, de base ou de « bas niveau ». Il ressort également de ces définitions qu'une trace est temporellement marquée, plus particulièrement lorsque Champin parle de séquence temporelle.

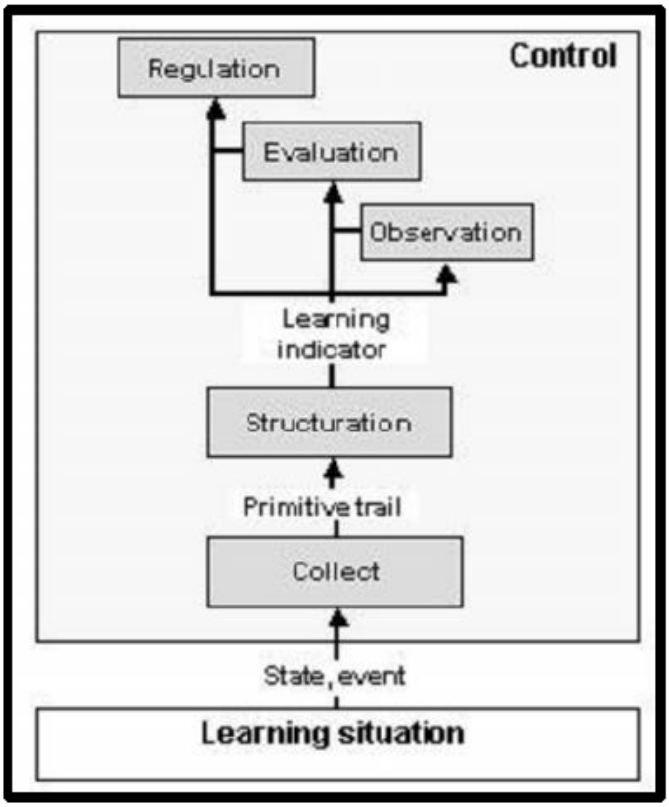


Figure 1: Le processus générale de gestion de trace [3].

Une trace d'apprentissage peut être temporaire si l'observation est directement faites puis traitée par un être humain ou enregistrée si elle est collectée et mémorisée à l'aide d'un instrument technique (papier- crayon, vidéo, caméra, magnétophone, ordinateurs …etc.).

Après être collectées, les traces enregistrées sont structurées dans un niveau d'informations plus évolué afin d'être exploiter pour l'observation, l'évaluation et la régulation, comme illustré sur la figure 1. L'observation dans le cas de pernin [3] est les différentes résultats de la structuration et qui ne portent aucune interprétation (ex :durée moyenne de consultation d'une activité).

3.2. Types de traces :

Les traces se présentent sous des formes de plus en plus variées, incluant des relations riches et des possibilités de navigation poussées : allant des simples clicks de la souris, URLs visitées et temps de consultation jusqu'à la voix enregistrée, le parcours chronologique au sein d'une activité, l'usage des services de communication dans l'espace pédagogique et les réponses aux exercices et questionnaires.

On regroupe ces traces suivant le contenu de l'information qu'elles portent. En conséquence, quatre groupes de traces se dégagent :

- Les traces informatives (les informations personnelles ex. nom, prénom, etc.) et les informations techniques (ex. adresse IP, navigateur, etc.).
- Les traces liées à l'exploitation d'une ressource (référence de la ressource, origine et historique des accès, etc.).
- Les traces associées à l'activité d'apprentissage (temps de réponse, résultats de tests, etc.).
- Les traces attachées à l'activité de communication (contenu d'un message, destinataire, etc.).

La définition de ces quatre groupes permet d'envisager un traitement unique pour l'ensemble des traces associées à un groupe. Les traces concernant la communication dans un groupe permettent d'évaluer la qualité de l'interaction. Aussi, les traces relatives à un historique de parcours peuvent permettre de dégager des profils d'apprenants. Les traces faisant référence au temps passé sur une activité permettent d'évaluer les compétences d'un apprenant, etc.

Suite à cette première classification, on définit encore trois classes d'exploitation :

- Caractère de la situation (individuelle ou collective).
- Valeur d'usage (qualitative ou quantitative).
- Cadre d'exploitation (étude du comportement ou étude de la connaissance).

Le choix des données à observer dépend de l'environnement d'interaction, et des objectifs de l'observation, que nous découvrons ci-après.

Objectifs de l'observation:

Dans le cadre des environnements d'apprentissage, l'objectif principal de l'observation est d'être un outil de support et de gestion pour l'apprenant. Toutefois, nous pouvons distinguer divers objectifs :

- Caractériser l'activité de l'apprenant et analyser son comportement.
- Interpréter l'interaction de l'apprenant avec le système.
- Identifier des comportements communs chez les apprenants.
- Déterminer le niveau de compréhension des apprenants au contenu proposé.

On regroupe ces objectifs en trois objectifs principaux, qui peuvent également se suivre comme trois étapes d'exploitation possibles des traces : l'observation (ou la validation des expérimentations et des hypothèses), l'évaluation et la régulation de l'apprentissage. Selon son rôle dans le processus d'apprentissage, chaque acteur (tuteur, enseignant, responsable de module/formation …) serait capable de : *(i)* analyser les données observées (activité d'observation) ; *(ii)* établir un diagnostic à partir des résultats

observés (activité d'évaluation) *;* et *(iii)* agir sur le processus d'apprentissage (activité de régulation). Par exemple, l'analyse des traces peut être effectuée par :

- un tuteur[1] désirant obtenir une vue de l'activité d'un apprenant afin d'évaluer les comportements et réguler le processus d'apprentissage.
- par un apprenant souhaitant analyser sa propre activité dans le cadre d'une autorégulation.
- par un chercheur souhaitant obtenir une vue sur des indicateurs d'apprentissage (définition chapitre afin d'analyser en détail les résultats obtenus et valider ses hypothèses,
- par d'autres acteurs (parents, institutions…), s'intéressant au déroulement de la formation.

Cette diversité met clairement en évidence la nécessité d'aider et d'assister les utilisateurs des traces dans leurs activités de production, d'exploitation et d'analyse. Pour cela, les traitements à appliquer aux traces, depuis leurs collectes, sont formalisés à l'aide de Systèmes à Base de Traces (SBT).

4. Systèmes à base de traces:

Plusieurs travaux de recherche se sont centrés sur le processus d'observation de la trace des utilisateurs dans un enseignement à distance. Malgré les légères différences trouvées dans le nombre de phases ou d'appellations attribuées aux étapes de ce processus, d'un travail à l'autre, dans la littérature, trois phases principales : la collecte, souvent suivie d'une étape de prétraitement, l'analyse et l'exploitation.

Pour permettre et faciliter la manipulation des traces, [5] proposent de formaliser ce processus à travers un système à base de traces (SBT). Un SBT est composé de plusieurs modules interdépendants, comme le montre cette figure :

[1] Tuteur : personne physique, expert d'un domaine de travail ou d'une partie, en charge de la formation a distance de personnes.

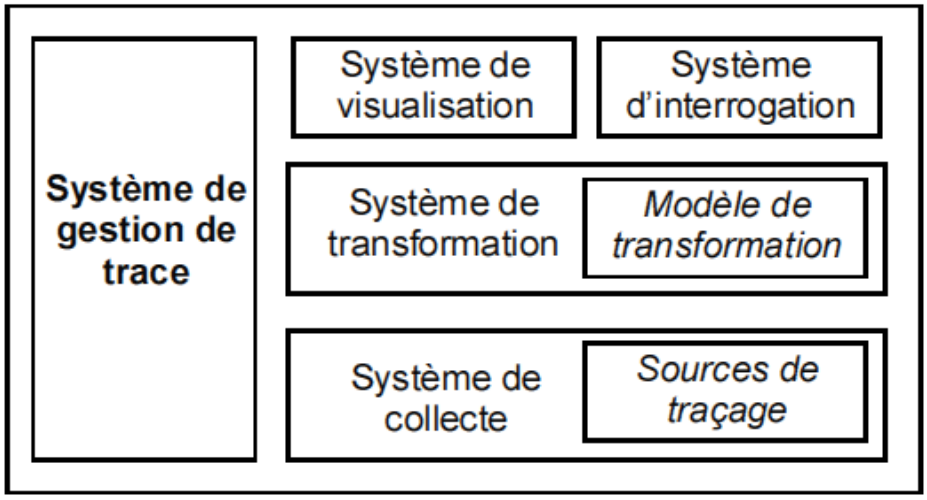


Figure 2. Schéma simplifié d'un Système à Base de Traces.

Le système de collecte capture les interactions par l'intermédiaire de sources de traçage (logiciels dédiés comme « *mini keylogger* », fichier journal système, historique des navigateurs …), et crée une première trace. Le système de transformation constitue le cœur du SBT. Il permet de générer de nouvelles informations à partir des traces collectées. Le choix du modèle de transformation à appliquer dépend de l'intention d'utilisation de cette trace. L'ensemble des traces collectées et transformées est alors accessible par l'intermédiaire d'un système de requête et d'un système de visualisation. Ce système de visualisation doit donner la possibilité d'avoir une vue sur les traces du SBT, afin de permettre l'analyse et l'interprétation de celles-ci. La visibilité des résultats pour l'utilisateur dépend de l'objectif de l'observation :

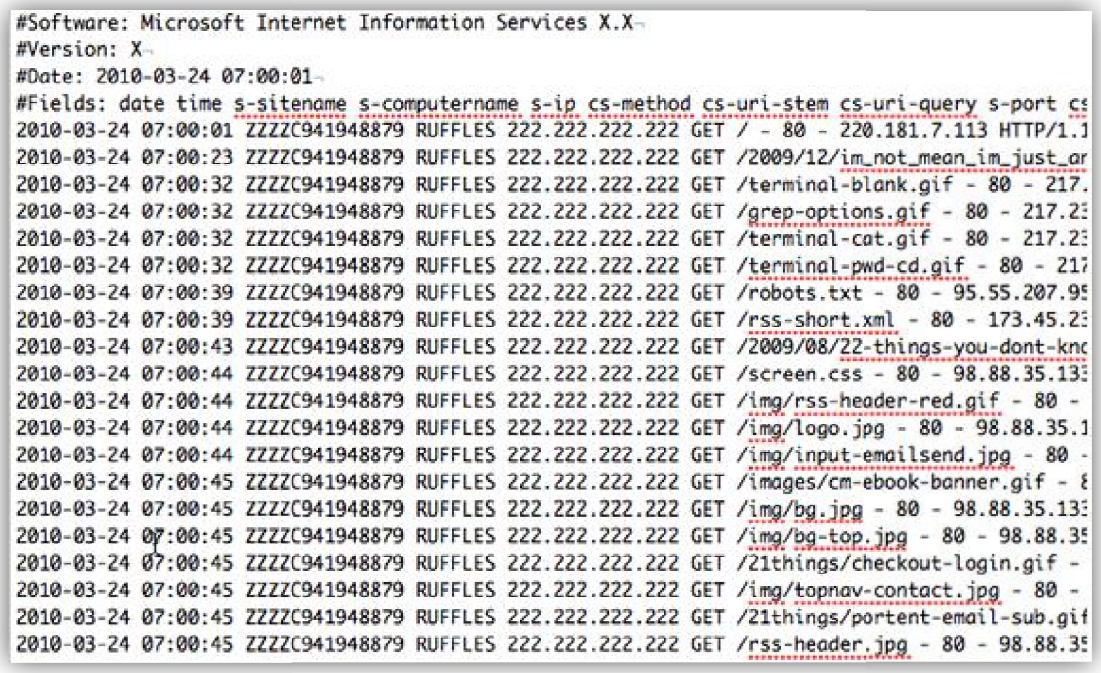
- Sans visualisation.
- Avec visualisation lors de l'activité.
- Visualisation après l'activité,

4.1. Sources d'observation:

Environ 78 % [6] des environnements d'enseignement intègrent un outil de capture des traces numériques. Ces systèmes utilisent différentes approches de collecte selon les sources d'observation : le serveur, le poste client, ou des mécanismes d'observation spécifiques au système d'apprentissage.

4.1.1. Approches centrées serveur

L'approche centrée serveur s'intéresse à la recherche des motifs de navigation des utilisateurs sur un site donné en se basant sur l'analyse des *logs* des serveurs Web. Ces *logs* contiennent l'ensemble des actions effectuées sur le serveur. La création des traces à partir de ces *logs* est un processus complexe qui nécessite de nombreuses opérations (filtrage, recomposition en sessions, etc.). De nombreuses informations peuvent ainsi être déterminées, telles que le navigateur et le système d'exploitation utilisés, le moment auquel une certaine activité a été effectuée, la durée passée sur une certaine page, le nombre de fois qu'une page spécifique a été visitée, ou encore l'adresse IP correspondant à chacun des événements.



```
#Software: Microsoft Internet Information Services X.X
#Version: X
#Date: 2010-03-24 07:00:01
#Fields: date time s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query s-port cs
2010-03-24 07:00:01 ZZZZC941948879 RUFFLES 222.222.222.222 GET / - 80 - 220.181.7.113 HTTP/1.1
2010-03-24 07:00:23 ZZZZC941948879 RUFFLES 222.222.222.222 GET /2009/12/im_not_mean_im_just_ar
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-blank.gif - 80 - 217.
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /grep-options.gif - 80 - 217.23
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-cat.gif - 80 - 217.23
2010-03-24 07:00:32 ZZZZC941948879 RUFFLES 222.222.222.222 GET /terminal-pwd-cd.gif - 80 - 217
2010-03-24 07:00:39 ZZZZC941948879 RUFFLES 222.222.222.222 GET /robots.txt - 80 - 95.55.207.95
2010-03-24 07:00:39 ZZZZC941948879 RUFFLES 222.222.222.222 GET /rss-short.xml - 80 - 173.45.23
2010-03-24 07:00:43 ZZZZC941948879 RUFFLES 222.222.222.222 GET /2009/08/22-things-you-dont-knc
2010-03-24 07:00:44 ZZZZC941948879 RUFFLES 222.222.222.222 GET /screen.css - 80 - 98.88.35.133
2010-03-24 07:00:44 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/rss-header-red.gif - 80 -
2010-03-24 07:00:44 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/logo.jpg - 80 - 98.88.35.1
2010-03-24 07:00:44 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/input-emailsend.jpg - 80 -
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /images/cm-ebook-banner.gif - 8
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/bg.jpg - 80 - 98.88.35.133
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/bg-top.jpg - 80 - 98.88.35
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /21things/checkout-login.gif -
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /img/topnav-contact.jpg - 80 -
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /21things/portent-email-sub.gif
2010-03-24 07:00:45 ZZZZC941948879 RUFFLES 222.222.222.222 GET /rss-header.jpg - 80 - 98.88.35
```

Figure 3: exemple fichier log

Cette approche est généralement utilisée dans les sites à vocation commerciale, souhaitant disposer de données précises sur leur fréquentation, afin de savoir quelles pages sont les plus visitées, comment les utilisateurs y arrivent, et comment les garder plus longtemps sur le site.

Dans le contexte de l'enseignement en ligne, si la plate-forme d'apprentissage est hébergée sur un serveur apache, par exemple, les données, présentent dans le fichier de *log*, peuvent être utilisées pour : évaluer l'efficacité d'un cursus en ligne ; quantifier les interactions entre les utilisateurs et les pages du cours ou de l'environnement de formation, etc. Ces données peuvent ainsi répondre à certaines questions: les pages *Web* du cours sont-elles adaptées au navigateur le plus utilisé par les apprenants ? Les apprenants accèdent-ils facilement aux pages essentielles du cours ? Une page spécifique est-elle

réellement nécessaire à ce cours ? Le temps passé par les apprenants sur une page particulière est-il adapté pour atteindre l'objectif visé ? Etc.

4.1.2. Approches centrées utilisateur

Si, pendant un exercice, l'apprenant effectue des recherches sur le Web, cette interaction n'est pas observée sur le serveur. Pourtant cette interaction peut être un élément important d'explication du parcours de l'apprenant, d'ailleurs notre travail est axé sur cet aspect de la formation. Il est donc intéressant d'instrumenter le poste client afin d'observer toutes les interactions propres à l'apprenant. Cette démarche n'est pas largement utilisée. En effet, si le recueil de données de trafic centrées-serveur est maintenant relativement standardisé, la collecte d'informations au niveau des postes utilisateurs est encore un domaine d'activité [7] Il est possible de recueillir es information par le biais d'une application dédiés au traçage.

4.1.3. Approches basées sur des logiciels spécifiques

Au-delà des approches qui distinguent la source d'observation, selon que cela soit fait sur le poste client ou sur le serveur, d'autres approches se focalisent sur l'identification de l'interaction au moment de la collecte à travers un outil spécifique à l'environnement tracé.

Parmi ces travaux, nous pouvons citer e-Médiathèque, un outil de travail collaboratif où toutes les interactions de l'utilisateur sont observées et traitées à partir de deux modèles [8] : l'un définit les objets observables (les outils mis à disposition tels que le navigateur Internet, la messagerie instantanée, ou les ressources telles que les textes, les images, etc.) ; et l'autre spécifie les actions réalisables par l'utilisateur (création, modification, suppression de contenus, etc.). Les traces sont ensuite créées en fonction des activités des utilisateurs, stockées dans un composant interne de l'application, et affichées en temps réel à l'utilisateur.

Dans le cas du *Knowledge Pool System* (KPS) de la fondation ARIADNE, chaque interaction de l'utilisateur avec l'outil *SILO* (l'interface entre les utilisateurs et le KPS) est enregistrée dans un fichier au format XML qui est centralisé dans chacun des viviers institutionnels qui constituent le KPS. Chaque enregistrement donne des informations, telles que le titre des ressources ayant été indexées, téléchargées ou supprimées, l'utilisateur à l'origine de l'enregistrement, la date de création de l'enregistrement, etc.

Si les approches spécifiques à un outil particulier présentent certains atouts qui comblent les lacunes des approches fondées sur les fichiers de *log*, elles souffrent de leur cloisonnement et de leur aspect propriétaire. Les traces collectées par les systèmes mentionnés ci-dessus présentent un format qui leur est spécifique et qui empêche leur traitement par d'autres systèmes à base de traces (SBT).

Synthèse : A travers cette étude, nous constatons que chaque approche dispose de certains avantages mais aussi des inconvénients.

Les logs sur les serveurs Web souffrent du manque d'informations sur l'activité exécutée sur le poste de l'apprenant, en dehors du serveur en question, alors que les logiciels spécifiques n'enregistrent que les interactions effectuées dans un environnement dédié et selon un format propriétaire. A contrario, les approches fondées sur l'analyse des fichiers de log côté client peuvent pallier ces inconvénients. Elles permettent la collecte de traces mêmes dans des systèmes hétérogènes. Ces traces renseignent, non seulement sur l'activité de l'apprenant au sein du cours ou de la plate-forme de formation, mais également en dehors de celle-ci, et fournissent des informations sur les moments d'inactivité. C'est, pour cela, que nous optons pour cette approche dans la suite de notre mémoire. Avant de présenter nos choix, nous poursuivons la description des étapes de traitements des traces car repose sur s outils et moyens sen notre disposition pou arriver au but final.

5. Déroulement de l'observation :

Comme Jermann [2] le précise, les traces sont enregistrées en vue d'être analysées. Mais avant de pouvoir les exploiter, il faut franchir plusieurs étapes.

5.1. Etape 1 : Collecte de trace

La qualité et la nature des traces enregistrées sont étroitement liées aux outils techniques utilisés afin d'assurer leurs collectes. Il existe trois modes principales comme designer sur cette figure 4 :

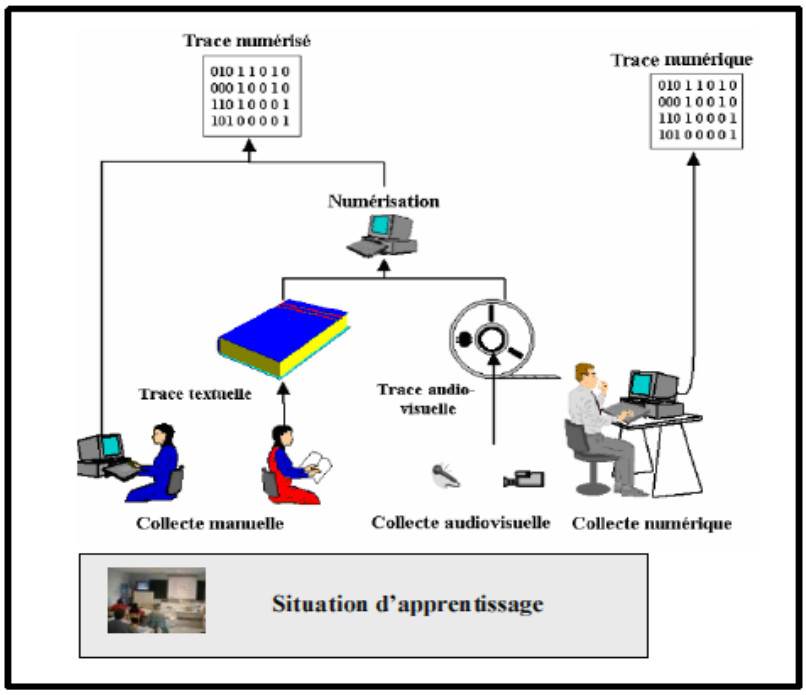


Figure 4: Mode de collecte de trace [Per,05].

Une collecte manuelle : qui est procédé par un observateur humain, acteur ou non dans la situation d'apprentissage avec l'outil papier crayon et l'éventuelle aide de quelque logicielle tel que Word, Excel …etc.

Collecte audiovisuelle : exécutée par un outil capable de créer des enregistrements visuels et audio de la situation d'apprentissage. Collecte numérique : menée en utilisant un environnement informatique sauvegardant.

l'activité de l'apprenant, tel que les outils de traçage. Les résultats de la collecte numérique est une trace numérique.

Le mode de collecte manuelle et audiovisuelle peut être complété par les outils d'acquisition et de numérisation permettant en conséquence de créer des traces numériques. L'observation peut être directe si la collecte de donnée est faite avec la progression de la situation de l'apprentissage ou indirecte si la donnée est collectée après l'expérience d'apprentissage par le biais d'une enquête exécutée auprès des acteurs de la situation d'apprentissage.

5.2. Etape 2 : Structuration de trace

Les traces d'apprentissage sont utilisées par l'enseignant, tuteur, apprenant, chercheur analyste…etc. ; pour observer, évaluer ou réguler la situation d'apprentissage. Ces différents types d'utilisation exigent une structuration de trace d'apprentissage en

indicateurs de haut niveau, réellement pertinents pour effectuer le diagnostic et aboutir à une décision plus adaptée.

5.3. Etape 3 : Exploitation de trace

Après avoir été structurées et traitées, les traces primaires et les indicateurs peuvent enfin être exploités. Les différentes données découlées depuis le traitement de traces sont analysées, et c'est en fonction de ces rôles dans une situation d'apprentissage que chaque acteur pourra [9]:

- Analyser et adapter les caractéristiques identifiables (activités d'observation).
- Établir un diagnostique sur les résultats et les progrès observées (activités d'évaluations).
- Agir sur la situation d'apprentissage (activité de régulation).

Nous proposons ici de décrire chacun de ces trois types et les rôles reliés.

Pour le premier type d'exploitation, l'identification des besoins et des manques des apprenants, il est possible d'exécutée l'observation :

- Par l'enseignant tuteur prévoyant obtenir des vues synoptiques de l'apprenant ou l'activité d'un groupe et les résultats dans le but de réguler la situation d'apprentissage.
- Par l'apprenant ou un groupe d'apprenants désirant l'analyse de leur propre parcours pour s'auto réguler.
- Par le chercheur demandant obtenir des vues synoptiques des indicateurs d'apprentissages pour faire une analyse précise sur les résultats obtenus.
- Par d'autres acteurs (parents, institutions…), plus tangentiel pour l'expérience d'apprentissage s'intéressant sur le prélèvement de traces d'apprentissage.

Le second type d'exploitation de trace concerne l'évaluation de la situation d'apprentissage. Un large spectre d'objectifs existent dont :

- L'estimation des performances des apprenants par l'enseignant à partir des indicateurs d'observation.
- L'évaluation par le tuteur des résultats observés et le comportement dans le but d'adapter l'organisation des activités d'apprentissage.

L'évaluation de l'apprenant par ses propres résultats dans l'objectif de s'orienter dans la formation avec l'approche d'autorégulation. L'évaluation par l'enseignant concepteur du scénario prescrit pour la situation d'apprentissage afin de le comparer avec les résultats aboutis par l'apprenant.

L'analyse par le chercheur analyste pour valider ces hypothèses prédéfinies pour son expérimentation sur des taches qu'il a affectées aux apprenants.

Le dernier type d'exploitation est la régulation, faisant face aux modifications des conditions sur le progrès de la situation d'apprentissage, en tenant compte bien sûr des acteurs de l'activité, précédés par des moyens d'observation et d'analyse d'indicateurs.

Les modifications peuvent être superficielles et consistent uniquement à fournir un ensemble de feedbacks adaptables pour l'apprenant sans pour autant modifier profondément l'organisation des activités d'apprentissage. Dans ce cas, l'acteur principal concerné sera le tuteur en se chargeant d'animer la situation d'apprentissage.

Un autre type de régulation, consiste à redéfinir l'organisation des activités d'apprentissage en fonction des événements annotés (organisation adaptée) ou en déléguant en permanence une planification de taches d'apprentissages depuis les données collectées dans la phase précédente (organisation dynamique).

6. Classification de trace d'apprentissage

Pour classer les traces, Gwenegan Rozé [10] a regroupé les traces suivant le contenu de l'information qu'elles portent. En conséquence, il a dégagé quatre groupes de traces:

Le premier groupe contient les traces portant des informations d'identification : les informations personnelles (nom, prénoms, âge, classe, ...etc.) et les informations techniques liées au support utilisé (version du système, adresse IP, navigateur, système d'exploitation, ...etc.).

Dans un deuxième groupe, il réuni les traces liées à l'exploitation d'une ressource : nom, référence de la ressource, nombre d'accès, durée de consultation, origine de l'accès et historique du parcours des ressources.

Le troisième groupe contient les traces relatives à l'activité d'apprentissage : qualité d'une production, temps de réponse, résultats de tests et réalisation d'actions.

Quant au dernier groupe, il réunit les traces liées à l'activité de communication : nombre de messages envoyés et lus dans un forum, contenu d'un message, destinataire, fréquence des messages sur une période donnée. On trouve également dans ce groupe, l'affectation d'un rôle à un acteur de la situation, la demande d'aide en ligne et la demande d'assistance au tuteur. La définition de ces quatre groupes permet d'envisager un traitement unique pour l'ensemble des traces associées à un groupe.

A la suite de cette première classification, il fallait chercher à définir une distinction dans les cas d'usage des traces collectées. Alors Gwenegan Rozé [10] défini encore trois classes d'exploitation.

La première classe permet de caractériser la situation d'apprentissage dans laquelle la trace a été produite : situation d'apprentissage collectif (par exemple : le nombre de

messages postés sur un forum) ou situation d'apprentissage individuel (par exemple : le temps de réponse à une question).

La deuxième classe détermine la valeur d'usage de la trace : la trace porte une valeur d'usage plutôt qualitative (par exemple : le résultat d'un test) ou plutôt quantitative (par exemple : la durée de connexion à une ressource).

La troisième classe identifie le cadre usuel d'exploitation de la trace : la trace apporte une information utilisée dans le cadre d'une étude de la connaissance de l'apprenant (exemple : la qualité d'une production) ou dans le cadre d'une étude de son comportement (par exemple : la fréquence des messages postés).

Cependant, il est à noter que les éléments d'une même classe ne sont pas mutuellement exclusifs, une même trace pouvant être utilisée dans différents contextes d'exploitation. Ainsi, les traces de réalisation d'actions peuvent être utilisées pour déterminer le comportement d'un apprenant lors d'une situation mais aussi pour définir son positionnement vis à vis de la connaissance mise en jeu lors de la situation d'apprentissage.

Dans le tableau 1 nous avons établie une vue synthétique des résultats utilisant les taxonomies de trace. Nous retrouvons les quatre types de traces définis dans le paragraphe précédent : les traces informatives, les traces liées à l'exploitation d'une ressource, les traces associées à l'activité d'apprentissage et les traces attachées à l'activité de communication ainsi que les trois classes d'exploitation : caractère de la situation (individuelle ou collective), valeur d'usage (qualitative ou quantitative) et cadre d'exploitation (étude du comportement ou étude de la connaissance).

Dans ce tableau un “+” signifie que la caractéristique à été trouvée quelque fois et “++” signifie souvent dans plus d’une dizaine d’articles.

TYPE	TRACES	Indiv	Coll	Qual	Quant	Comp	Conn	Exploitation possible
Informations	Informations personnelles (âge, genre, ...)	++	+					Identifier l'apprenant
	Informations techniques (IP, browser, SE, ...)	++	+					Identifier l'apprenant
Ressources	Nom (référence) de la ressource traité par un apprenant à un moment donné	++	+	+			+	Informer/Tenir au courant le tuteur
	Nombre d'accès	++			++	+	+	Dégager des informations sur l'apprenant (capacités, compétences, ...)
	Durée de consultation (/connexion)	++			++	++	+	Evaluer le niveau d'activité
	Origine de l'accès à la ressource	++		++			++	Dégager le cheminement conceptuel
	Historique du parcours des ressources	++		++			++	Dégager le cheminement conceptuel / profil de l'apprenant
Activité d'apprentissage	Qualité d'une production	++		++			++	Évaluation de l'assimilation de concepts/connaissances
	Temps de réponse à une question	++			++	+	+	Évaluation
	Tests antérieurs	++		++			++	Capitalisation pour permettre une adaptation
	Réalisations d'actions (ex : exécution, débogage)	++		+		+	+	Suivi du travail de l'apprenant / Respect du scénario

Activité de communication	Nombre des messages envoyés (mail)		++		++	+		Améliorer la régulation d'un groupe
	Nombre de messages reçus (mail)		++		++	+		Améliorer la régulation d'un groupe
	Nombre des messages postés (forum)		++		++	+		Évaluer l'interaction dans un groupe
	Nombre des messages lus (forum)		++		++	+		Évaluer l'interaction dans un groupe
	Fréquence des messages (sur une période donnée)		++		++	+		Améliorer la régulation d'un groupe / Évaluer la réactivité
	Destinataire des messages		++			+		Évaluer l'interaction dans un groupe
	Message de communication		++	+		+	+	Évaluer la qualité des interactions
	Demande d'aide en ligne	++	+	+	+		+	Évaluer l'adéquation d'une ressource / Fournir une aide
	Demande d'assistance au tuteur	++	+	+	+		+	Régulation de la situation

Légende :

Indiv : Individuelle ;
Coll : Collective ;
Qual : Qualitative :
Quant : Quantitative ;
Comp : Comportement ;
Conn : connaissance

Figure 5: Taxonomie des traces numériques

7. Conclusion :

Nous avons vu à travers ce chapitre, emploient la notion de trace, qui a un rôle et une utilisation particulière dans chaque approche. Toutefois nous qualifiant la trace d'apprentissage comme étant une donnée issue d'observation permettant la régulation, l'évaluation, l'analyse et la compréhension de l'activité d'apprentissage. De manière abstraite, nous définissons la trace comme une séquence temporelle d'observés. Le terme séquence temporelle montre l'existence d'une relation d'ordre organisant les données de la trace par rapport à un repère de temps et le terme observé désigne que les données de la trace sont issues d'une observation.

Néanmoins, pour déterminer le type de parcours d'un apprenant nous avons besoin de connaitre les pages visités, leur ordre de visite, leur contenu en terme de mot renvoyant au cours, ainsi il sera possible de déterminer a quel point les pages visités sont proche du cours et déduire quel type parcours l'apprenant a fait.

Chapitre II - Métadonnées et Ontologies

1. Introduction :

De nos jours. Il y a une explosion du volume des informations numériques. Cette surabondance d'informations nécessite de disposer de nouvelles méthodes, de nouveaux modèles capables d'extraire des informations d'un tas de connaissances mal structurées.

Pour réussir à contenir cette masse d'information émergente, il a fallu faire une sorte de recueil de es données les plus pertinentes, les cataloguer et les présenter sous forme de résumer a consulter pour éviter la lecture total d'un document qui ne correspond pas a une requête (recherche) ; cela a été l'apparition des métadonnées.

Les métadonnées informatique sont des informations sur des objets (image, texte, page web, son) qui décrivent aussi bien le **contenu** de ces objets que leur **gestion**, leur **structure**, leur **contexte** ou les **conditions d'accès**. Les métadonnées, utilisées abondamment par les professionnels de la documentation, peuvent déjà faciliter l'accès à certains documents archivés. Toutefois, le recours à des ontologies, permettant d'organiser, de **structurer les connaissances**, commence à s'imposer afin d'obtenir des informations « plus intelligentes » qu'on pourra utiliser.

De manière générale, l'utilisation de connaissances en informatique a pour but de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue, une coopération entre le système et les utilisateurs (système d'aide à la décision, système d'enseignement assisté par ordinateur, recherche d'information). Pour cela, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais également à la sémantique qui leur est associée, afin qu'une communication efficace soit possible. Les ontologies visent à représenter cette connaissance en étant à la fois interprétables par l'homme et par la machine.

Dans ce chapitre, le recours aux métadonnées est analysé dans le contexte du Web. Une place importante est donnée à la description des technologies basées sur XML qui jouent un rôle fondamental, aussi bien pour décrire les informations que pour les traiter. Ensuite nous relèverons les différentes définitions qui ont été attribuées à la notion d'ontologie, nous verrons aussi les différents éléments dont elle est composée et les besoins auxquels elle répond. Ensuite, nous aborderons les différents formalismes de représentations. Puis, nous décrivons les approches et les méthodologies de construction d'ontologie. Pour enfin finir par la présentation des langages de description et outils de construction d'ontologies.

2. Les métadonnées :

2.1. Qu'est-ce que le concept de métadonnées ?

Le concept de métadonnées a évolué avec l'utilisation de l'Internet. Elles étaient initialement limitées aux informations d'archivage, permettant de retrouver un ouvrage ou un document dans une bibliothèque ou un service d'archives. Les métadonnées peuvent aussi répondre à d'autres objectifs : savoir quel usage peut être fait d'un ouvrage ou découvrir ce que contient effectivement une ressource.

Avec l'informatisation de la gestion des établissements publics ou commerciaux, les métadonnées ne sont plus seulement utilisées pour retrouver un ouvrage ou un enregistrement sonore, mais aussi pour stocker d'autres informations, invisibles au lecteur : l'état du stock, le nombre d'emprunts ou le prix d'achat par exemple. En particulier, lorsqu'il s'agit de documents à diffusion limitée, des métadonnées peuvent être pertinentes pour la sécurité ou la fiabilité des informations.

2.1.1. Comment trouver une information ?

On peut considérer qu'il y a actuellement des millions de pages accessibles par leur adresse sur le Web. Cette adresse joue le rôle d'un numéro de référence dans une gigantesque bibliothèque. Mais, pour connaître les adresses des pages cherchées, il faut généralement interroger les sites qui proposent des moteurs de recherche, tels que Google, Lycos ou Yahoo. Ces sites utilisent un robot qui recherche sur tout l'Internet, regardant le contenu des documents à la place du lecteur.

La plupart des moteurs de recherche ignorent les métadonnées. Ils se contentent généralement d'examiner le texte contenu dans le titre, l'adresse, l'en-tête ou le début d'un document et d'y appliquer des méthodes statistiques (fréquence des mots, pondération selon l'emplacement dans le document, fréquence des accès, proximité, etc.) : Lycos, par exemple, ne retient qu'une centaine de caractères pour décrire une page Web. À l'inverse, Yahoo qui n'utilise pas de robot, mais fait appel à des indexeurs humains pour cataloguer les informations du Web en fonction de rubriques très générales, peut être considéré comme un annuaire.

Le manque de précision des documents sélectionnés par les moteurs de recherche, tant pour leur contenu que par la quantité des adresses proposées, incite à envisager d'autres solutions. Alors que les bibliothécaires n'ont pu se décider à utiliser un format commun pendant tant d'années, il ne paraît pas envisageable d'imposer pour l'Internet un ensemble commun de métadonnées. Toutefois, l'usage d'une même technologie peut contribuer à faciliter les échanges, malgré cette absence de standard.

2.1.2. Que contiennent les métadonnées ?

À la différence des informations de catalogage traditionnelles qui sont fournies généralement par des spécialistes à l'intention de lecteurs humains, les métadonnées sur les pages Web sont destinées à être traitées par des machines. Le format MARC, la Dublin Core. L'en-tête de la TEI .ou simplement les propriétés d'un fichier MS Office, sont des métadonnées conçues pour être comprises par l'ordinateur. Toutefois, il s'agit essentiellement de métadonnées descriptives du contenu.

Plus généralement, on peut considérer que les métadonnées peuvent fournir toutes sortes d'informations relatives à une ressource ou à son usage :

- des métadonnées descriptives (du contenu, de l'origine de l'information, bibliographique...) ;
- des métadonnées administratives (juridiques, commerciales…
- des métadonnées structurelles (relations entre composants d'une collection fractionnement...).

2.1.3. Qui détermine les métadonnées ?

À la différence des informations de catalogage qui sont fournies par des spécialistes, les métadonnées pour des documents sur l'Internet peuvent provenir de plusieurs sources comme, par exemple :

- de celui qui détient tous les droits sur le document (propriétaire) ;
- de celui qui détient le document (gestionnaire) ;
- de celui qui en fait commerce ;
- de celui qui l'évalue (pédagogue, critique, intermédiaire) ;
- de celui qui l'utilise (lecteur, abonné, client, acheteur).

L'utilisation d'une technologie qui offre la possibilité d'intégration de plusieurs sources revêt donc une importance primordiale dans le cas de sources multiples, personnelles ou publiques. De plus, il est essentiel dans un contexte comme l'Internet que l'origine des informations puisse être certifiée.

2.1.4. Sur quoi portent les métadonnées ?

Dans le domaine documentaire, les ressources électroniques sont rarement des objets isolés : on parle de fonds documentaires, de corpus, de collections ou d'œuvres qui partagent certaines caractéristiques. Dans une page Web, on peut également identifier des objets de plus petite taille, comme par exemple une image, une citation ou une annotation. Quelle que soit la granularité de la ressource, il faut pouvoir la décrire avec des métadonnées.

Le traitement de métadonnées définies à plusieurs niveaux peut donc devenir assez complexe. Le recours à des modèles de représentation de connaissance ou des ontologies peut en faciliter l'interprétation.

2.1.5. Où placer les métadonnées ?

À la différence des informations de catalogage qui sont stockées généralement indépendamment du document lui-même, les métadonnées décrivant des ressources de l'Internet peuvent faire l'objet d'une localisation plus variée :

- encapsulation des métadonnées dans la ressource, comme, par exemple, les métadonnées contenues dans l'en-tête d'un document codé avec la TEI ;
- association de métadonnées externes à la ressource, dans un document séparé (MARC) ;
- métadonnées indépendantes, reliées au document par une URI (HTML, XML) ;
- groupement de métadonnées dans une base de données de catalogage qui donne accès à un ensemble de ressources, sur le modèle des notices bibliographiques de l'archivage traditionnel ;
- encapsulation de la ressource dans les métadonnées la décrivant.

Ces différentes localisations peuvent être envisagées selon la technologie numérique mise en œuvre sur le Web.

2.2. Documents structurés

Dans le domaine documentaire, on assiste à la modification du support et de la diffusion des documents. Cette évolution touche aussi bien les documents administratifs, techniques que les documents commerciaux, les œuvres littéraires ou scientifiques, dans le domaine professionnel comme dans l'environnement personnel. L'informatisation des catalogues, puis la généralisation de la notion de document numérique et surtout l'accès à distance aux documents contribuent à leur dématérialisation. Mais la première évolution majeure apportée par l'informatisation des documents concerne la possibilité de structurer les documents selon leur contenu.

2.2.1. Syntaxe

Dans le monde documentaire, la structuration des ressources numériques a commencé avec SGML, bien avant l'apparition de XML. Elle s'est renforcée d'autant plus depuis que XML est apparu comme un standard pour structurer des documents.

2.2.2. XML

XML (eXtensible Markup Language) est un sous-ensemble de SGML, qui permet de définir toutes sortes de documents structurés. Plus simple à utiliser que le SGML, XML s'est rapidement substitué à SGML dans le domaine de la documentation structurée. XML offre une syntaxe souple pour structurer des documents, en fournissant un ensemble de règles de création de vocabulaires (noms de balises et d'attributs). Les schémas XML peuvent ensuite être utilisés pour autoriser la composition de vocabulaires XML.

De plus, XML peut être considéré comme la technologie de base qui permet de partager et de structurer à la fois les documents et les données sur le Web. Le codage des documents XML en Unicode conforte cette destination internationale. Associé aux autres recommandations du World Wide Web Consortium (W3C) concernant l'exploitation des documents, XML offre un environnement applicatif très riche.

La technologie XML présente de nombreux avantages, rappelés ici :

Les documents XML peuvent être décrits par des modèles de documents, appelé « schémas ». Un schéma décrit non seulement la terminologie (les noms des balises), mais aussi des contraintes d'utilisation (structure, type de contenu) ;

Le mécanisme des « espaces de noms » permet d'apporter des extensions à un schéma en déclarant un nouveau vocabulaire : de nouveaux noms d'éléments ou d'attributs ;

Le langage de transformation XSLT (eXtensible Stylesheet Language Transformations) permet de convertir efficacement des documents XML en documents XHTML, de les restructurer, de construire automatiquement des index, d'en extraire des informations, etc. La conversion peut être effectuée dynamiquement, lors de la consultation des documents, soit sur le site du serveur, soit sur le site du client. Mais peu de navigateurs offrent actuellement cette dernière possibilité ;

La désignation d'un fragment de document est possible en utilisant XPointer (XML Pointer Language). C'est le langage de base de description d'une identification de ressource. XPointer utilise le langage XPath pour sélectionner précisément des éléments dans la structure d'un document XML ;

Le langage XPath (XML Path Language) est utilisé par XPointer et par XSLT : en s'appuyant sur la structure logique du document, sur le type des éléments, sur les valeurs des attributs, sur les caractères contenus ou sur les positions relatives, XPath permet de se déplacer dans la structure d'un document XML. Un puissant mécanisme de comparaison permet de retrouver des éléments selon leur motif, c'est-à-dire non seulement en fonction de leur contenu, mais aussi de leur structure ;

le langage de requête XQuery permet d'extraire des informations de documents XML. La désignation dans le document XML utilise la syntaxe XPath. Le résultat de la requête est structuré en XML.

En utilisant les technologies XML mentionnées ci-avant, il est donc possible d'extraire automatiquement des métadonnées d'un document XML. Il est également aisé de transformer les documents XML ou des métadonnées dans d'autres formats.

En revanche, XML n'impose aucune contrainte sémantique sur la signification des documents. C'est en ajoutant progressivement des descriptions aux données et aux documents déjà existants sur le Web, que XML, RDF et OWL permettront au Web d'être une infrastructure globale pour partager des informations et les rechercher de manière plus efficace : ce que le W3C désigne sous le vocable de Semantic Web.

Modèles

De nombreuses descriptions de documents spécifiées en XML prévoient des métadonnées figurant dans l'en-tête du document structuré ou décrivant des images (XML, IPTC, MIX). Chaque domaine d'application propose son modèle : la DocBook, pour les documents techniques, la TEI pour les œuvres littéraires, l'EAD pour les archives, etc. Mais la plupart de ces modèles intègrent les concepts du DC et éventuellement ajoutent des métadonnées complémentaires.

Métadonnée*s*

Les concepts du Dublin Core ont été étendus afin de compléter certains éléments descriptifs : des qualificatifs permettent de préciser le sens d'un élément (raffinement) ; d'autres qualificatifs identifient des schémas d'encodage (définition de vocabulaire, notations, règles d'interprétation). Des extensions pourraient encore être envisagées afin d'apporter plus d'informations concernant l'usage et la gestion des documents, notamment.

Les concepts du DC peuvent être représentés selon différentes syntaxes : en HTML, en XML, dans l'en-tête de la TEI ou en RDF/XML.

De nombreux logiciels sont disponibles (DC *tools*) pour faciliter la création des métadonnées du DC : saisie dans des formulaires pour HTML, extraction, conversion de formats (HTML, TEI, USMARC, MARC 21, UNIMARC), production d'un fichier RDF en XML, etc.

3. Ontologies

De manière générale, l'utilisation de connaissances en informatique a pour but de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un

dialogue, une coopération entre le système et les utilisateurs (système d'aide à la décision, système d'enseignement assisté par ordinateur, recherche d'information). Pour cela, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais également à la sémantique qui leur est associée, afin qu'une communication efficace soit possible. Les ontologies visent à représenter cette connaissance en étant à la fois interprétables par l'homme et par la machine.

La tâche des ontologies est de définir les primitives et leur signification, celles-ci sont nécessaires pour la représentation des connaissances dans un contexte donné. Définir une ontologie est une tâche de modélisation menée à partir de l'expression linguistique des connaissances.

3.1. Origines des ontologies

Le terme « ontologie » a été utilisé pour la première fois par les philosophes GREC dans une discipline qui a plus 2300 ans, qui traite les différents genres d'entités dans le monde, et les relations entre ces gens. Dans cette discipline le mot « ontologie » se décompose en deux mots Ontos l'être, ce qui est, et de Logos qui signifie discours. En résume l'ontologie c'est l'étude de ce qui existe.

La notion d'ontologie a été abordée pour la première fois par John McCarthy dans le domaine de l'intelligence artificielle (IA). Il affirmait déjà en 1980 que les concepteurs des systèmes intelligents fondés sur la logique devraient d'abord énumérer tout ce qui existe [11]. Cette approche présentée par John McCarthy n'est pas la seule puisque par la suite plusieurs définitions ont été proposées par d'autres auteurs du domaine.

3.2. Définitions

La définition donnée aux ontologies a évoluée au court du temps au sein de la communauté de l'IA. La plus communément admise est celle donnée par T. Gruber qui la définie comme étant ***une spécification formelle est explicite d'une conceptualisation*** [12].

Cette définition fait suite à des travaux antérieurs qui décrivaient les ontologies comme *la définition des termes et relations de bases, comprenant aussi bien le vocabulaire d'un domaine que les règles qui permettent de les combiner afin d'étendre ce vocabulaire* [13]. Cette vision des ontologies a été reprise et étendue par R. Studer [14] et c'est cette définition que nous retenons : ***« les ontologies sont des spécifications formelles et explicites d'une conceptualisation partagé ».***

Cette définition a été étoffée dans [15] où une ontologie est définie comme étant

Un ensemble de définitions, de primitives, de représentation de connaissances spécifiques au contenu : classes, relations, fonctions et constantes d'objet.

La même notion est également développée dans [16] : **une ontologie est une théorie logique dont les modèles contraignent une certaine conceptualisation, sans la spécifier exactement**.

Pour lui la définition de Gruber fait appel à la signification implicite d'une conceptualisation, c'est pourquoi il la précise ; qu'il considère les ontologies comme des bases de connaissance particulières.

Pour [17] ***une ontologie est une description formelle d'entités et leurs propriétés, relations, contraintes, comportement***.

Elle est simplifiée dans [18] où une ontologie est définie comme un ***ensemble de définitions de concepts et leurs relations, à ne pas confondre avec un modèle qui est un ensemble d'instances de ces concepts***. [19] donne comme complément de définition: **une ontologie est un ensemble de spécifications de concepts compréhensible par une machine**.

Cette idée est renforcée car une ontologie fournit la structure de base, l'armature autour de laquelle une base de connaissances peut être construite.

En effet les logiciels ont besoin d'une représentation du monde aussi fidèle que possible, afin que les connaissances qu'ils avancent soient cohérentes, et que le mode de raisonnement qu'ils appliquent sur ces connaissances produit des résultats corrects.

On analysant toutes ces définition, on constate que des divergences règnent entre les auteurs et les chercheurs du domaine, mais une unanimité est établie auteur de deux principes : une ontologie est relative à un domaine, et est constituée de concepts et de relations les reliant les uns aux autres.

3.3. Composante d'une ontologie

En se basant sur les deux principes cités ci-dessus, on constate que les ontologies fournissent un vocabulaire commun d'un secteur et définissent – avec différents niveaux de formalité - la signification des termes et des relations entre elles. Les connaissances dans les ontologies sont véhiculées à l'aide de cinq éléments [12] : Concepts ; Relations ; Fonctions ; Axiomes ; Instances.

3.3.1. Les Concepts

Ils sont appelés aussi termes ou classes de l'ontologie. Un concept est un constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable.

L'ensemble des propriétés d'un concept constitue sa compréhension ou son intension et l'ensemble des êtres qu'il englobe, son extension.

Selon [13]]un concept se définit à trois niveaux :

- *Un concept est une signification* : Sa place dans un système de significations permet de le comprendre, de le distinguer et de le différencier par rapport à d'autres concepts.
- *Un concept est une construction :* Comprendre un concept revient à construire l'objet dont il est le concept.
- *Un concept est une prescription :* On le comprend en exécutant l'action qu'il entreprend. Selon [14], ces concepts peuvent être classifiés selon plusieurs dimensions :
- Niveau d'abstraction (concret ou abstrait) ;
- Atomicité (élémentaire ou composée) ;
- Niveau de réalité (réel ou fictif).

En résumé, un concept peut être tout ce qui peut être évoqué et, partant, peut consister en la description d'une tâche, d'une fonction, d'une action, d'une stratégie ou d'un processus de raisonnement, etc.

3.3.2. Les Relations

Représentent un type d'interaction, ou bien des associations existant entre les concepts d'un domaine. Elles se définissent formellement à partir d'un produit de n concepts : R : C1× C2× … ×Cn-1 --> Cn

sous-classe-de (Spécialisation, généralisation), partie-de (agrégation ou composition) associée-à, instance-de sont des exemples de relations binaires.

Voici quelques relations les plus courantes:

1) L'équivalence : une relation R est une relation d'équivalence si et seulement si : R est symétrique, réflexive et transitive. On écrit :

(R est une relation d'équivalence)⇔((R symétrique) ^ (R réflexive) ^ (R transitive))

2) la cardinalité : c'est le nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs. Exemple : une pièce a au moins une porte, un humain a entre zéro et deux jambes.

3) L'incompatibilité : Deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts. Exemple : les relations «être rouge » et «être vert» sont incompatibles ; *4) L'inverse* : Deux relations binaires sont inverses l'une de l'autre si, quand l'une lie deux instances I1 et I2, l'autre lie I2 et I1. Exemple : les relations « a pour père » et « a pour enfant » sont inverses l'une de l'autre ;

5) L'exclusivité : Deux relations sont exclusives si, quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.

Exemple : l'appartenance et le non appartenance sont exclusives. Et bien d'autres relations…

3.3.3. Fonctions :

Ce sont des cas particuliers de relations dans lesquelles le Néme élément de la relation est défini de manière unique à partir des n-1 premiers. Formellement, les fonctions sont définies ainsi : F: C1× C2× … ×Cn-1 --> Cn.

Constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine, traduites par l'ontologie. Ils ont pour objectif de représenter des concepts et des relations dans un langage logique permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique.

L'utilisation des axiomes sert à définir le sens des entités, mettre des restrictions sur la valeur des attributs, examiner la conformité des informations spécifiées ou en déduire de nouvelles.

4. Les ontologies : différents besoins :

Dans cette section, nous allons voir pourquoi a-t-on besoin des ontologies. Les ontologies sont utilisées dans plusieurs domaines, les plus répandus sont :

- Communication.
- Interopérabilité entre les systèmes.
- Ingénierie des systèmes.

La figure ci-dessous montre les domaines d'utilisation des ontologies :

Figure 6 : Domaines d'utilisation d'ontologies

4.1. Communication :

Il existe trois types de communication dans un projet : communication homme-homme, homme-système ou entre les différents modules du système. Ces trois types possèdent tous des caractéristiques particulières qui engendrent certains problèmes auxquels les ontologies peuvent apporter des solutions.

La communication entre humain pose surtout des problèmes quand les acteurs de cette communication ne sont pas du même domaine et ne parlent donc pas forcément le même langage.

Elle devient efficace s'ils ont des connaissances ou des points de vue partagés. Ces connaissances partagées peuvent être obtenues si le domaine est explicitement décrit sans confusion terminologique ou conceptuelle pour être compris de la même façon par tout le monde.

Une ontologie facilite la communication en fournissant une spécification explicite d'un domaine qui représente un modèle normatif. De plus, les ontologies permettent d'assurer la consistance et d'enlever l'ambigüité dans les descriptions es connaissances concernant un domaine spécifique. Finalement, les ontologies peuvent intégrer différentes perspectives des utilisateurs. Quand les utilisateurs (qui ont différentes perspectives du domaine) partagent une ontologie, ils ont une perspective standard.

4.2. Interopérabilité :

L'interopérabilité implique la possibilité de pouvoir demander et recevoir des services entre des systèmes interopérables.

Deux systèmes sont considérés interopérables s'ils vérifient les deux conditions suivantes:

- Ils opèrent comme une unité afin de réaliser une tâche commune.
- Ils peuvent s'échanger des messages et des requêtes.

Les ontologies permettent de faciliter l'interopérabilité en intégrant les connaissances concernant différents domaines dont l'objectif est de décrire un domaine unifié accomplir une tâche commune. Elles permettent aussi d'intégrer les différents vocabulaires concernant certains domaines. Pour ce faire, les ontologies de ces domaines doivent être intégrées afin de partager un même vocabulaire.

4.3. Construction des ontologies :

4.3.1. Les méthodologies de construction d'ontologies :

Le processus de construction d'une ontologie est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, voire les futurs utilisateurs de l'ontologie [19]. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent.

La figure ci-dessous représente le processus de construction d'ontologie :

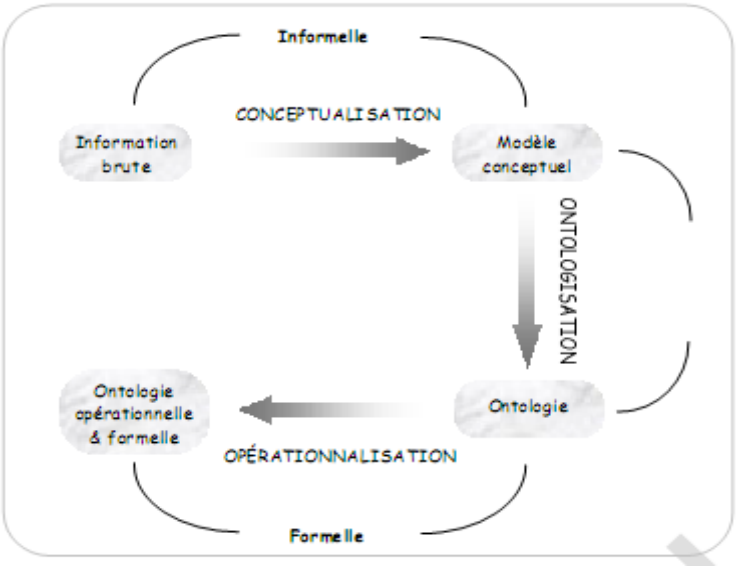


Figure 7 : processus de construction d'ontologie

L'évaluation des besoins :

Le but visé par la construction d'une ontologie se décline en 3 aspects [19]:

L'objectif opérationnel **:** il est indispensable de bien préciser l'objectif opérationnel de l'ontologie, à travers des scénarios d'usage.

Le domaine de connaissance **:** il doit être délimité aussi précisément que possible.

Les utilisateurs **:** ils doivent être identifiés, ce qui permet de choisir, en accord avec l'objectif opérationnel, le degré de formalisme de l'ontologie, et sa granularité.

Une fois le but défini, le processus de construction de l'ontologie peut démarrer.

Conceptualisation

l'objectif est d'organiser et de structurer la connaissance acquise durant l'étape de spécification en utilisant des représentations externes qui sont indépendantes des paradigmes de représentation de connaissances et des langages d'implémentation dans lesquels l'ontologie va être formalisée et implémentée. L'idée est de combler graduellement le canal entre les moyens d'expressions des intéressés et les langages d'implantation des ontologies. Les représentations intermédiaires utilisées sont :

Les taxonomies de concepts, les diagrammes des relations binaires, le glossaire des termes, le dictionnaire des concepts, le tableau des relations binaires, spécifier des contraintes sur les attributs dans une table d'attributs, spécifier des axiomes sur les concepts dans une table d'axiomes logiques, décrire les instances des concepts dans une table d'instances.

Formalisation :

Consiste en une formalisation partielle sans perte d'information, du modèle conceptuel obtenu dans l'étape précédente. Ce qui permet de faciliter sa représentation ultérieure dans un langage formel et opérationnel.

Elle effectue une transcription des connaissances dans un certain formalisme de connaissance, ce formalisme devant être aussi générique que possible, mais sémantiquement clair.

Le modèle obtenu est souvent qualifié de semi-formel (car certaines connaissances ne peuvent pas être totalement formalisées).

5. Langages de représentation :

5.1. Langages de représentation d'ontologie :

Plusieurs langages de spécification d'ontologies (ou langages d'ontologies) ont été développés pendant les dernières années. Certains d'entre eux sont basés sur la syntaxe de XML, tels que XOL (Ontology Exchange Language), SHOE (Simple HTML Ontology Extension - qui a été précédemment basé sur le HTML), OML (Ontology Markup Language), RDF (Resource Description Framework), RDF Schéma.

Les deux derniers sont des langages créés par des groupes de travail du World Wide Web Consortium (W3C).

La figure présente des langages de spécification d'ontologie, qui ont été récemment développés. La figure ci-dessous représente les rapports principaux entre tous ces langages sous la forme d'une pyramide des langages du Web sémantique.

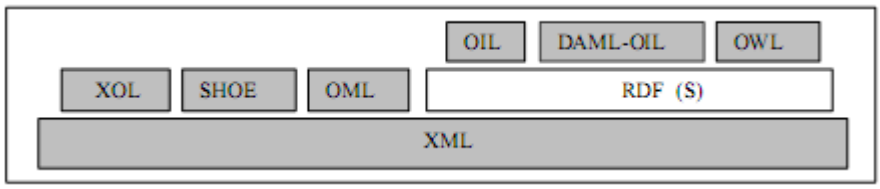


Figure 8 : La pyramide des langages basés Web.

XML

Est un langage permettant de générer des balises pour la structuration de données et de documents. Il permet la représentation et l'échange de documents semi-structurés. XML-schéma permet de définir la structure, les contraintes, et la sémantique de documents XML. Ce langage n'est pas vu comme un langage d'ontologies car il a été créé pour vérifier la structure de documents XML. Les primitives qu'il met en place sont plutôt orientées application que concept. En effet, la sémantique définie dans le document est interprétable dans le contexte de l'opération faite sur le document mais ne permet pas d'établir des inférences en dehors de ce contexte. XML et XML-schéma sont considérés comme des langages définissant le format de « message » alors qu'un langage d'ontologies a pour but de « représenter » la connaissance.

RDF :

RDF est un langage d'assertion et d'annotations. Les assertions affirment l'existence de relations entre les objets. Elles sont donc adaptées à l'expression des annotations que l'on veut associer aux ressources du Web. RDF est un langage formel qui permet d'affirmer des relations entre des « ressources ». Le modèle RDF définit trois types d'objets:

• *Ressources* : les ressources sont tous les objets décrits par RDF. Généralement, ces ressources peuvent être aussi bien des pages Web que tout objet ou personne du monde réel. Les ressources sont alors identifiées par leur URI (Uniform Resource Identifier) ;

• *Propriétés* : une propriété est un attribut, un aspect, une caractéristique qui s'applique à une ressource. Il peut également s'agir d'une mise en relation avec une autre ressource ;

• *Valeurs* : les valeurs en question sont les valeurs particulières que prennent les propriétés. Ces trois types d'objets peuvent être mis en relation par des assertions, c'est à dire des triplets (ressource, propriété, valeur), ou encore (sujet, prédicat, objet).

OWL :

OWL signifie Web Ontology Language, défini par le W3C dans, le langage OWL est basé sur la recherche effectuée dans le domaine de la logique de description. OWL permet de décrire des ontologies, c'est-à-dire qu'il permet de définir des terminologies pour décrire des domaines concrets. Une terminologie se constitue de concepts et de propriétés (aussi appelés rôles en logiques de description). Un domaine se compose d'instance de concepts.

SKOS [W3C]

SKOS ou Simple Knowledge Organization System (Système simple d'organisation des connaissances) est une famille de langages formels permettant une représentation standard des thésaurus, classifications ou tout autre type de vocabulaire contrôlé et structuré. Construit sur la base du modèle de données standard RDF, son principal objectif est de permettre la publication facile de vocabulaires structurés pour leur utilisation dans le cadre du Web sémantique. SKOS est, depuis le 18 août 2009, une recommandation du W3C.

Le développement de SKOS a impliqué des acteurs à la fois de la communauté RDF et des experts en Science de l'information. SKOS cherche à être au maximum compatible avec les standards tels ceux des thésaurus, monolingue ou multilingue.

Les représentations conceptuelles réalisées à l'aide de SKOS peuvent satisfaire des besoins de traitement restreints à un organisme, mais aussi, dans la perspective du Web sémantique, contribuer à la constitution d'une structure de concepts mis en commun et partagés à l'échelle du Web sous forme de ressource exploitable par les outils RDF ou autres.

6. Conclusion :

Dans ce chapitre nous avons abordé les notions de métadonnées, ensuite nous avons abordé la notion d'ontologie, les modèles d'organisation de la connaissance qui sont proposés par les ontologies, l'apport sémantique des ontologies est dans le contexte qu'elles expriment. Néanmoins, dans la mesure où elles sont utilisées directement ou indirectement par des êtres humains elles ont pour but de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue, une coopération entre le système et les utilisateurs (système d'aide à la décision, système d'enseignement assisté par ordinateur, recherche d'information).

Ainsi les ontologies, permettant d'organiser, de structurer les connaissances, pour obtenir des informations « plus intelligentes » qu'on pourra utiliser.

Pour résumer, dans ce chapitre nous avons parlé des outils «sémantique » que l'on a utilisé tout au long de ce travail, notamment les ontologies qui seront utiles lors de l'élaboration d'un cours modèle, d'XML pour formaliser et structurer les données a fin d'en faciliter l'exploitation.

Chapitre III – Démarche de travail

1. Introduction

Avec l'évolution des systèmes d'apprentissage et avec les avancées technologiques dans le domaine informatique, de nouvelles attentes ont émergé.et parmi ces attentes, nous trouvons la personnalisation et l'adaptation de l'apprentissage.

Notre démarche a pour but finale de fournir les résultats probant pour le tuteur a fin de prendre une décision concernant le contenu du cours, sa structuration et de détecter les apprenant en difficulté. Pour se faire nous avons opté pour la démarche suivante qui se base sur les traces d'apprentissage pour définir le parcours d'un apprenant durant sa session d'apprentissage et aussi sur l'ontologie du domaine du cours étudié ainsi que les mesures de similarités sémantiques existantes entre les document consulté en cours et en dehors du cours.

2. Structure du parcours d'un apprenant

Il existe dans la littérature trois approches de collecte de traces : des approches orientées clients basées sur des logiciels spécifiques de collecte, d'autres orientées serveur et celles non médiatisées par ordinateur (collecte manuelle ou audiovisuelle) utilisé uniquement pour des expérimentations.

Dans notre situation, pour avoir une conception complète de toute l'activité de l'apprenant, nous avons opté pour l'approche de collecte orientée clients. Il s'agit de tracer toutes les interactions de l'apprenant, à travers un outil d'observation installé sur sa machine, et qui sera activé par sa propre volonté (remédier au problème d'éthique). En conséquence, les traces générées seront structurées et sauvegardées dans un fichier XML, comportant les champs suivants :

Information sur l'apprenant : nom et identifiant de l'apprenant dans l'environnement d'apprentissage.

Id de session : identifiant de la session d'apprentissage,

Les pages consultées : l'ordre de la page, l'URL, le titre de la page et la date consultation, durée de consultation.

Ce modèle général de trace provenant d'une interaction peut être utilisé pour de nombreux cas d'étude.

3. Milieu d'apprentissage

La navigation d'un apprenant dans un système d'apprentissage peut être définie comme étant une activité au cours de laquelle un apprenant va passer de pages en pages par des clics sur des liens hypertextes. Les possibilités de navigation sont donc proposées explicitement par les concepteurs de plateforme d'apprentissage, alors que les apprenants peuvent éprouver le besoin de rechercher ou de solliciter des informations supplémentaires ou complémentaire dues à un besoin personnel et concernant un ou plusieurs mots ou concepts d'un contenu pédagogique auxquels il a été confronté dans son environnement d'apprentissage.

Par conséquent, nous pouvons avancer qu'un apprenant au cours de son apprentissage interagit avec tout le milieu qui l'entoure pour acquérir et enrichir ses connaissances dans le domaine étudié.

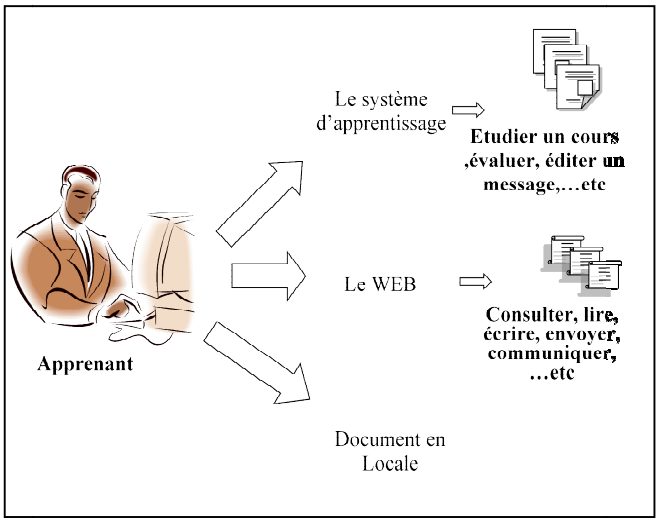


Figure9 :Interactions de l'apprenant durant une session d'apprentissage.

En revanche, si on prend en considération l'environnement numérique comme étant une partie du milieu d'apprentissage, pendant une session, l'apprenant pourra consulter plusieurs rubriques propres au système d'apprentissage comme : la rubrique Communication, Travaux et Exercices, Evaluation et essentiellement le cours, toutefois il peut consulter des documents externes au système d'apprentissage qu'il soit sur le Web ou en locale (disque dur CD, etc.), la figure 9 résume bien ses différent interactions.

L'objectif de notre mémoire est de mettre en évidence la similarité existante entre les concepts vus en cours avec ceux visités en dehors du cours (environnement extérieur), ceci dans le but de pouvoir détecter les manques qui poussent l'apprenant à visiter des documents externes ou encore à communiquer avec ses paires ou avec le tuteur sur des concepts vus en cours.

Le milieu externe que nous prenant en considération dans cette étude concerne le Web, cette restriction n'est pas arbitraire. L'analyse des adresses (URL) seules ne permet pas d'obtenir une information suffisamment fine, il est donc indispensable d'analyser le contenu de ces ressources. Par ailleurs, le contenu des ressources consultées sur le web est naturellement accessible, contrairement aux documents locaux (se trouvant sur la machine ou tout objet amovible de l'apprenant) qui sont difficiles d'accès, en plus la majorité de cette documentation a été probablement téléchargé depuis le Web. Ceci justifie alors le choix du Web.

Par conséquent, reconnaitre les concepts du cours qui posent problème va pousser et l'auteur à réviser son cours pour une bonne restructuration ou enrichissement et le tuteur à reconnaitre les apprenants en difficultés pour pouvoir les aider au bon moment (formule plausible, un tuteur pour un apprenant en difficulté).

Pour ce faire nous avons opté pour une méthode qui nous permet à la fois de détecter les concepts du domaine qui posent problème et de détecter les apprenants qui sollicitent ces concepts depuis l'extérieur (détecter les apprenants en difficultés), ceci grâce à l'analyse du parcours d'apprentissage à travers le calcul de similarité de visite. Sachant que les concepts du domaine sont bien modélisés à travers une ontologie du domaine propre au cours enseigné. Dans ce qui suit nous dévoilons l'utilité d'intégrer une ontologie dans les systèmes d'apprentissage et spécialement dans notre cas d'étude où nous décrierons le modèle d'ontologie du domaine utilisé.

4. Développement d'une ontologie du domaine

Chaque système d'apprentissage requière un modèle de domaine, un modèle intégrant des connaissances sur le domaine à enseigner. Le modèle de domaine apporte une aide dans la démarche pédagogique puisqu'il définit un ordre dans l'apprentissage des notions, conduisant à des parcours bien précis à travers les ressources pédagogiques.

Pour arriver à faire une analyse concrète il nous a fallu élaborer une ontologie d'un cours, et notre chois c'est porté sur le cours d'algorithmique. Pour qu'au final, lors de la recherche des termes dans le corpus constitué, on puisse les comparer a l'ontologie du domaine, on parle ici de projection.

En effet, les termes contenus dans les pages visités seront comparé à ceux contenue dans le cours, page après page, et pour chaque page, on définira les concepts référencés a partir des termes identifiés dans la page.

Nous devons apporter une précision quand a l'élaboration de l'ontologie du domaine, car elle doit être hiérarchisé au sein d'une structure et lié par des relations syntaxiques (synonymes) ou sémantiques.

Nous avons opté pour le format de réorientation SKOS, assez facile à exploiter

Ci-dessous nous montrons une capture d'écran de notre fichier « ontologie de cours », nous nous sommes arrêtés au nombre quatre concepts pour alléger l'écriture de l'ontologie et pour faciliter la gestion du code et des informations traitées ;

- Les boucles,
- les structures de données,
- les procédures,
- les conditions.

Chacun de ces quatre concepts regroupe un certain nombre de termes, organisé en terme général « prefered_label » et en termes alternatifs « alter_label », et cette modélisation accepte aussi différentes langues, ainsi, nous verrons dans notre ontologie la structure suivante ;

```
<Concept ord="1" name="structure de donnees">
        <preferlab lang="fr">structure de donnees</preferlab>
        <preferlab lang="en">data structure</preferlab>
        <altlab lang="fr">vecteur</altlab>
        <altlab lang="en">file</altlab>
</Concept>
```

```
<Concepts name="algorithmique">

    <Concept ord="1" name="structure de donnees">
        <preferlab lang="fr">structure de donnees</preferlab>
        <preferlab lang="en">data structure</preferlab>
        <altlab lang="fr">vecteur</altlab>
            <altlab lang="en">file</altlab>
        <altlab lang="fr">liste chainée</altlab>
        <altlab lang="fr">matrice</altlab>
        <altlab lang="fr">pile</altlab>
        <altlab lang="fr">tableau</altlab>
     </Concept>


    <Concept ord="2" name="les procedures">
        <preferlab lang="en">function</preferlab>
        <preferlab lang="fr">fonction</preferlab>
        <preferlab lang="fr">procedure</preferlab>
        <altlab lang="fr">methode</altlab>
            <altlab lang="fr">paramètres</altlab>
        <altlab lang="fr">Fonctions personnalisées</altlab>
        <altlab lang="fr">Variables publiques</altlab>
        <altlab lang="fr">Variables privées</altlab>

    </Concept>

    <Concept ord="3" name="les conditions">
        <preferlab lang="fr">conditions</preferlab>
        <altlab lang="fr">Structures conditionnelles</altlab>
            <altlab lang="en">La condition if... else</altlab>
```

Figure 10 : capture d'écran de l'ontologie retenu.

5. Analyse des données d'usage du web

Le Web offre des ressources pédagogiques abondantes, de plus c'est les ressources qui sont largement consultées par les apprenants. Ces ressources participent d'une manière ou d'une autre à l'échec ou à la réussite dans l'apprentissage. Donc nous allons nous reposer uniquement sur ces ressources pour effectuer notre analyse quant au contexte d'interaction hors environnement d'apprentissage.

Nous allons essayer alors à travers cette étude de reconnaitre le type de parcours entrepris par l'apprenant et ceci à travers la reconstruction de son parcours à base de trace d'apprentissage laissée pendant sa session formation.

Par conséquent, grâce au fichier trace, nous pouvons restituer le parcours de l'apprenant, le résultat sera sous forme d'une séquence de pages (documents) visitées pendant une ou plusieurs session d'apprentissage.

Nous allons essayer de calculer la distance de similarité entre deux pages qui se suivent dans la navigation. Nous allons refaire la procédure pour chaques deux pages successives dans le parcours, au finale nous allons calculer la moyenne de cette distance. Ceci va nous permettre de reconnaitre le type de parcours de l'apprenant.

Pour ce faire, on calcule la relation sémantique entre les concepts des pages parcourus par l'apprenant, sachant que ces concepts concernent les concepts ontologiques qui constituent le cours. L'algorithme se base sur l'idée que l'ontologie définit le contexte des concepts à étudier.

Mais avant tous, nous devons d'abord reconnaitre les informations englobées sur les pages visitées, car la similarité sémantique réside sur le contenu informationnel de ces pages. Il est important donc de récupérer les métadonnées des pages consultées pour pouvoir calculer leurs similarités.

5.1. L'approche suivie :

5.1.1. Génération du corpus :

Pour effectuer l'indexation, nous devons disposer du contenu textuel des pages Web. Ici encore, un autre problème se pose : la reconstitution des pages Web consultées lors d'une session de navigation. Notre solution à l'égard de ce problème est d'intégrer un module d'aspiration des pages, à partir des URL.

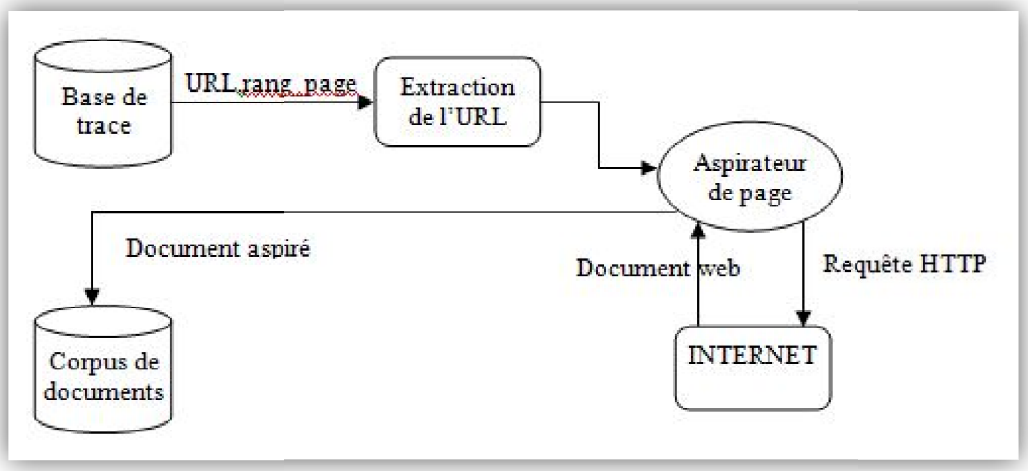


Figure 11: Constitution de corpus de document

L'aspiration des pages visitées par les apprenants à partir des données du trafic permet de reconnaitre le contenu du parcours. La base de trace contient des fichiers XML contenant l'URL des pages web et grâce au module extraction de l'URL, nous récupérerons ces URL des pages et identifiants pour qu'ils soient aspirés du web par l'aspirateur de page et sera sauvegardée dans un répertoire pour pouvoir constituer un corpus de documents visités sur le web.

5.1.2. Le processus d'indexation

Afin d'évaluer la similarité entre les pages Web consultées et l'ontologie du domaine du cours, puis des pages entre elles qui est l'objectif de cette étude, nous modelons chaque page par ces mots clés. Il est à noter que le cours est scénarisé par une ontologie du domaine, en revanche les pages consultées par l'apprenant en dehors de l'environnement d'apprentissage ne sont pas forcément indexées.

Les pages visitées sont accompagnées de métadonnées par conséquent les mot extrait sont contenus généralement dans la balise « meta name = keyword » , exemple ;

<meta name="keywords" content="file, conditions, tutoriels, programmation" />

Notations

Un corpus C est un ensemble de documents $\{d_1, d_2, \ldots d_N\}$.

M est le nombre de concept de l'ontologie du domaine

N est le nombre de documents du corpus.

Pour chaque document $\boldsymbol{d_i}$ on lui associe un vecteur booléen de concept, notée $Vq=<c_1,c_2, \ldots c_m>$. Si $c_m \neq 0$, cela indique la présence du concept recherché dans le document d_i.

5.1.3. Mise en œuvre de l'indexation conceptuelle

Par conséquent, une fois que nous disposons du contenu intégral de chaque page ou document du fichier de trace (corpus du document), nous procédons à leur indexation grâce au module d'indexation ainsi illustré sur figure 12.

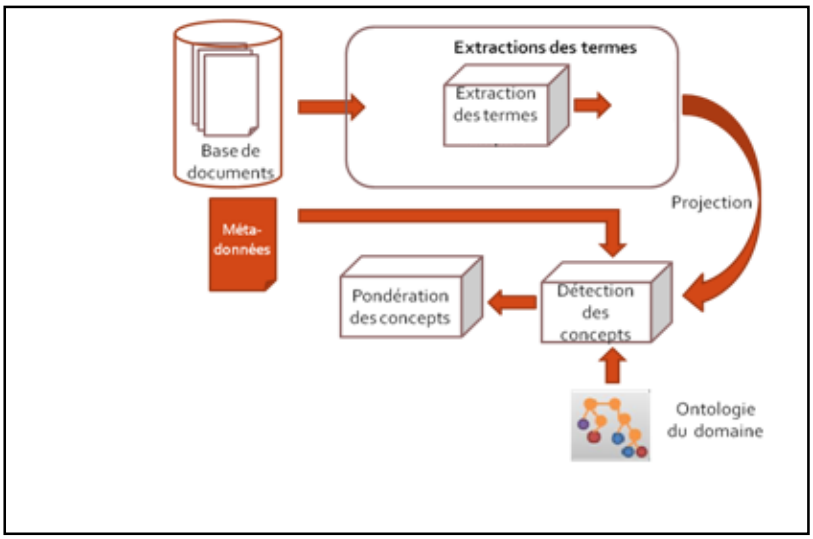


Figure 12 : indexation conceptuelle

Pour réaliser une indexation conceptuelle, nous avons besoin de ressources externes qui possèdent au moins une organisation par concepts et une structure. Nous pouvons donc utiliser dans notre cas d'étude une ontologie de domaine qui est propre au cours enseigné.

Pour une indexation conceptuelle, il nous faut alors :

- Une ontologie du domaine
- Un dispositif qui associe un terme à un concept (projection) ;
- Un outil pour l'extraction des mots clés sélection des termes à partir des pages web.

À partir d'un ensemble de documents Web, on procède à l'extraction des mots clés et les convertit en concepts en les projetant sur l'ontologie du domaine du cours et cela dans le but que les pages soit représentées par des vecteurs (concept, a valeurs booléennes).

Après indexation du fichier trace, nous procédons au calcul de similarité sémantique entre les pages consultées identifiées par leurs URL et concepts dans le nouveau fichier trace indexé.

5.1.4. Similarité sémantique entre deux (pages) documents successives :

Chaque page (URL visité) est représenté par un vecteur booléen de concepts (indexe). Rappelons que la représentation vectorielle booléenne donne comme information de la présence ou l'absence d'un mot du lexique dans le document, et non sa fréquence À cet effet nous procédons au calcul de la distance sémantique entre les deux vecteurs booléens correspondant à deux pages qui se suivent dans la navigation de l'apprenant. Ceci va nous aider à identifier le type de parcours de l'apprenant.

Imput : Fichier trace XML indexée
Output : Une mesure réelle entre 0 et 1

1. Pour chaque deux pages qui se suivent Pi et Pi+1, nous calculons la distance de similarité entre leurs vecteurs de concepts correspondants. A cet effet nous allons utiliser la mesure de Jaccard qui se fond sur la présence-absence des mots, ce qui justifie notre choix. La mesure de similarité de Jaccard [Rosset & al,08] est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs :

$$\text{Sim(Pi, Pi+1)} = \frac{Nc}{N1+N2-Nc}$$

Nc = le nombre de concept en commun entre les deux pages.
N1 = taille du lexique du vecteur de concept de la page Pi (i.e. nombre de concept différents dans Pi).
N2 = taille du lexique du vecteur de concept de la page Pi+1.

2. Refaire la procédure pour chaque deux page qui se suivent dans le temps.

3. En final, nous calculons la moyenne de cette distance

$$\frac{\sum_{1}^{n-1} sim(Pi, Pi+1)}{n-1}$$

Avec n est le nombre de page (URL) contenu dans le fichier trace.

5.1.5. Types de parcours :

Le taux de similarité calculé ajouté, nous permet de reconnaitre le type de parcours. Pour identifier ces types de parcours, nous nous appuyons ici sur la classification proposée par [Canter & al, 85] qui distingue quatre grandes catégories de navigation :

- Tant que la distance est bonne (proche de 1) le parcours est structuré (lecture approfondie) ce qui interprète que l'apprenant se forme de façon précise sur le domaine enseigné, il consulte des documents qui se rapprochent sémantiquement.
- Déstructuré (papillonnage) dans le cas contraire, donc l'apprenant consulte des pages qui sont soit dissimilaire ou d'une distance grande.
- Si le taux d'activité de type recherche sur un ou plusieurs concepts est élevé nous résumons un parcours de recherche, et la difficulté réside sur les concepts rédigés comme requête à travers les différents moteurs de recherche utilisée pendant sa recherche.
- Si la durée moyenne de consultation est très faible (inférieure au temps de référence) nous déduisons un parcours de survol.

6. Conclusion :

Durant ce chapitre, nous avons vu le besoin d'observer les comportements de navigation d'un apprenant afin de détecter le type de parcours qu'il entreprend pendant son apprentissage. Ce parcours diffère d'un apprenant à un autre selon les besoins et les difficultés rencontrer pendant son apprentissage. La détection de type de parcours pourra nous aider à reconnaitre les apprenants en difficultés, car un parcours déstructuré est un parcours qui nécessite une attention de la part de ceux qui participent au processus d'apprentissage.

Chapitre IV - Conception & Réalisation

1. Introduction :

Dans se dernier chapitre, nous allons parler de l'étape qui consiste à concrétiser nos recherches et notre solution informatique proposée pour le suivi des apprenants dans un système d'apprentissage a distance, afin de détecter les besoins et les manques des apprenant. Nous allons présenter l'application ; son intérêt, son but, et ses diverses applications possibles.

Nous commencerons par présenter les outils utilisés lors de la conception et de la réalisation de notre solution puis nous passerons à des exemples concrets qui montreront le fonctionnement des principaux modules de notre application. En effet nous sommes passés par un processus de conception, utilisant le langage de modélisation UML pour représenter différentes entités, suivi d'une réalisation qui est concrétisé par le langage de programmation Java et XML pour la structuration des documents et des données.

Nous terminerons par une démonstration qui mettra en évidence notre contribution dans le domaine de la formation à distance, les points fort ainsi que les avantages lié a l'exploration des données d'usages du web par des apprenants, enfin nous ferons le point sur les acquis, les avancés théorique et pratique, et les perspectives de notre travail.

2. Présentation d'UML :

2.1. Définition :

UML *(Unified Modeling Language)* se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

2.2. Modélisation UML

La modélisation consiste à créer une représentation simplifiée d'un problème: **le modèle**. Grâce au modèle il est possible de représenter simplement un problème, un concept et le simuler. La modélisation comporte deux composantes :

- **L'analyse**, c'est-à-dire l'étude du problème
- **La conception**, soit la mise au point d'une solution au problème
- **Le modèle** constitue ainsi une représentation possible du système pour un point de vue donné.

Pour favoriser la réussite d'un projet, les auteurs d'UML conseillent une démarche qui doit être :

Une démarche itérative et incrémentale : Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maîtriser l'inconnu.

Guidée par les besoins de l'utilisateur : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leurs besoins. Chaque étape sera affinée et validé en fonction des besoins des utilisateurs.

Centré sur l'architecture logicielle : C'est la clé de voûte du succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

3. Diagramme de cas d'utilisation

Un cas d'utilisation : Un cas d'utilisation décrit l'interaction et les dialogues entre l'acteur et le système. Les cas d'utilisations sont une technique puissante pour consigner et traduire le Comportement détaillé du système.

La relation include : Une relation inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B, signifie qu'une instance de A contient le comportement décrit dans B, le cas d'utilisation A ne peut pas être utilisé seul.

Relation extend : Une relation d'extension d'un cas d'utilisation A par rapport à un cas d'utilisation B, signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

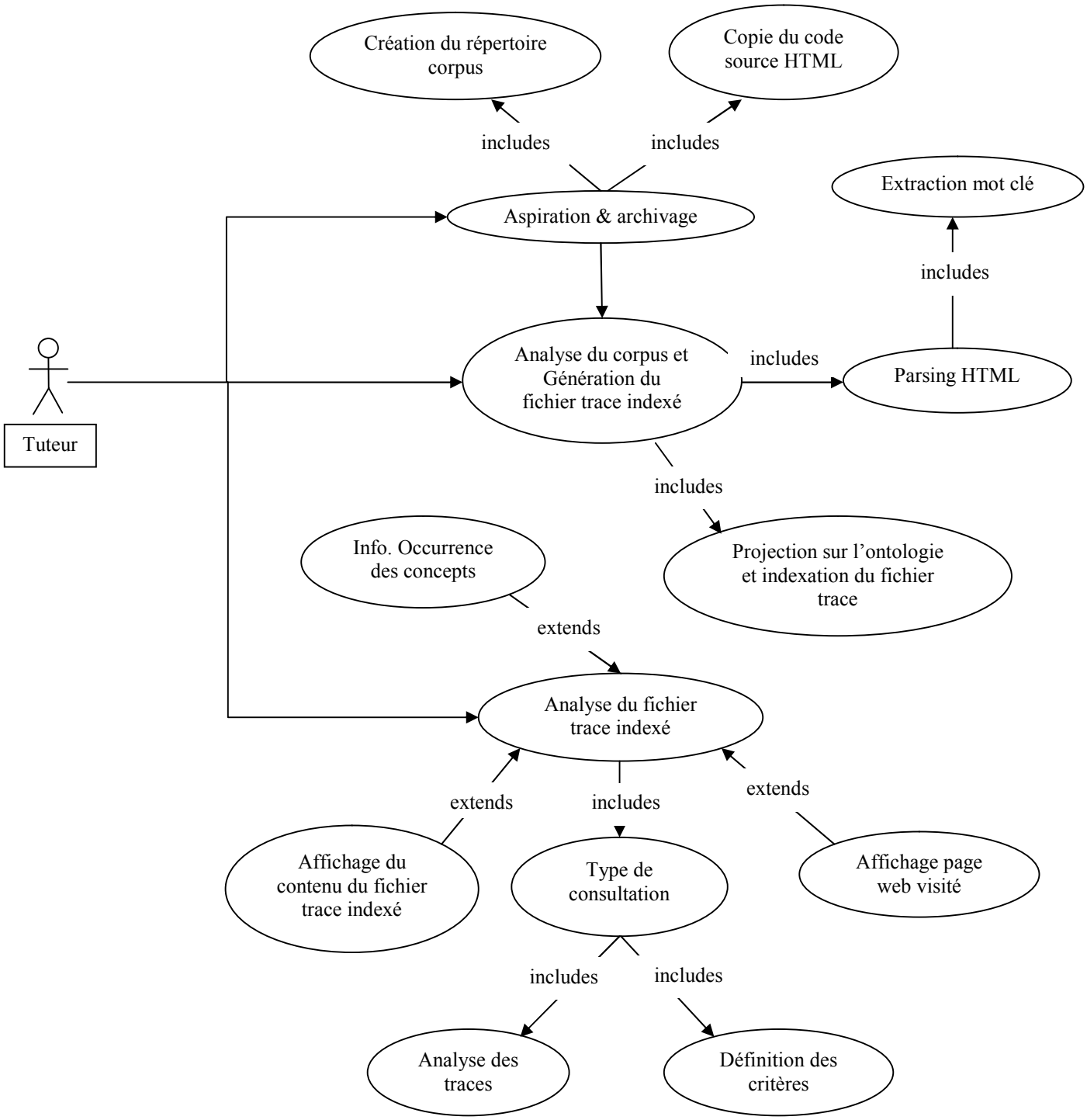


Figure 13 : diagramme de cas d'utilisation

Dans cette représentation de notre application, nous distinguons trois premières interfaces qui représentent les trois étapes de la solution ; l'aspiration du corpus, l'indexation du fichier trace, et l'analyse des résultats.

- L'aspiration consiste en la lecture du ficher trace, pour en extraire les URL, copier le code source de la page puis créer le répertoire corpus qui contiendra les pages dans l'ordre de visite.
- L'indexation du fichier trace consiste en l'extraction des mots clés des page web (contenu dans la balise méta) et les confronter a l'ontologie du cour qui produira un vecteur de booléens qui représentera l'existence ou non de concepts dans les pages visité.
- En fin viens l'analyse, et celle-ci nous donne un aperçu du contenu des pages visités, les infos sur la session et les concepts les plus récurrents. Elle propose aussi un module de régulation des critères pour l'évaluation relative des données.

4. Diagramme de séquence

4.1. Chargement d'un fichier trace et création d'un corpus

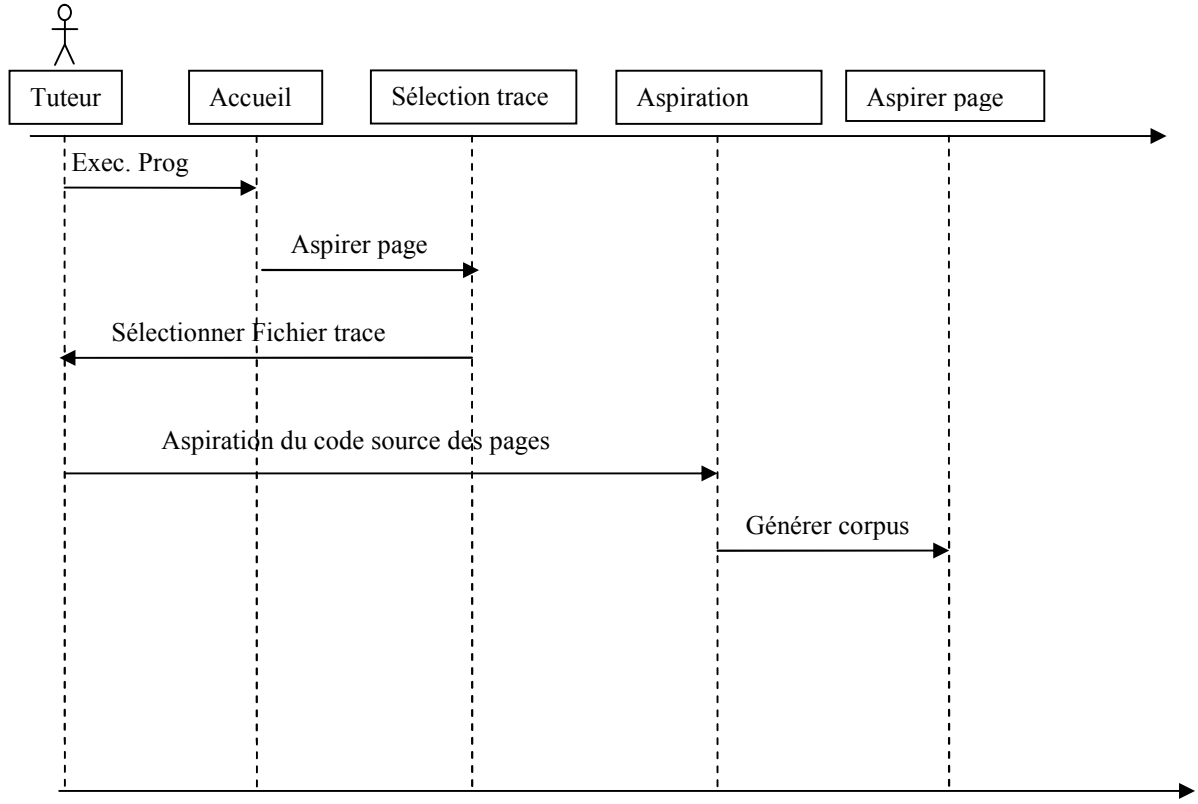


Figure 15: Chargement d'un fichier

(aspiration)

Dans ce cas, le tuteur exécute le programme, et la première interface qu'il rencontre est l'accueil, qui lui proposera trois bouton, ici le tuteur va faire en sorte de recueillir l'ensemble des pages visitées et cela en les aspirants. Pour ce faire, il devra charger un fichier de trace au format XML contenant les URL visité et ainsi récupérer leur code source chronologiquement par rapport a l'ordre de visite par l'apprenant.

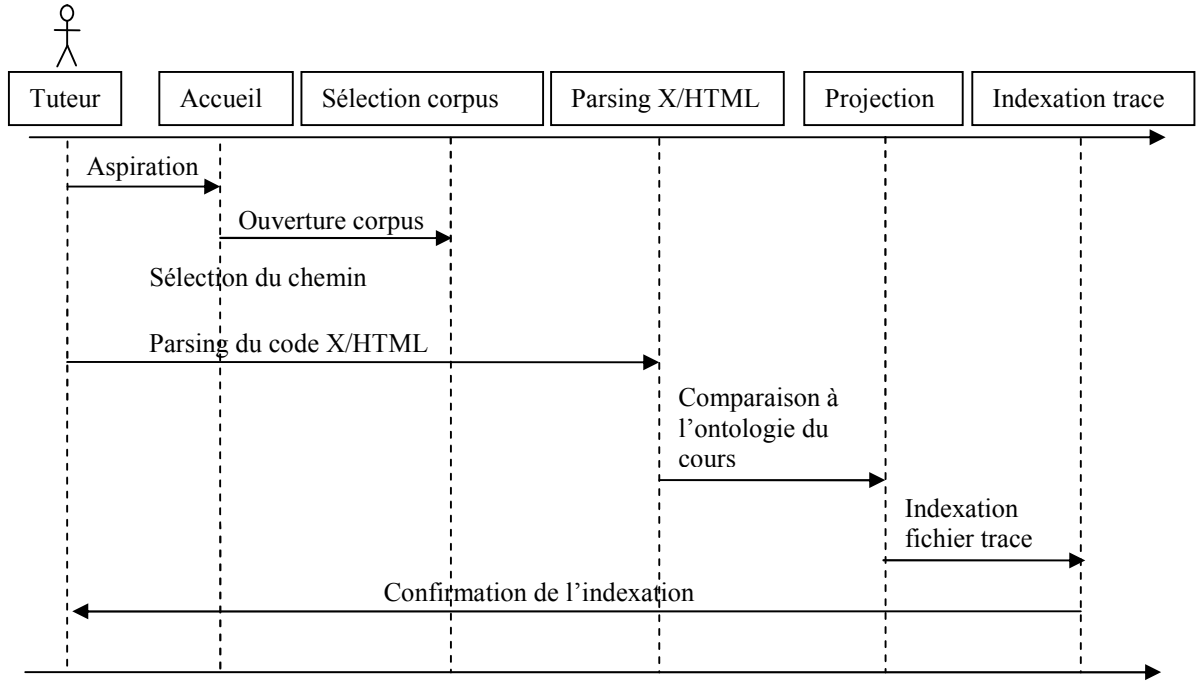


Figure 16 : Parsing du corpus et indexation du fichier trace

4.2. Parsing du corpus et indexation du fichier trace

Dans cette étape, l'application va proposer au tuteur un bouton « ouverture du corpus » qui doit en définitive permettre de sélectionner le répertoire qui sera analyser pour en extraire le type de parcours. Ce mécanisme se déroule en deux phase, une phase de chargement du chemin, une seconde phase pour l'indexation du fichier trace précédent, cette étape a pour but d'ajouter a ce fichier trace des donnée signifiant la présence ou non des concepts du cours dans chaque page

```
<?xml version="1.0" encoding="UTF-8"?>
<user user_name="NADIR" user_id="Id01" user_login="Nad_PC" da
  <page title="Les boucles" order="1" date="09/06/2012" time=
    <c1 name="structure de donnees"></c1>
    <c2 name="les procedures"></c2>
    <c3 name="les conditions"></c3>
    <c4 name="les boucles"></c4>
  </page>
  <page title="Cours d'algorithmique : les boucles" order="2"
    <c1 name="structure de donnees"></c1>
    <c2 name="les procedures"></c2>
    <c3 name="les conditions"></c3>
    <c4 name="les boucles"></c4>
  </page>
```

Figure 17 : fichier trace simple (avant indexation)

```
<?xml version="1.0" encoding="UTF-8"?>
<user user_name="NADIR" user_id="Id01" user_login="Nad_PC" dat
  <page title="Les boucles" order="1" date="09/06/2012" time="
    <c1 name="structure de donnees">1</c1>
    <c2 name="les procedures">0</c2>
    <c3 name="les conditions">1</c3>
    <c4 name="les boucles">0</c4>
  </page>
  <page title="Cours d'algorithmique : les boucles" order="2"
    <c1 name="structure de donnees">1</c1>
    <c2 name="les procedures">0</c2>
    <c3 name="les conditions">1</c3>
    <c4 name="les boucles">0</c4>
  </page>
```

Figure 18 : Fichier trace indexé

4.3. Traitement et lecture du fichier trace indexé

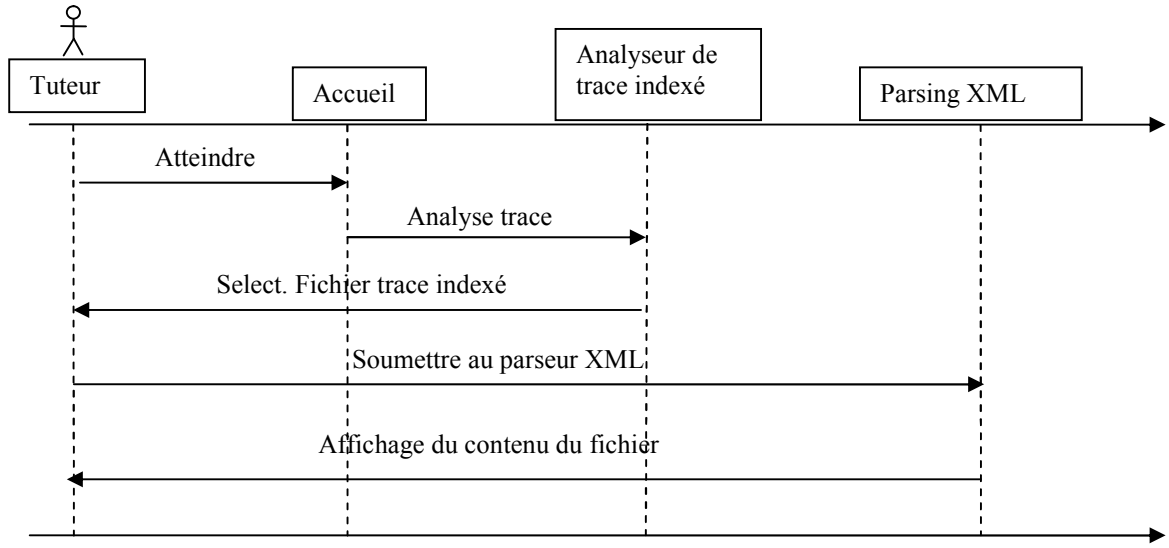


Figure 19 : Traitement et lecture du fichier trace indexé

Dans se cas-ci-dessus le tuteur lance le module d'analyse des données d'usage du web, cela se traduit par le chargement du fichier trace indexé, puis l'affichage de son contenu a l'écran. Cette interface utilisateur lui permettra notamment trois possibilités :

- l'affichage d'une des pages web dont l'URL apparaitra et sera sélectionnée préalablement.
- L'analyse proprement dite des données contenu dans ce fichier indexé et extraire le type de parcours de l'apprenant
- Visualiser les concepts rencontrés en dehors de sa plateforme tout au long de sa session d'apprentissage.

4.4. Extraction du type de parcours

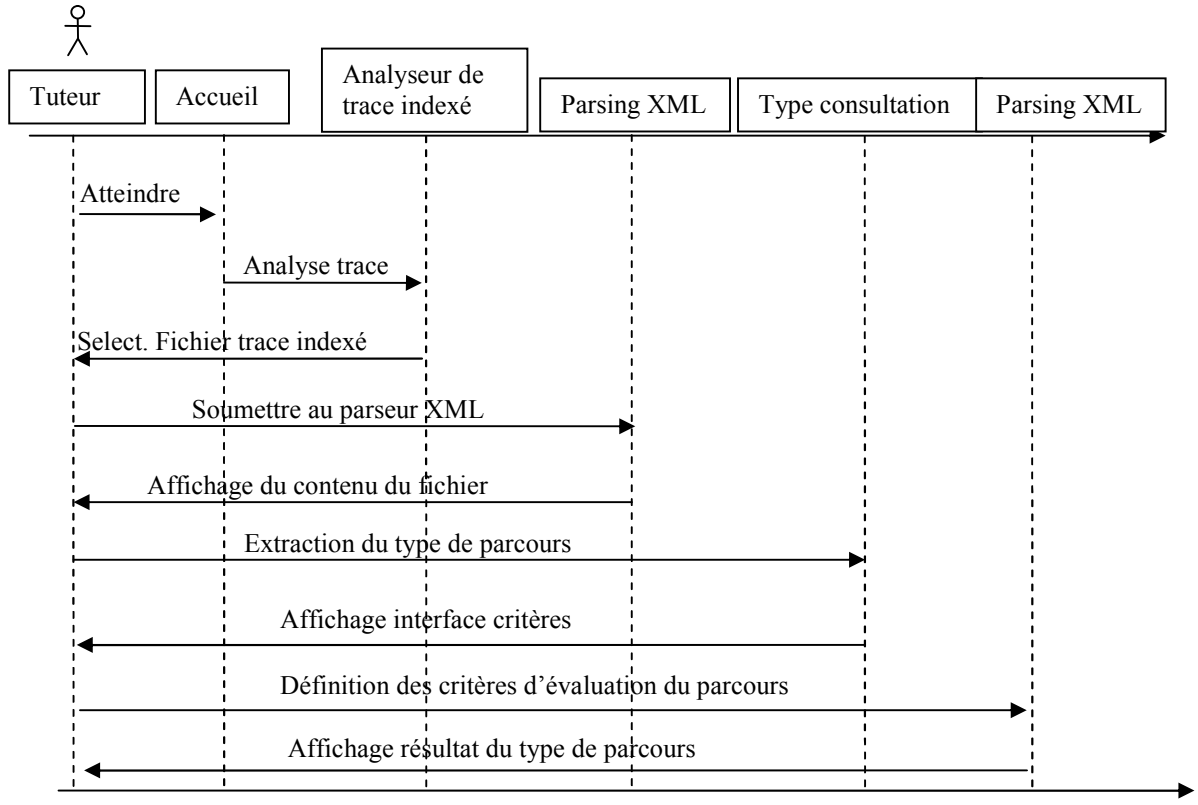


Figure 20 : Extraction du type de parcours

Pour exécuter cette partie de l'application, le tuteur aura à sa disposition le bouton « concepts les plus visités » pour lancer ce module. En suite une interface lui sera proposé pour insérer les valeurs de référence par rapport aux quelles les données seront traduites.

5. Réalisation

Dans cette partie nous allons montrer des interfaces utilisateur de notre application, nous montrerons des captures d'écran d'un exemple d'exécution qui mettra en valeur le bon déroulement et la suite logique des cas d'utilisation.

Nous allons donc commencer par un exemple d'aspiration de pages web, dont les URL sont contenu dans un fichier trace sous format XML. La deuxième étape consistera à faire en sorte de parcourir les pages stocké localement et a indexé le fichier trace précédent avec les concepts référencés des les pages.

Finalement, nous donnerons quelques exemples d'exécution du module de d'analyse de type de parcours et cela en donnant des valeurs de référence différentes, ainsi, cela démontrera que les données d'usage du web d'un apprenant peuvent être interprétées différemment selon la référence que l'on fixe au départ.

5.1. L'accueil

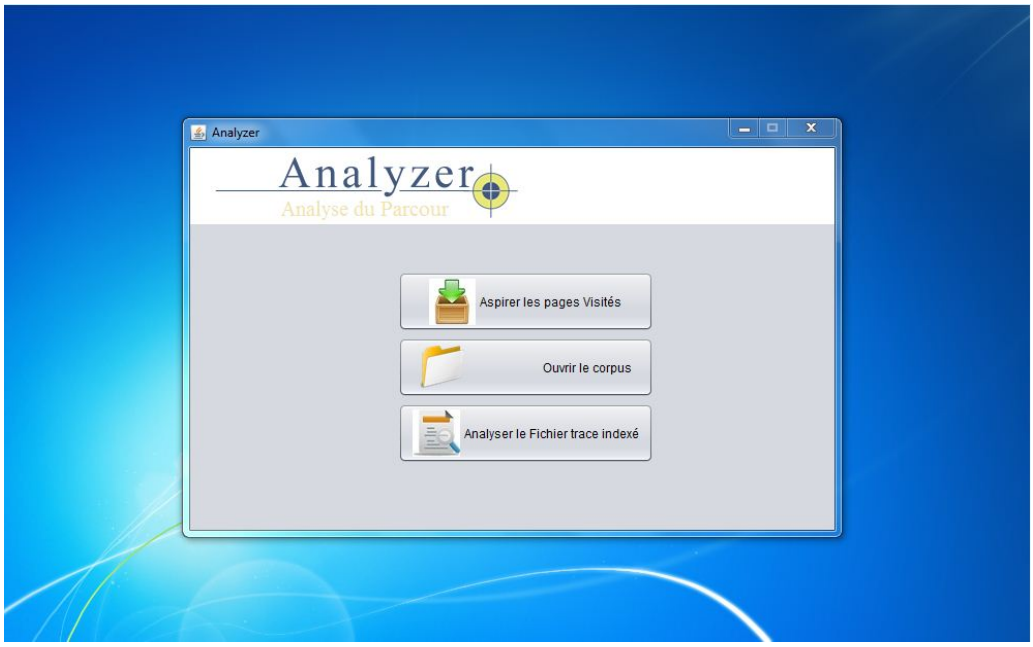


Figure 21 : vue accueil

Cette interface représente le point de repère de l'utilisateur (tuteur), a partir de cet emplacement, il peut a atteindre tout les module programmés dans cette application.

Il peut aspirer les pages, charger le corpus et analyser les données et en affiché le contenu.

5.2. Aspiration

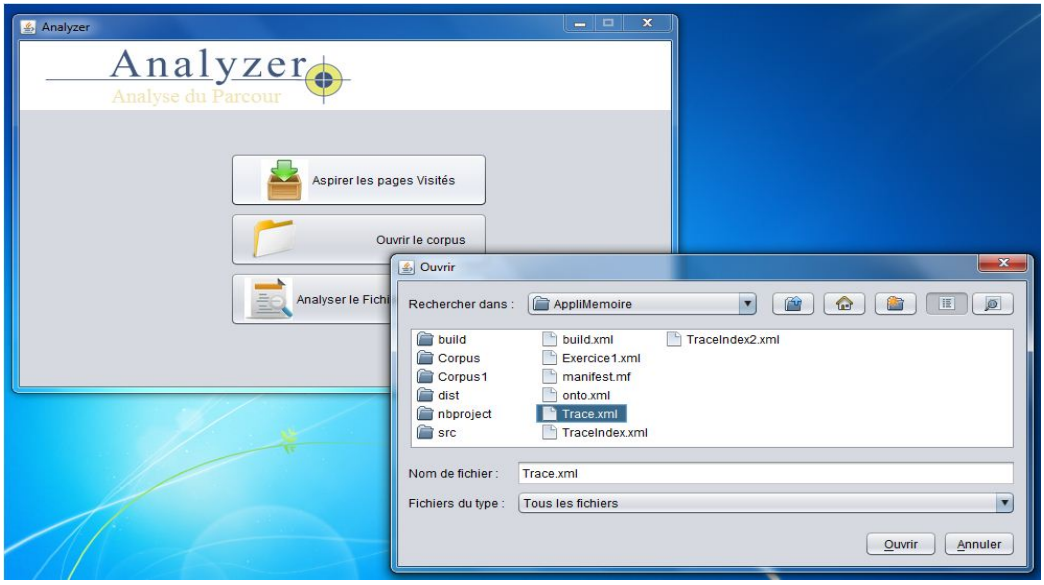


Figure 22 : aspiration

L'aspiration consiste a utiliser le fichier trace pour télécharger les pages web visité selon l'ordre de visite ; dans cet exemple le fichier en question est nommé « **Trace.xml** »

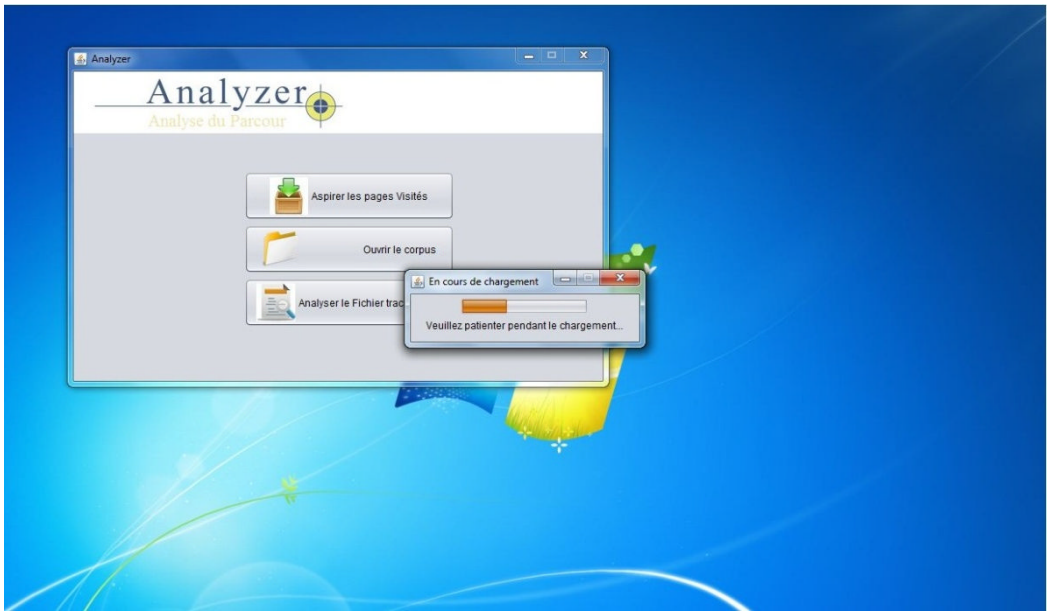


Figure 23 : aspiration en cours

Cette capture nous montre une boite de dialogua signifiant que le téléchargement du corpus est en cour d'exécution. Cela renseigne sur le déroulement des événements et aussi du temps que cela prendra.

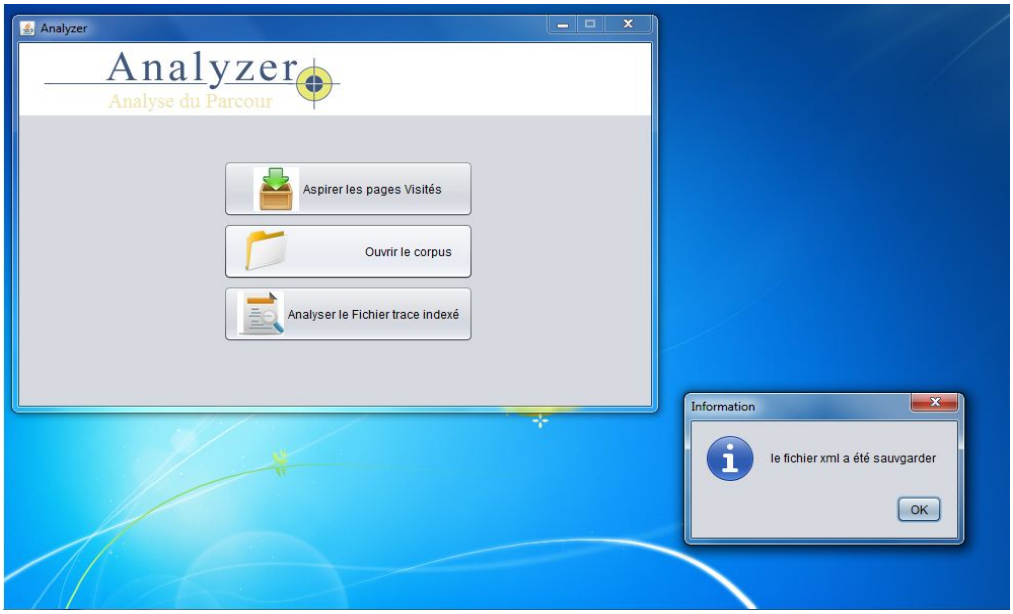


Figure 24 : confirmation

Une fois le téléchargement des pages visité fait, le programme annonce le succès de l'opération et renseigne l'utilisateur de l'endroit ou sont sauvegardés les documents récupérés.

5.3. Indexation

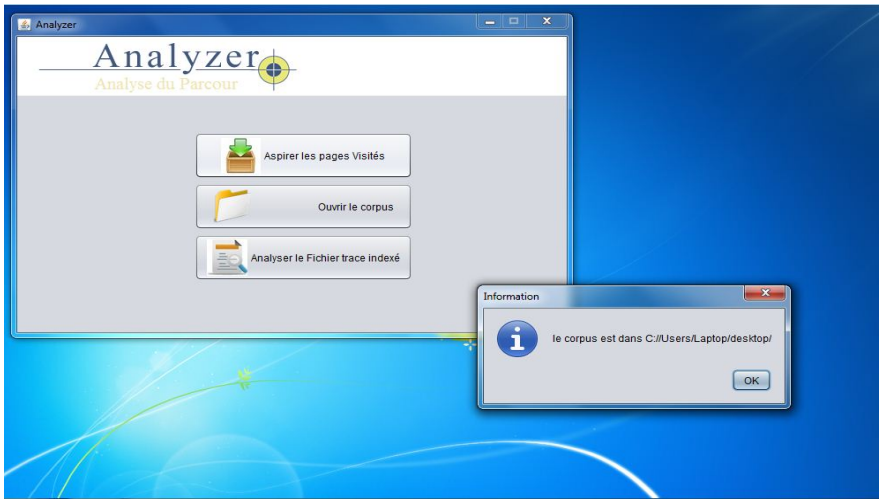


Figure 25 : confirmation indexation

L'indexation se déroule de la même manière que l'aspiration, c'est-à-dire qu'il y a une phase de sélection de chemin, dans ce cas présent, le chemin du corpus sur le disc, en suite les pages sont parsés par un module qui compare les termes contenus a l'ontologie du cours, on pale de « **projection** », à l'issue de cette projection, le fichier

trace se voit modifié, on y ajoute des valeurs booléennes représentants l'existence ou non des concepts du cours dans les pages visités.

5.4. Analyse du type de parcours

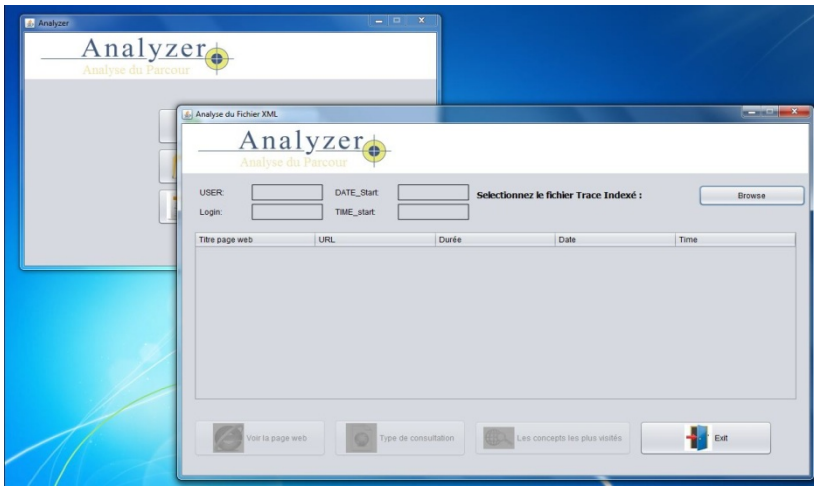


Figure 26 : analyse du parcours

L'analyse du type de parcours se base essentiellement sur le contenu du fichier trce indexé ; nombre de moteur de recherche utilisé, temps passé sur chaque page et temps moyen de la session de travail web ainsi que le calcul de l'indice de Jaccard.

Sélection du fichier trace « indexé »

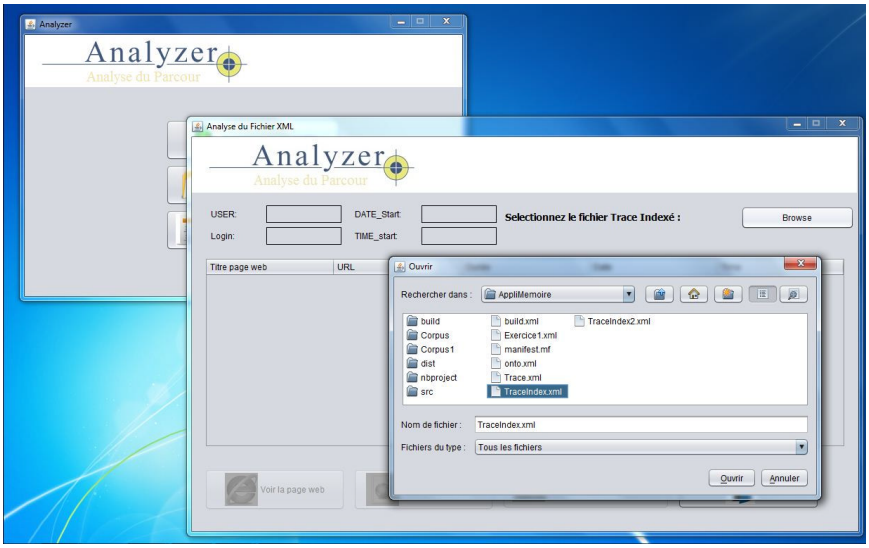


Figure 27 : sélection trace indexé

Une fois ce fichier trace sélectionné, il permettra de visualiser toutes les informations utile telles que le nom des pages, les URL, la durée de consultation…

Affichage du contenu du fichier trace indexé

A ce niveau les informations sont encore bruts, dans cet exemple la liste d'information est courte, donc facile a lire, mais le problème se pose au moment ou un apprenant consulte un nombre important et ensuite on doit gérer plusieurs apprenants, c'est la qu'intervient le module d'analyse automatique de traces.

Affichage des concepts les plus visités

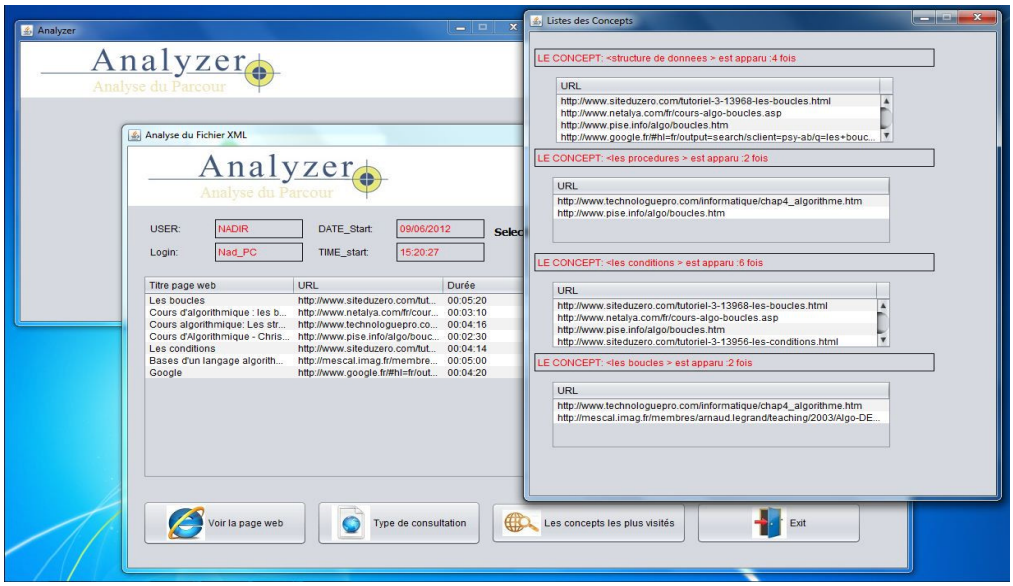


Figure 28 : affichage des concepts vus par l'apprenant

Ici nous montrons une situation dates utile, où le tuteur peut directement consulter les concepts les plus recherché ou les plus étudiés en dehors de la plateforme, cela peut aider a mieux cerner les lacune des apprenant ou encore soit a enrichir le cours ou l'améliorer.

Définition des critères et interprétation des résultats

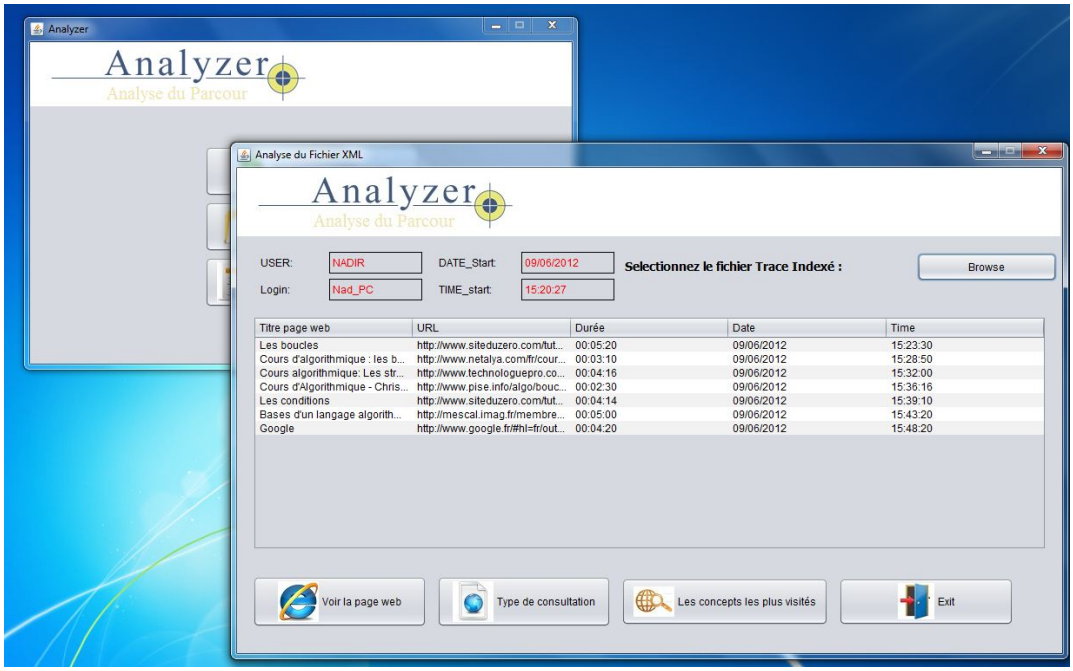


Figure 29 : définitions des critères d'évaluation

La définition de critère est en somme un point de repère par rapport au quels les donnée seront traduites en information. Si les valeurs change, les résultats changerons de plus cela reflète la réalité si l'on suppose que l'on est dans une situation ou l'on a deux groupes d'apprenant ; l'un plus rapide que l'autre, a e moment la, le temps de référence pour une durée moyenne de consultation se verra être modifié d'un groupe à l'autre ainsi que le seuil de Jaccard.

Tout d'abord, il faut bien comprendre que le fichier trace contient des « **données** » et de ce fait elles sont statique par rapport à leur traductionnel en « **informations** ».

Le fichier trace donne un seuil de Jaccard de, une durée moyenne de et content un seul moteur de recherche.

Premier cas

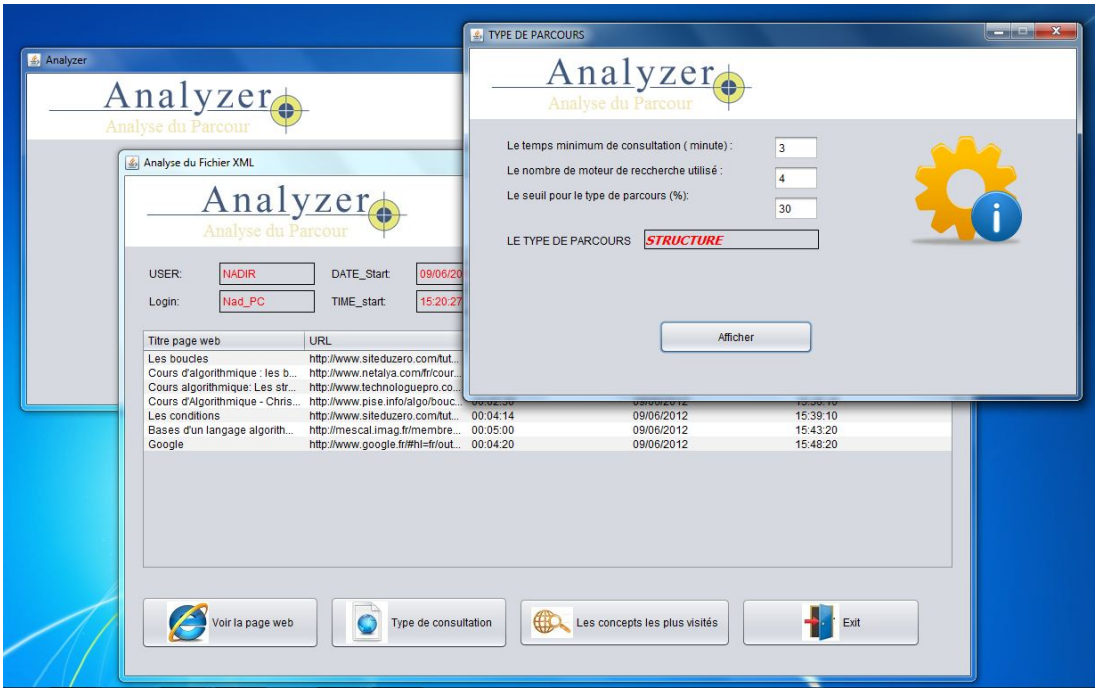


Figure 30 : parcours structuré

Ici nous fixons à l'avance que la durée moyenne de consultation est de 3 minutes, le nombre de moteur de recherche est de 4 ainsi qu'un seuil de Jaccard est de 30%, le résultat « structuré » est due au fait que les contrainte de départ respecte les donnée du fichier race.

Deuxième cas

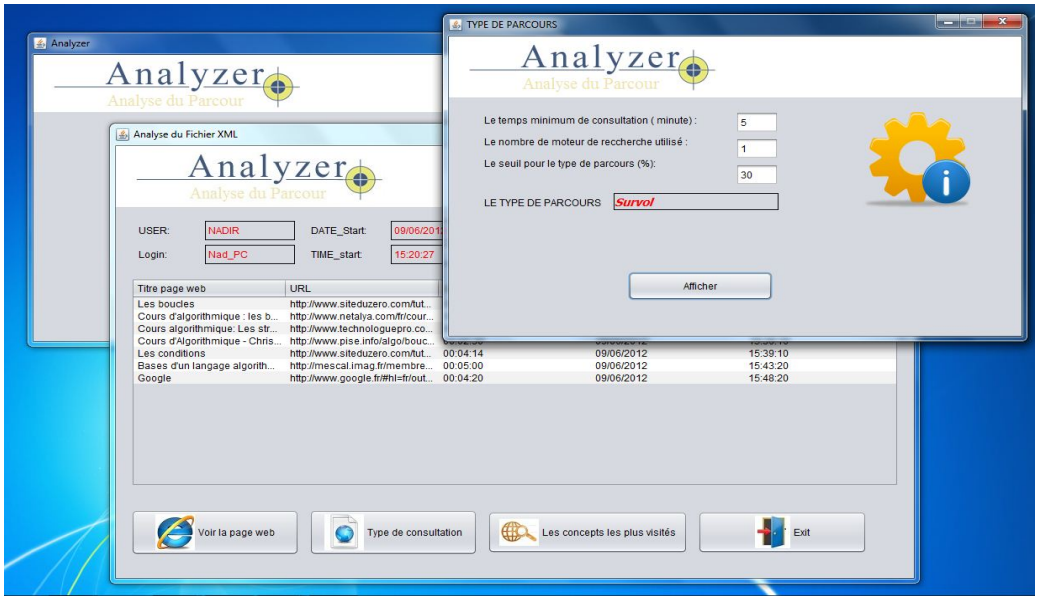


Figure 31 : survol

Dans ce deuxième cas, le survol est caractéristique d'une durée moyenne par page trop courte (par rapport a la référence 5minutes).

Troisième cas

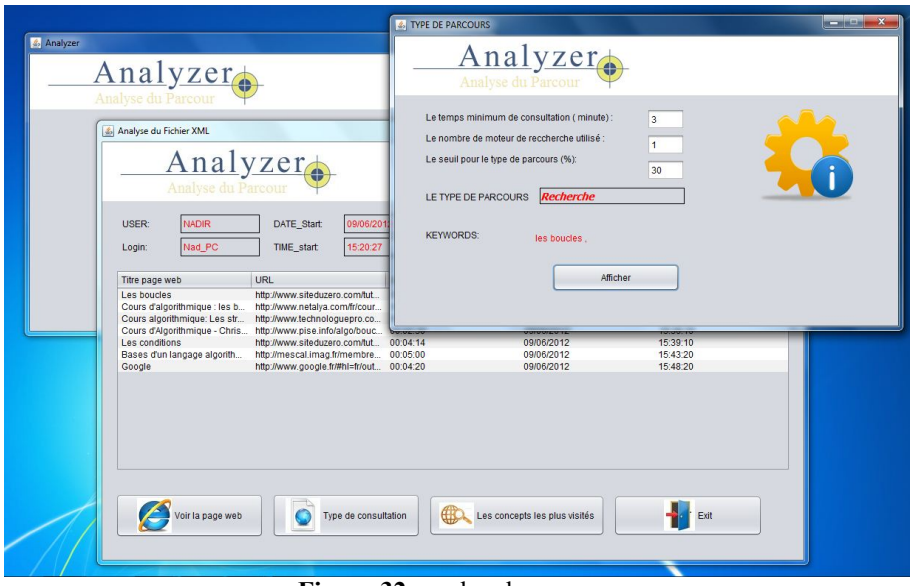


Figure 32 : recherche

Ici le nombre de moteur de recherche pour le quel le système définira le parcours comme étant un parcours recherche correspond a la valeur fixé.

Quatrième cas

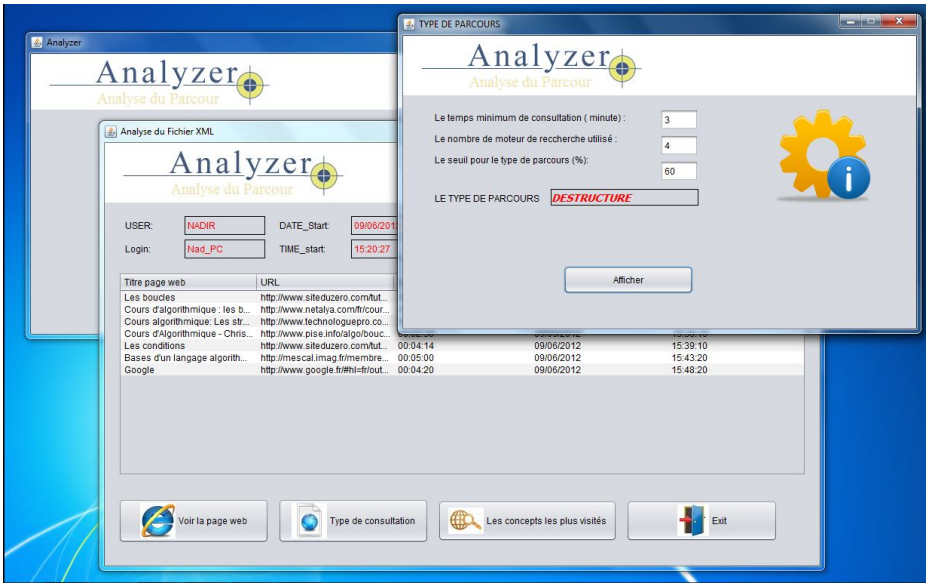


Figure 33 : déstructuré

Dans cette situation, les valeurs introduites sont en quelques sorte trop exigeantes vis-à-vis du contenu de la trace ; le temps fixé correspond au fichier donc ce n'est pas un survol, le nombre de moteur de recherche est bien supérieur a celui utilisé, cela indique la dernière mesure significative est l'indice de Jaccard, et comme il trop inferieur, cela donne un parcours « **déstructuré** ».

6. Conclusion

Dans ce chapitre, nous avons essayé de donner un compte rendu clair et concis des résultats aux quels nous sommes arrivés.

Nous avons montré la difficulté de traiter une telle masse d'information, d'où l'idée d'un système d'infirmation pour gérer et décortiquer toute ces données, mais aussi nous avons souligné l'importance de la traduction des donnée en information et cela en prenant en compte les diverses interprétation possible dues aux critères non standardisé.

Dans cette application, nous avons tenté de mettre en valeur l'utilité d'un traitement automatique des données d'usage du Web ainsi que la prise en considération des valeurs aux quel en mesure nos résultats et cela pour donner un produit final qui se rapproche le plus possible de la réalité.

Conclusion générale

Notre mémoire, nous a permit de découvrir de nouveaux concepts aussi bien pratique que théorique, nous avons pu maitriser certains aspect d'un nouveau domaine qui est les systèmes d'apprentissages, mais aussi de mettre en pratique nos acquis au terme de notre cursus universitaires.

A travers ce travail nous avons vu le besoin d'observer les interactions de l'apprenant durant les sessions d'apprentissages afin de définir le parcours de visite d'un apprenant et de reconnaitre les difficultés rencontré et de marquer les concepts non bien assimilés pendant son apprentissage et ceci à travers le recueil et l'interprétation des traces de navigation web durant une session d'apprentissage.

Ainsi, nous avons considéré seulement le contexte du web, nos perspectives visent à faire une analyse sur le contexte communication, c'est-à-dire détecter les concepts partagés.

ANNEXE

6.1. ANNEXE A

I. L'API JDOM

JDOM est une API open source Java dont le but est de représenter et manipuler un document XML de manière intuitive pour un développeur Java sans requérir une connaissance pointue de XML. Par exemple, JDOM utilise des classes plutôt que des interfaces. Ainsi pour créer un nouvel élément, il faut simplement instancier une classe.

Malgré la similitude de nom entre JDOM et DOM, ces deux API sont très différentes. JDOM est une API uniquement Java car elle s'appuie sur un ensemble de classes de l'API Java notamment celles de l'API Collection.

Le site officiel de l'API est à l'url http://www.jdom.org/

I.1 L'historique de JDOM :

En 2000, Brett McLaughlin et Jason Hunter développent une nouvelle API dédiée aux traitements de documents XML en Java. Le but est de fournir une API plus conviviale à utiliser en Java que SAX ou DOM.

L'historique de JDOM est marquée par plusieurs versions bêta et stables :

- La version bêta 3 est diffusée en avril 2000.
- La version 1.0 a été publié en septembre 2004.
- La version 1.1 a été publiée en novembre 2007.

JDOM a fait l'objet d'une spécification sous la Java Specification Request numéro 102 (JSR-102) : malheureusement celle-ci n'a pas aboutie.

I.2 La présentation de JDOM

Le but de JDOM n'est pas de définir un nouveau type de parseur mais de faciliter la manipulation au sens large de document XML : lecture d'un document, représentation sous forme d'arborescence, manipulation de cet arbre, définition d'un nouveau document, exportation vers plusieurs formats cibles ...

Dans le rôle de manipulation sous forme d'arbre, JDOM possède moins de fonctionnalités que DOM mais en contre partie il offre une plus grande facilité pour répondre aux cas les plus classiques d'utilisation.

Cette facilité d'utilisation de JDOM lui permet d'être une API dont l'utilisation est assez répandue.

JDOM est donc un modèle de documents objets open source dédié à Java pour encapsuler un document XML. JDOM propose aussi une intégration de SAX, DOM, XSLT et XPath.

JDOM n'est pas un parseur : il a d'ailleurs besoin d'un parseur externe de type SAX ou DOM pour analyser un document et créer la hiérarchie d'objets relative à un document XML. L'utilisation d'un parseur de type SAX est recommandée car elle consomme moins de ressources que DOM pour cette opération. Par défaut, JDOM utilise le parseur défini via JAXP.

Un document XML est encapsulé dans un objet de type Document qui peut contenir des objets de type Comment, ProcessingInstruction et l'élément racine du document encapsulé dans un objet de type Element.

Les éléments d'un document sont encapsulés dans des classes dédiées : Element, Attribute, Text, ProcessingInstruction, Namespace, Comment, DocType, EntityRef, CDATA.

Un objet de type Element peut contenir des objets de type Comment, Text et d'autres objets de type Element.

A l'exception des objets de type Namespace, les éléments sont créés en utilisant leur constructeur.

JDOM vérifie que les données contenues dans les éléments respectent la norme XML : par exemple, il n'est pas possible de créer un commentaire contenant deux caractères moins qui se suivent. Une fois un document XML encapsulé dans un arbre d'objets, il est possible de modifier cet arbre dans le respect des spécifications de XML.

JDOM permet d'exporter un arbre d'objets d'un document XML dans un flux, un arbre DOM ou un ensemble d'événements SAX.

JDOM interagit donc avec SAX et DOM pour créer un document en utilisant ces parseurs ou pour exporter un document vers ces API, ce qui permet de facilement intégrer JDOM dans des traitements existants. JDOM propose cependant sa propre API.

I.3Les fonctionnalités et les caractéristiques :

JDOM propose plusieurs fonctionnalités :

- Création de documents XML

- Encapsulation d'un document XML sous la forme d'objets Java de l'API
- Exportation d'un document dans un fichier, un flux SAX ou un arbre DOM
- Support de XSLT
- Support de XPath

Les points caractéristiques de l'API JDOM sont :

- elle est développée spécifiquement en et pour Java en utilisant les fonctionnalités de Java au niveau syntaxique et sémantique (utilisation des collections de Java 2, de l'opérateur new pour instancier des éléments, redéfinition des méthodes equals(), hashCode(), toString(), implémentation des interfaces Cloneable et Serializable, ...)
- elle se veut intuitive et productive notamment grâce à des classes dédiées à chaque élément instancié via leur constructeur et l'utilisation de getter/setter

Exemple pour obtenir le texte d'un élément

DOM : String content = element.getFirstChild().getValue();

JDOM : String text = element.getText();

- elle se veut rapide et légère
- elle veut masquer la complexité de certains aspects de XML tout en respectant ses spécifications
- elle doit permettre les interactions entre SAX et DOM. JDOM peut encapsuler un document XML dans un hiérarchie d'objets à partir d'un flux, d'un arbre DOM ou d'événements SAX. Il est aussi capable d'exporter un document dans ces différents formats.

Il est légitime de se demander qu'elle est l'utilité de proposer une nouvelle API pour manipuler des documents XML en Java alors que plusieurs standards existent déjà. En fait le besoin est réel car JDOM propose des réponses à certaines faiblesses de SAX et DOM.

DOM est une API indépendante de tout langage : son implémentation en Java ne tient donc pas compte des spécificités et standards de Java ce qui rend sa mise en œuvre peu aisée. JDOM est plus intuitif et facile à mettre en œuvre que DOM.

Comme DOM, JDOM encapsule un document XML entier dans un arbre d'objets. Par contre chaque élément du document est encapsulé dans une classe dédiée selon son type et non sous la forme d'un objet de type Node.

JDOM peut être utilisé comme une alternative à DOM pour manipuler un document XML. JDOM n'est pas un remplaçant à DOM puisque ce n'est pas un parseur et il propose des interactions avec DOM en entrée et en sortie.

L'utilisation de DOM requiert de nombreuses ressources notamment à cause de son API qui de surcroît n'est pas intuitive en Java puisque développée de façon indépendante de tout langage et que son organisation est proche de celle des spécifications XML (tous les éléments sont des Nodes par exemple).SAX est particulièrement bien adapté à la lecture rapide avec peu de ressources d'un document XML mais son modèle de traitement par événements n'est pas intuitive et surtout SAX ne permet de modifier ni de naviguer dans un document. JDOM propose d'apporter une solution à ces différents problèmes dans une seule et même API.

I.4 Les différentes entités de JDOM

Pour traiter un document XML, JDOM définit plusieurs entités qui peuvent être regroupées en trois groupes :

- les éléments de l'arbre
  - le document : la classe Document
  - les éléments : la classe Element
  - les commentaires : la classe Comment
  - les attributs : la classe Attribute
  - etc ...
- les entités pour obtenir un parseur :
  - les classes SAXBuilder et DOMBuilder
- les entités pour produire un document
  - les classes XMLOutputter, SAXOutputter, DOMOutputter

Ces classes sont regroupées dans cinq packages :

- org.jdom
- org.jdom.adapters
- org.jdom.input
- org.jdom.output
- org.jdom.transform

6.2. ANNEXE B

I.Présentation de XML

XML (entendez eXtensible Markup Language et traduisez Langage à balises étendu, ou Langage à balises extensible) est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (markup).

Contrairement à HTML, qui est à considérer comme un langage défini et figé (avec un nombre de balises limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte (Qui n'a jamais désiré une balise qui n'existait pas ?).

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir.

En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement À HTML). Ainsi, XML permet de séparer le contenu de la présentation.. ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations !

XML a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) dès 1996. Depuis le 10 février 1998, les spécifications XML 1.0 ont été reconnues comme recommandations par le W3C, ce qui en fait un langage reconnu. (Tous les documents liés à la norme XML sont consultables et téléchargeables sur le site web du W3C, http://www.w3.org/XML/)

XML est un sous ensemble de SGML (Standard Generalized Markup Language), défini par le standard ISO8879 en 1986, utilisé dans le milieu de la Gestion Electronique Documentaire (GED). XML reprend la majeure partie des fonctionnalités de SGML, il s'agit donc d'une simplification de SGML afin de le rendre utilisable sur le web !

II.Mise en page de XML

XML est un format de description des données et non de leur représentation, comme c'est le cas avec HTML. La mise en page des données est assurée par un langage de mise en page tiers. A l'heure actuelle (fin de l'année 2000) il existe trois solutions pour mettre en forme un document XML :

- CSS (Cascading StyleSheet), la solution la plus utilisée actuellement, étant donné qu'il s'agit d'un standard qui a déjà fait ses preuves avec HTML

- XSL (eXtensible StyleSheet Language), un langage de feuilles de style extensible développé spécialement pour XML. Toutefois, ce nouveau langage n'est pas reconnu pour l'instant comme un standard officiel

- XSLT (eXtensible StyleSheet Language Transformation). Il s'agit d'une recommandation W3C du 16 novembre 1999, permettant de transformer un document XML en document HTML accompagné de feuilles de style

III. Structure des documents XML

XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (Document Type Definition). Il s'agit d'un fichier décrivant la structure des documents y faisant référence grâce à un langage adapté. Ainsi un document XML doit suivre scrupuleusement les conventions de notation XML et peut éventuellement faire référence à une DTD décrivant l'imbrication des éléments possibles. Un document suivant les règles de XML est appelé document bien formé. Un document XML possédant une DTD et étant conforme à celle-ci est appelé document valide.

IV. Décodage d'un document XML

XML permet donc de définir un format d'échange selon les besoins de l'utilisateur et offre des mécanismes pour vérifier la validité du document produit. Il est donc essentiel pour le receveur d'un document XML de pouvoir extraire les données du document. Cette opération est possible à l'aide d'un outil appelé analyseur (en anglais parser, parfois francisé en parseur).

Le parseur permet d'une part d'extraire les données d'un document XML (on parle d'analyse du document ou de parsing) ainsi que de vérifier éventuellement la validité du document.

V. Les avantages de XML

Voici les principaux atouts de XML :

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

- Autodescriptif et extensible

- 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 (c'est-à-dire 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 : Mathematical Markup Language permet de représenter des formules mathématique

- CML : Chemical Markup Language permet de décrire des composés chimiques

- SMIL : Synchronized Multimedia Integration Language permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte,...

Bibliographie

[1] Cram, D., Jouvin, D., et Mille, A. (2007). Visualisation interactive de traces et flexivité : application à l‟EIAH collaboratif synchrone eMédiathèque.STICEF, volume 14.

[2] Jermann, P., Soller, A., et Muehlenbrock, M., « From Mirroring to Guiding: A Review State of the Art Technologyfor supporting Collaborative Learning ». Proceedings of the First European Conference on Computer-Supported Collaborative Learning, 2001.

[3] Pernin, J.-P. (2005). CSE, un modèle de traitement de traces, CLIPS-IMAG.

[4] Champin, P.-A., et Prié, Y. (2002). « MUSETTE : un modèle pour réutiliser l'expérience sur le web sémantique ». Journées scientifiques Web sémantique, Paris.

[5] Settouti, L., Prié, Y. , Mille, A. , et Marty, J-C. (2006). « Système à base de traces pour l'apprentissage humain ». Colloque international TICE 2006, Technologies de l‟Information et de la Communication dans l‟Enseignement Supérieur et l‟Entreprise.

[6] Fisher, F. (2005). L‟utilisation des traces numériques dans l‟enseignement à distance. Rapport de recherche bibliographique.

[7] Beauvisage, T. (2004). Sémantique des parcours des utilisateurs sur le Web. Thèse de doctorat, Université de Paris X : 361p.

[8] Michel, C., Prié, Y., Le Graet, L. (2005). « Construction d‟une base de connaissance pour l‟évaluation de l‟usage d‟un environnement STIC ». 17eme Conférence Internationale Francophone sur l‟Interaction Homme-Machine, Toulouse, France, pp. 199-202.

[9] Kaleidoscope JEIRP TRAILS, (2005) "State of art of tracking and analysing usage". Délivrable de la tâche 32.3 du projet DPULS» .2005

[10] Gwenegan, R. (2005). Structuration et analyse de traces hybrides issues de situation d'apprentissage. Rapport de Master, Laboratoire CLIPS-IMAG.

[11] VALERY PSYCHE « Proposition d’une méthode d’ingénierie ontologique pour les EIAH : application aux systèmes auteurs » Programme de doctorat en informatique, Mai 2004 Université du Québec à Montréal Canada

[12] T. R. Gruber. A translation approach to portable ontology specications.Knowl. Acquis., 5(2):199-220, 1993.

[13] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout.Enabling technology for knowledge sharing. AI Magazine, 12(3):36-56, August 1991

[14] THOMAS R. GRUBER « A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, Vol.5 » 1993

[15] Nicola Guarino. « Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology » 1997. CIE 1997, M. T. Pazienza (Eds.), Springer Verlag, pp. 139-170.

[16] GRUNINGER M. & FOX M. S « Methodology for the design and evaluation of ontologies, in Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing » 1995.

[17] M. IKEDA, Y. HAYASHI, J. LAI, W. CHEN, J. BOURDEAU, K. SETA AND R. MIZOGUCHI « An ontology more than a shared vocabulary. Workshop on Ontologies for Intelligent Educational Systems » Juillet 19-23, 1999 Ninth International Conference on Artificial Intelligence in Education, AI-ED'99. Le Mans, France http://www.ei.sanken.osaka-u.ac.jp/aied99/aied99-onto.html

[18] GOMEZ PEREZ A., BENJAMINS V.R. « Overview of knowledge sharing and components: Ontologies and problem Solving Methods. Workshop on Ontologies and problem-SolvingMethods » 1999. Stockholm (Suède).

[19] FRÉDÉRIC FÜRST ingénierie des connaissances octobre 2002