Recherche d'information dans les Documents Pédagogiques Structurés Adaptée aux Besoins Spécifiques des Apprenants.

Iddir Ounnaci, Rachid Ahmed-Ouamer

** Laboratoire de Recherche en Informatique LARI, Département d'Informatique*
Université Mouloud Mammeri de Tizi-Ouzou, 15000 Tizi-Ouzou, Algérie {iddsoft, rachid.ahmedouamer}@yahoo.fr

Résumé : *Le web évolue vers de plus en plus de structuration et de prise en compte de la sémantique, en particulier avec XML et les ontologies. Par ailleurs, l'accès aux informations du web nécessite l'usage d'outils de recherche d'information (RI). De nombreuses méthodes issues de la RI traditionnelle ont été étendues aux documents structurés. D'autre part, des approches ont été proposées pour prendre en compte la sémantique dans les documents structurés à l'aide notamment de ressources sémantiques externes à la collection de documents initiale sur lesquelles il est nécessaire de disposer de mesures de similarité sémantique pour pouvoir effectuer des comparaisons entre concepts. La plupart des approches précédentes ne prennent pas en compte les relations entre concepts et ne sont pas adaptées aux besoins spécifiques des apprenants. Dans ce papier est proposé un système de RI sémantique dans les documents pédagogiques structurés, adaptée aux besoins et aux préférences de l'apprenant. Cette approche est basée sur* une représentation des nœuds de l'arbre d'un document et de la requête par des vecteurs sémantiques de concepts. *Les tests effectués montrent la faisabilité de l'approche proposée.*

Abstract: *The web is increasingly moving towards structuring and to king into consideration of semantics, particularly with XML and ontology. In addition, access to information requires the use of web tools for information retrieval (IR). Many methods from traditional IR were extended to structured documents. On the other hand, approaches have been proposed to account for the semantics in structured documents by using such external semantic resources to the collection of original documents on which it is necessary to provide semantic similarity measures in order to perform comparisons between concepts. Most previous approaches do not take into account the relations between concepts and are not tailored to the specific needs of learners. In this paper is proposed a semantic IR system of structured educational documents, adapted to the needs and preferences of the learner. This approach is based on a representation of the nodes of the tree of a document and of the query by semantic vectors of concepts. Tests show the feasibility of the proposed approach.*

Mots-clés : Recherche d'information – XML – Document structuré – Document pédagogique – Ontologie – Web sémantique – Modèle élève – e-Learning.
Keywords*:* Information Retrieval - XML - Structured Document - Pedagogic document- Ontology - Semantic web - Student model - e-Learning.

1. INTRODUCTION GENERALE

Avec l'augmentation rapide du volume documentaire stocké sous format numérique, et l'avènement du Web, la quantité d'informations disponible ne cesse de croître au cours de ces dernières années. Il est devenu alors très difficile de trouver une information ou un document qui répond à un besoin utilisateur. Il a fallu donc envisager le développement des outils automatiques qui permettent l'accès ciblé et efficace à cette masse de données.

De plus, la notion de document électronique a considérablement évolué. Nous sommes passés d'un monde où le concept dominant était celui du document *plat* à savoir d'un texte constitué d'une suite de mots sans aucune information de structure, à un monde où le document est devenu un objet plus complexe, structuré, et pouvant comporter déférents médias. Avec cette évolution de la nature des sources d'informations, de nouveaux besoins qui visent à exploiter la richesse présentée dans ces documents sont apparus. Le format d'un document est aujourd'hui défini par une structure logique décrite par des instances du langage XML. Le format XML permet par exemple de structurer un document de manière logique, par exemple sous forme de chapitres, sections, et paragraphes. Chaque document XML est ainsi défini par une arborescence logique formée d'éléments (l'information structurelle) et de son contenu (image, texte, etc.). L'arborescence du document donne la possibilité d'accéder à des éléments plus fins que le document entier et permet d'envisager une recherche plus précise et focalisée.

La plupart des approches actuelles dans la recherche des documents semi-structurés (documents XML) sont basées sur des systèmes d'indexation à base de mots clés ou encore sur les termes. Les seules informations utilisées concernant ces termes sont leurs fréquences d'apparition dans les documents ou les éléments du document. Ainsi, ces approches ne prennent pas en considération le sens du mot. Elles ne distinguent pas les mots selon leurs contextes d'apparition. Ces termes présentent une forte ambigüité. En effet un mot peut varier de sens selon le contexte où il apparait (phénomène de *polysémie*). Aussi, ces approches ne prennent pas en compte la *synonymie* (deux mots graphiquement différents peuvent avoir le même sens). Par conséquent, dans ces systèmes, il est impossible de trouver des parties des documents représentés par un mot M_1 synonyme d'un mot M_2, où M_2 représentant une requête.

Par conséquent, un SRI basé sur les mots peut renvoyer un document non pertinent, bien que le document satisfasse la requête. Pour pallier à ce manque, de nouveaux modèles flexibles ont été proposés. C'est l'objet de la recherche d'information sémantique (conceptuelle).

Dans ce papier est proposé un modèle de recherche d'information sémantique (conceptuelle) dans des documents pédagogiques semi-structurés, adapté aux besoins et aux préférences de l'apprenant, afin de pouvoir facilement effectuer la correspondance entre la requête apprenant et les index des documents (pédagogiques) XML disponibles dans

une collection, et cela par l'utilisation de ressources sémantiques externes telles que les ontologies pour améliorer l'efficacité du processus de recherche.

La section 2 présente la RI conceptuelle dans les documents semi-structurés pédagogiques où l'ontologie de l'e-learning de l'informatique développée est précisée. La section 3 est consacrée à la RI sémantique dans les documents semi-structurés pédagogiques. Dans la section 4 est détaillée notre approche d'indexation et d'interrogation (section 5) qui se base sur une représentation des nœuds de l'arbre d'un document et de la requête par des vecteurs sémantiques de concepts. La section 6 est relative à la mise en œuvre de cette approche dans le contexte du web sémantique. La dernière section fait la synthèse de cette étude.

2. Recherche d'information Conceptuelle Dans Les Documents Semi-Structures Pedagogiques

La RI conceptuelle dans les documents pédagogiques semi-structurés consiste à identifier les éléments des documents XML les plus pertinents par rapport à une requête apprenant, en prenant compte de la *sémantique des termes* d'indexation et du *profil de l'apprenant.* Ce type d'indexation passe du niveau des mots au niveau des concepts (les sens des mots) pour mieux décrire le contenu du document et de la requête de l'apprenant, et cela en utilisant des ressources sémantiques. Ces ressources offrent aussi des connaissances sur les relations entre les concepts dans le texte, ce qui permet une meilleure représentation du document et de la requête. Différents types de ressources sémantiques peuvent être distingués parmi lesquels se trouvent les *ontologies.*
Historiquement, l'ontologie est un concept philosophique qui désigne la science de l'être en général. Elle décrit une théorie à propos de la nature de l'existence selon le paradigme "On ne cherche pas à comprendre le monde mais à le représenter " [Roche 05]. Plus tard, le terme est repris en informatique dans le domaine de l'intelligence artificielle, pour désigner une ressource sémantique caractérisée surtout par une structure hiérarchique de termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances, afin de résoudre les problèmes de modélisation des connaissances et plus précisément, en ingénierie des connaissances.

2.1. Ontologie de l'e-Learning

La motivation de diffusion des savoirs (connaissances) et leurs acquisitions par des apprenants est centrale pour l'e-learning. Dans ce contexte, les ontologies ont un rôle principal à tenir pour le partage, la dissémination de ses connaissances. Quel que soit le domaine enseigné, une formation repose essentiellement sur trois éléments principaux : les *acteurs* (intervenants), le *domaine d'enseignement* et les *ressources pédagogiques* utilisées pour l'apprentissage. Ces éléments sont modélisés ici comme des sous ontologies de l'ontologie globale de l'e-learning.
Les différents acteurs sur les quels est organisée une formation à distance (e-learning) sont classés en deux catégories principales : les apprenants et les facilitateurs. Dans notre cas, le seul acteur pris en considération est l'apprenant, et son profil (débutant, intermédiaire, expert) est la seule propriété prise en compte.

Le découpage des connaissances du domaine d'enseignement permet de classifier les connaissances d'un domaine à enseigner spécifique suivant un cadre générique réutilisable organisé autour des concepts suivants [Ahmed-Ouamer, 1996] :

- **Didacticiel** : un didacticiel est un logiciel pédagogique dédie, d'aide à l'enseignement et/ou à la formation personnalisée. Il est constitué d'une collection de scenarios et enseigne des concepts ;
- **Concept :** un concept est constitué d'un ensemble d'éléments de connaissance évalués, il peut être lié à d'autres concepts par diverses relations ;
- **Elément de connaissance :** c'est le granule de la matière à enseigner. Il est présenté seul ou combiné avec d'autres éléments de connaissance ;
- **Scénario :** c'est un ensemble d'exposés, d'exercices d'assimilation et de contrôle de connaissances ;
- **Profil initial :** le profil initial est décrit par la liste des concepts que l'élève est supposé a priori posséder ;
- **Profil final** (l'objectif d'enseignement) : est un ensemble de concepts à faire acquérir à l'apprenant.

La figure suivante présente le diagramme de l'ontologie du domaine à enseigner :

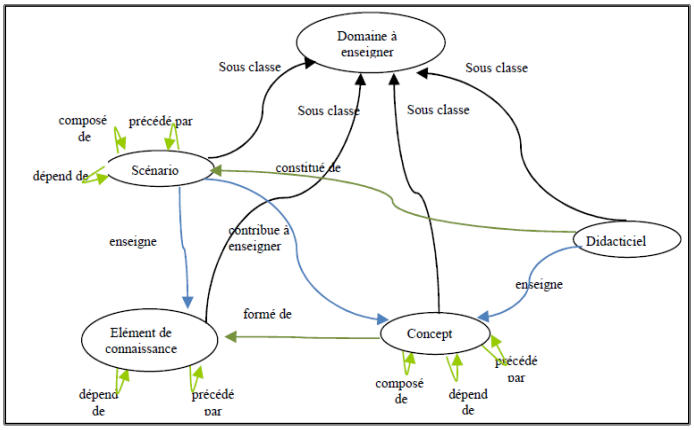

Figure 1 : Ontologie du domaine à enseigner.

2.2. Ontologie de l'e-Learning de l'Informatique

L'ontologie globale de l'e-learning présentée précédemment comporte des éléments communs (invariants) à toute formation à distance : ce sont les *acteurs* et les *ressources pédagogiques*. En revanche la définition de l'ontologie d'un domaine particulier à enseigner est déduite à partir de la définition et du classement des concepts et des relations de ce domaine à enseigner [Cassin et al., 2003], [Hwang, 2003]. Ainsi, l'ontologie de l'e-learning de l'informatique est obtenue par *instanciation* de l'ontologie du domaine à enseigner précédente, et par le classement des connaissances relatives à l'informatique sous forme de : *didacticiels*,

scénarios, *concepts* et *éléments de connaissance*. La figure suivante présente un extrait de l'ontologie du domaine à enseigner dans le cas de l'informatique. Les liens entre concepts de l'ontologie et les profils apprenants ne sont pas décrits ici.

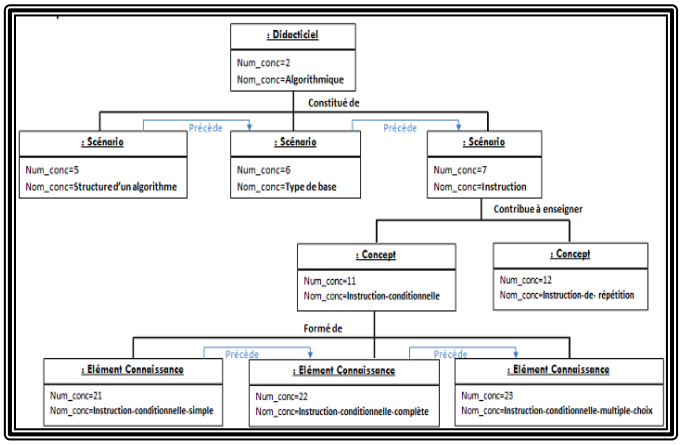

Figure 2 : Extrait de l'ontologie du domaine d'enseignement.

3. L'APPROCHE DE RI SEMANTIQUE DANS LES DOCUMENTS SEMI-STRUCTURES PEDAGOGIQUES

Le modèle logique de représentation des documents que nous utilisons s'appuie sur le modèle DOM (Document Object Model) [Apparao et al., 1998], où un document XML est modélisé par un arbre de nœuds. Les nœuds de cet arbre sont typés (éléments, attributs, texte) qui sont reliés par des relations de structure (parent-fils, ancêtre-descendant).

Nous avons opté pour ce modèle car, il permet la navigation dans la structure en arbre des documents XML, de représenter le contenu et la structure, afin de pouvoir interroger ces documents et récupérer la partie de ces derniers qui réponde le mieux à la requête apprenant. La Figure ci-dessous illustre la modélisation d'un document sous la forme d'un arbre de nœuds.

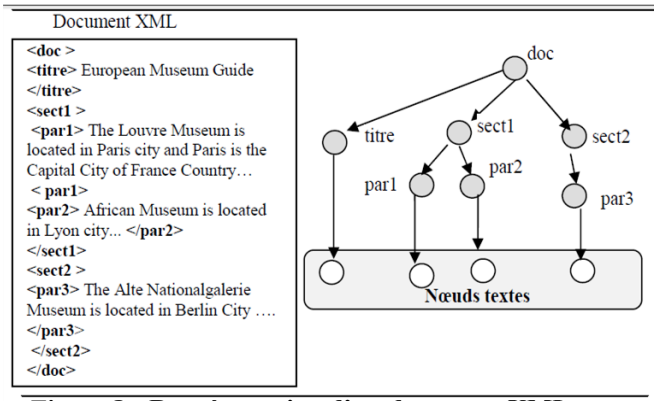

Figure 3 : Représentation d'un document XML sous forme d'arbre.

4. PROCESSUS D'INDEXATION ET D'INTERROGATION

Le processus de la RI sémantique (conceptuelle) est effectué en trois étapes principales:

- **L'identification des concepts:** cette étape consiste à repérer les concepts représentatifs des nœuds textes et des requêtes apprenant, à l'aide d'un analyseur syntaxique
- **La construction de l'index conceptuel d'un nœud**: l'objectif de cette étape est de construire les vecteurs de concepts indexant les nœuds à partir de leurs nœuds descendants.
- **L'appariement nœuds / requête :** cette étape vise à attribuer des scores de pertinence aux nœuds des documents pédagogiques XML vis-à-vis de la requête de l'apprenant.

Dans notre approche, nous proposons d'utiliser le modèle vectoriel sémantique [Woods, 1997], [Berry et al ; 1999], et une ontologie de domaine de l'informatique, afin de construire les vecteurs de concepts représentatifs des nœuds de l'arbre d'un document pédagogique semi-structuré (XML). Le modèle vectoriel sémantique permet de représenter le contenu textuel d'un document ou d'une requête par des vecteurs de concepts sémantique.

Soient Ω l'ontologie de domaine de l'informatique et C_Ω l'ensemble de ses concepts de cardinalité n (n=|C_Ω|). Un espace conceptuel E_Ω sur Ω est l'ensemble C_Ω.

$$E_\Omega=\{c_1,\ldots,c_k,\ldots,c_n\}.$$

Ainsi, dans cet espace conceptuel, un nœud texte $\boldsymbol{N_t^J}$ est représenté par un vecteur de poids des concepts :

$\overrightarrow{\boldsymbol{N_t^J}} = (\boldsymbol{w_{1j}}, \ldots, \boldsymbol{w_{kj}}, \ldots, \boldsymbol{w_{nj}})$ où w_{kj} est le poids du concept c_k dans le nœud texte $\boldsymbol{N_t^J}$, et de la même façon une requête q est représentée dans l'espace d'indexation $C_\Omega$par un vecteur des poids des concepts qui composent la requête :

$\vec{\boldsymbol{q}} = (\boldsymbol{w_1}, \ldots . \boldsymbol{w_k}, \ldots . , \boldsymbol{w_n})$

4.1. Construction des Index de Nœuds

Dans notre approche, la construction des index s'est basée sur l'idée de propagation des concepts, qui nous a été inspirée de [Cui& al, 2003], [Harrathi et al, 2010]. Les documents semi-structurés possèdent une structure arborescente, alors les index des nœuds sont imbriqués les uns dans les autres et par conséquent, l'index d'un nœud de type *élément* contient les index de ses nœuds descendants de type *texte*. Ainsi, les concepts des nœuds de type texte sont donc **propagés** dans l'arbre des documents XML (voir la Figure 4).

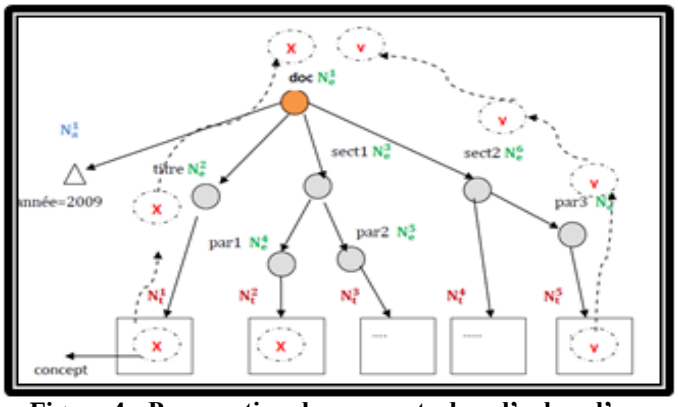

Figure 4 : Propagation des concepts dans l'arbre d'un document XML.

Afin d'identifier la partie du document qui répond le mieux à la requête apprenant, nous proposons, une méthode de propagation des concepts et des poids, en partant des nœuds feuilles (textes) jusqu'à la racine du document.

L'idée de construction des index dans notre approche est basée d'une part sur la distance entre les nœuds, telle que « plus la distance entre un nœud de type texte et son ancêtre est importante, moins il contribue à sa représentation », et d'autre part sur la fréquence d'apparition des concepts telle que « pour un nœud ancêtre, plus un concept apparait souvent dans tous ces nœuds descendants, plus il contribue à sa représentation, même si sa fréquence dans chaque nœud est faible. »
Nous modélisons notre idée par l'utilisation dans la fonction de propagation du paramètre $Dist(N_e, N_t^k)$,qui représente la distance entre le nœud de type élément $\boldsymbol{N_e}$ et de ses nœuds descendants de type texte $\boldsymbol{N_t^k}$dans l'arbre du document, c'est à dire le nombre d'arcs séparant les deux nœuds, et du paramètre $|N_t^{c_j^e}|$,qui représente le nombre de nœuds texte descendants de $\boldsymbol{N_e}$ contenant le concept c_j. Plus le nombre $|N_t^{c_j^e}|$ est grand, plus le concept c_j contribue dans la représentation du nœud N_e.

Comme nous utilisons le modèle vectoriel sémantique pour la représentation interne des index, le vecteur d'un nœud de type élément est construit à partir des vecteurs de ses nœuds descendants de type texte en utilisant l'opérateur somme entre les vecteurs.

Etant donnés un nœud de type élément $\boldsymbol{N_e}$ et un ensemble $\boldsymbol{Ens_{N_e}}$de ses nœuds descendants de type texte : Ens_{N_e}= $\{N_t^1,\ldots,N_t^k,\ldots,N_t^m\}$, le vecteur sémantique représentant le nœud N_e en tenant compte de notre proposition est calculé de la façon suivante :

$$\overrightarrow{\boldsymbol{N_e}} = \sum_{k=1}^{m} \frac{\left|N_t^{c_j^e}\right|}{|N_t|} * \frac{1}{Dist(N_e,N_t^k)} * \overrightarrow{N_t^k} \quad (*)$$

Où $\overrightarrow{\boldsymbol{N_t^k}}$est le vecteur sémantique représentant le k-ième nœud texte descendant du nœud élément N_e, *et* $\frac{1}{dist(N_e,N_t^k)}$ est un paramètre permettant de quantifier l'importance de la distance séparant les nœuds dans la formule de propagation et $|N_t|$ est le nombre de nœud texte descendants de $\boldsymbol{N_e}$. Ainsi, La formule de calcul de poids w_j du concept c_j dans le vecteur $\overrightarrow{\boldsymbol{N_e}}$en tenant compte de notre idée est :

$$w_j = \sum_{k=1}^{m} \frac{\left|N_t^{c_j^e}\right|}{|N_t|} * \frac{1}{Dist(N_e,N_t^k)} \ (*) \text{wkj pour} 1 \le j \le n$$

4.2. *Appariement Nœuds / Requête*

Cette étape consiste à attribuer des scores de pertinences aux éléments d'un document XML (nœud texte ou nœud élément) en comparant la représentation de la requête avec les représentations des nœuds, dans le but de renvoyer les unités d'information les plus *spécifiques* (tous leurs contenus concernent la requête) et les plus *exhaustives* (contiennent les informations requises dans la requête) à l'apprenant. Le calcul du score des nœuds et la pondération des concepts sont des éléments prépondérants dans la phase d'évaluation de la pertinence d'un nœud vis-à-vis d'une requête apprenant.

4.2.1. *Pondération des Concepts*

Dans la recherche de documents semi-structurés (documents XML), le poids d'un terme exprime son importance de manière locale au sein du document ou de l'élément et de manière globale au sein de la collection. Le poids d'un terme est généralement évalué selon trois dimensions :

- La fréquence d'un terme dans le nœud texte (TF);
- La fréquence inverse de document pour le terme (IDF) ;
- La fréquence inverse de l'élément pour le terme (IEF).

Une étude sur la pondération des termes [Sauvagnat et al., 2006], a montré que la combinaison de TF et IEF donne la meilleure performance. Ainsi, nous adoptons ces mesures pour calculer les pondérations des concepts. Ainsi, dans notre approche le poids d'un concept $c_j$dans un nœud texte N_t^i(dénoté par Wij) est exprimé par la formule suivante:

$$\boldsymbol{w_{ij} = cf_j^i * iecf_j}$$

Avec:

- $\boldsymbol{cf_j^i}$= *nombre d'occurence du cocept* c_j *dans un* nœud *texte* $\boldsymbol{N_t^i}$.
- $\boldsymbol{iecf_j = log \frac{|N_t|}{|N_t^{c_j}|}}$

Où $|N_t|$ est le nombre total de nœuds textes de la collection et $\left|\boldsymbol{N_t^{c_j}}\right|$est lenombre total de nœuds textes contenant le conceptc_j.

4.2.2. *Calcul des Scores des Nœuds*

Dans la recherche des documents semi-structurés, un nœud texte ou élément est considéré très pertinent s'il est très *exhaustif* et très *spécifique*.

L'appariement nœud/requête vise à attribuer des scores de pertinence aux éléments d'un document (les nœuds de type texte et les nœuds de type élément dans l'arbre XML).

❖ **Nœud de type texte**

Un nœud de type texte est représenté par un vecteur de poids des concepts :

$$\overrightarrow{\boldsymbol{N_t^j}} = \left(\boldsymbol{w_{1j}, \ldots, w_{kj}, \ldots, w_{nj}}\right).$$

Où w_{kj} est le poids du concept c_k dans le nœud N^j. De la même manière, une requête q est représentée par un vecteur de poids des concepts.

$$\boldsymbol{\overrightarrow{q} = (w_1, \ldots . w_k, \ldots , w_n).}$$

Le vecteur des poids des concepts de la requête est calculé selon le type du concept recherché, par exemple, si le concept recherché par l'apprenant est de type *didacticiel,* alors le système cherche tous les concepts concernant ce didacticiel, grâce à l'exploitation des relations sémantiques existantes entre les concepts de notre ontologie « **constitué de, enseigne…** ».

Par exemple pour un espace conceptuel d'indexation **E** formé de 8 concepts suivants:
E= {Algorithme, BDD, instruction, entité, association, UML, Déclaration, structure_d'un_algorithme}, et une requête q = 'Algorithme', le vecteur sémantique représentant la requête sera comme suit : $\vec{q} = (1,0,1,0,0,0,1,1)$**,** car le concept recherché par l'apprenant est de type didacticiel, et les concepts instruction, déclaration et structure_d'un_algorithme font partir de ce concept.

Généralement, la mesure de la proximité entre document et requête est mesurée grâce à la mesure de cosinus [Salton et al., 1983].

Ainsi, Le score de pertinence d'un nœud N_t^j vis-à-vis une requête q est obtenue en utilisant la mesure de cosinus comme suit :

$$score(q, N_t^j) = \frac{\sum_{i=1}^{n} w_{ij} * w_i}{\left[\sum_1^n w_{ij}^2\right]^{\frac{1}{2}} * \left[\sum_1^n w_i^2\right]^{\frac{1}{2}}}$$

❖ **Nœud de type élément**

Puisque l'information textuelle est située dans les nœuds texte, l'évaluation de l'exhaustivité et la spécificité d'un nœud élément consiste à répondre à la question suivante : à quel point les descendants (nœuds textes) du nœud contiennent-ils et concernent-ils des informations demandées par la requête ?

Pour répondre à cette question, nous proposons d'introduire le nombre des nœuds descendants qui sont pertinents $|N_t^P|$, et le nombre de descendants qui ne sont pas pertinents $|N_t^{NP}|$. Le calcul de la valeur de pertinence d'un nœud se base sur deux intuitions :

- Si $|N_t^P|$est très grand, alors la probabilité que le nœud contient des informations demandées par la requête est très grande.

- Si $|N_t^{NP}|$est très petit, alors la probabilité que le nœud concerne la requête est très grande.

La valeur de pertinence d'un nœud de type élément est alors calculée selon la formule suivante :

$$\textbf{Pertinence } (q, N_e) = |N_t^P| * \frac{|N_t^P|}{|N_t^P| + |N_t^{NP}|} * \textbf{score } (q, N_e).$$

Où :

- $|N_t^P|$est l'ensemble des nœuds texte descendants de N_e qui sont pertinents (ayant un score non nul).
- $|N_t^{NP}|$ est l'ensemble des nœuds texte descendants de Ne qui ne sont pas pertinents (ayant un score nul).
- score (q, N_e) est le score de pertinence calculée par la formule précédente.

5. INTERROGATION

L'interrogation permet à l'apprenant de fouiller dans la base de documents, et cela par l'introduction d'une requête via un moteur de recherche. Dans notre approche un nœud texte et une requête sont présentés par des vecteurs de poids de concept en utilisant le modèle vectoriel sémantique. Le moteur de recherche ordonne les documents (fragments de document) en fonction de leur ressemblance avec la requête.

Pour interroger un document semi-structuré, il est nécessaire de concevoir des modèles d'indexation permettant d'accéder rapidement à un document en spécifiant des conditions sur son contenu textuel et sur sa structure. Le modèle d'indexation doit identifier les relations structurelles dans un document XML (ancêtre descendant, suivant-précèdent).

Dans notre approche, nous nous appuyons sur le modèle DOM. Le modèle DOM permet de modéliser la structure d'un document XML par un arbre de nœuds. Ces nœuds sont typés et sont reliés par des relations de structure (parent-fils, ancêtre descendant). Le type d'un nœud peut être un élément, un attribut, un texte.

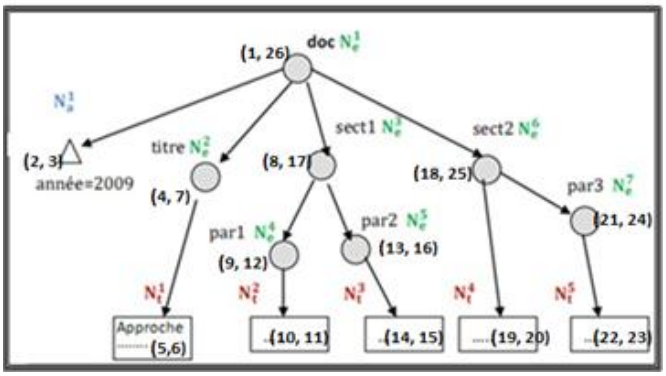

Figure 5 : Valeurs de début et fin assignées aux nœuds d'un document XML.

La numérotation d'un arbre consiste à utiliser deux identificateurs pour un nœud d'arbre XML : début et fin [Harrathi et al., 2007]. Les valeurs de début et fin sont assignées aux nœuds comme suit:

- début : l'ordre d'apparition d'un nœud dans la lecture séquentielle du document XML.
- fin : l'ordre de disparition d'un nœud dans la lecture séquentielle du document XML.

Pour naviguer aisément dans l'arbre, permettre l'accès rapide à un nœud, et déterminer rapidement les relations ancêtres-descendants, l'approche proposée consiste à définir un nœud de la structure par le *n-uplet* suivant:

<début, fin, parent, type, nom, valeur >

Où :

- ***début :*** le premier identificateur unique du nœud qui représente l'ordre d'apparition du nœud dans la lecture séquentielle du document XML.
- ***fin*** : le deuxième identificateur unique du nœud qui représente l'ordre de disparition du nœud dans la lecture séquentielle du document XML.
- ***Parent*** : l'identificateur (la valeur de début) du nœud parent
- ***Type*** : le type du nœud (élément, attribut, texte)

- ***Nom*** : nom du nœud (nom de la balise ou de l'attribut, dans le cas d'un nœud de type texte le nom vaut nul)
- ***Valeur*** : la valeur du nœud (la valeur de l'attribut ou le contenu textuel, dans le cas d'un nœud de type élément la valeur vaut nul).

Si l'on transpose tous les nœuds dans un espace à deux dimensions basé sur les coordonnées de *début* et *fin*, on peut exploiter les propriétés suivantes illustrées par l'exemple de la figure **5,** Etant donné un certain nœud *n* (le nœud *doc/sect1* dans l'exemple ci-dessous):

- Tous les ancêtres de *n* sont au-dessus à gauche de la position de *n* dans le plan.
- Tous ses descendants sont en dessous à droite.
- Tous les nœuds le précédent dans la lecture séquentielle du document sont en dessous à gauche.
- La partition du plan au-dessus à droite comprend tous les nœuds successeurs dans la lecture séquentielle du document.

Le modèle d'indexation proposé permet d'identifier de façon unique chaque nœud de l'arbre d'un document par un intervalle [début, fin]. Ainsi, la relation de structure entre deux nœuds *u* et *v* est résolue aisément comme suit :

- **ancêtre-descendant**, u est un ancêtre (descendant) de v si seulement si l'intervalle de u contient (est inclus dans) l'intervalle de v.
- **prédécesseur-successeur,** u précède (succède) v si seulement si l'intervalle de u précède (succède) l'intervalle de v.

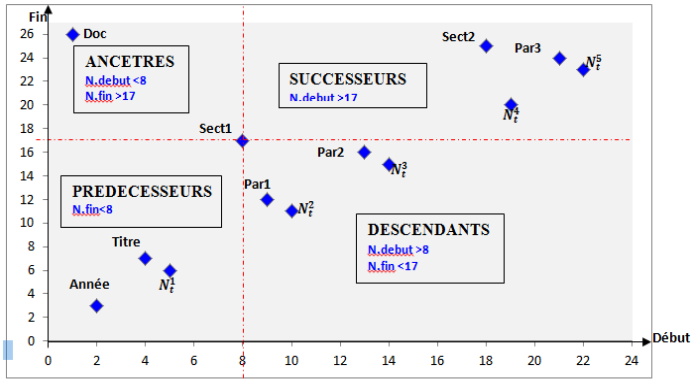

Figure 6: Représentation du document doc.xml dans un espace deux dimensions basé sur les coordonnées de début et fin.

Pour extraire les descripteurs **(début, fin, parent, type, nom, valeur)** décrivant un nœud de l'arbre d'un document XML, il est nécessaire d'accéder à un document par le biais d'un «parseur». Un parseur est une application dont le rôle est de convertir un flux de balisage en une structure de sortie accessible par un programme. Il existe deux types deparseurs:

- Parseurs orientés événements: API[1].**SAX (**Simple API for XML).
- Parseur de type arbre : API.**DOM** (*Document Object Model*).

Afin d'évaluer l'extraction des descripteurs des nœuds, nous proposons d'utiliser une approche orientée base de données relationnelle pour l'implémentation. Les tables définies pour le stockage des données sont décrites comme suit :

- ***Document (Idf_doc,*** *nom_doc)*
- ***Elément*** *(**idf doc, début,** nom_ele, fin, parent)*
- ***Attribut*** *(**idf doc, début**, fin, parent, nom_att, valeur_att)*
- ***Texte*** *(**idf doc, début,** valeur, fin, parent)*

Ainsi, nous proposons d'utiliser les méthodes du parseur SAX pour extraire les descripteurs d'un nœud. Les principales méthodes utilisées sont:

- **Stardocument :** méthode invoquée à chaque fois qu'un nouveau document est rencontré.
- **StartElement:** méthode invoquée à chaque fois qu'une balise ouvrante est rencontrée.
- **EndElement:** méthode invoquée à chaque fois qu'une balise fermante est rencontrée.
- **Characters**: méthode invoquée à chaque fois qu'un texte est rencontré.

Pour implémenter ces méthodes, nous avons utilisé une pile pour mémoriser les valeurs de débuts des nœuds dont la valeur de fin n'est pas calculée. Cette pile est manipulée selon la méthode invoquée comme suit :

- Début d'un élément (la méthode startElement) :
  - Empiler la valeur de début au sommet de la pile.
- Fin d'un élément (la méthode endElement) :
  - Calculer la valeur de début de l'élément (sommet de la pile) ;
  - Dépiler le sommet de la pile ;
  - Calculer la valeur du parent de l'élément (sommet de la pile).

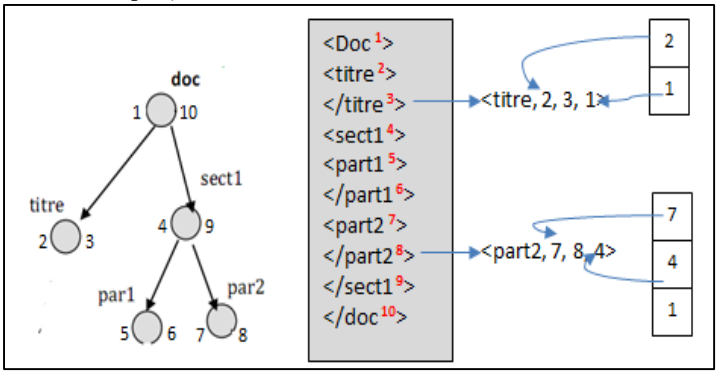

Figure 7 : Exemple d'extraction des descripteurs d'un nœud.

L'algorithme proposé permet aussi de calculer d'autres attributs d'un nœud comme sa hauteur (profondeur de la pile - 1) et son chemin (les éléments de la pile).

[1]Application Programming Interface

5.1. Fonctionnement du Moteur de Recherche

L'interrogation dans notre approche concerne une recherche sur les ressources de type cours, ainsi le fonctionnement du moteur de repose sur l'exploitation des relations existantes entre les concepts de l'ontologie du domaine à enseigner, et du profil de l'apprenant, ce dernier est déterminant pour le résultat d'une recherche, tel qu'une même requête provenant de deux apprenants ayant un niveau différent aura deux réponses différentes c.-à-d. les documents retournés ne seront pas identiques.

Par exemple, si le concept recherché par l'apprenant est de type :

- « **Didacticiel** » alors le système retourne à l'apprenant tous les scénarios qui lui appartiennent et cela grâce à l'exploitation de la relation « **Composé de** ».

- « **Scénario** » alors le système retourne à l'apprenant en plus de(s) scénario(s) concerné(s) d'autres scénarios, et cela selon le profil de l'apprenant, tels que si le profil de l'apprenant est :
  - « **Expert** », alors le système retourne en plus le(s) scénario(s) suivant(s), et cela par l'exploitation de la relation inverse de la relation « **précédé par** » existante entre les scénarios.
  - « **Débutant** », alors le système retourne en plus le(s) scénario(s) précédent (s), et cela par l'exploitation de la relation « **précédé par** » existante entre les scénarios.
  - « **Intermédiaire** », alors le système retourne en plus le(s) scénarios suivant(s), et précédent(s).

- « **Concept** » alors retourner tous les scénarios qui contribuent à enseigner ce concept, et cela grâce à la relation « **contribue à enseigner** » existante entre le scénario et le concept, et d'autres scénarios, et cela selon le profil de l'apprenant, tels que si le profil de l'apprenant est :
  - « **Expert** », alors le système retourne en plus les scénarios qui contribuent à enseigner le(s) concept(s) suivant(s) du concept concerné et cela par l'exploitation de la relation inverse de la relation « **précédé par** » existante entre les concepts eux-mêmes.
  - « **Débutant** », alors le système retourne en plus le(s) scénario(s) précédent(s), et cela par l'exploitation de la relation « **précédé par** » existante entre les scénarios.
  - « **Intermédiaire** », alors le système retourne en plus le(s) scénarios suivant(s), et précédent(s).

6. IMPLEMENTATION

Nous avons implémenté le système sous forme d'une application Web. Pour cela nous avons utilisé le langage Java et les Servlets qui permettent une grande flexibilité et la portabilité de l'application. Celle-ci rentre dans le cadre de la nouvelle génération du Web (le Web sémantique). En effet nous avons utilisé le langage OWL pour représenter l'ontologie de domaine informatique et l'API Jena pour sa manipulation. La figure 7 présente l'architecture générale du prototype développé.

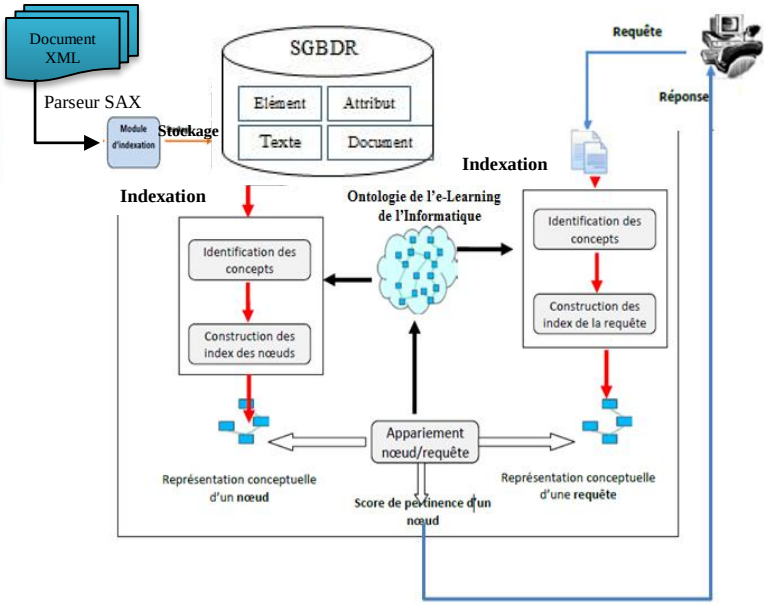

Figure 8 : Architecture générale du prototype développé.

Cette application est destinée aux apprenants du domaine de l'informatique (algorithmique et base de données). Les apprenants effectuent des recherches sur la collection indexée, et cela selon leurs profils comme illustré dans les figures 9 et 10.

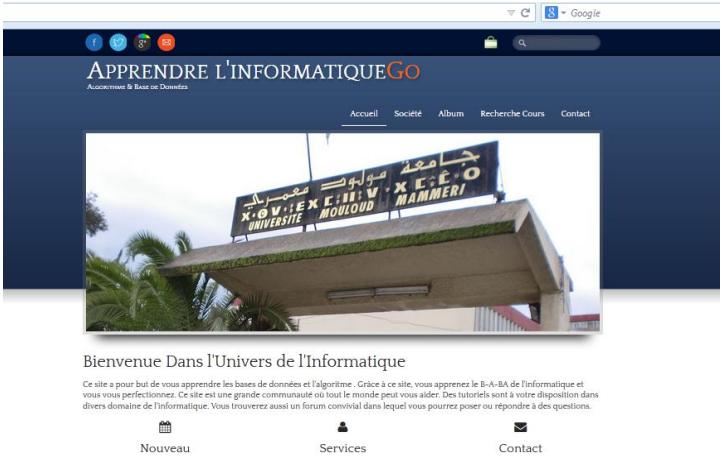

Figure 9 : Interface principale de notre système.

Figure 10 : Interface de recherche des cours.

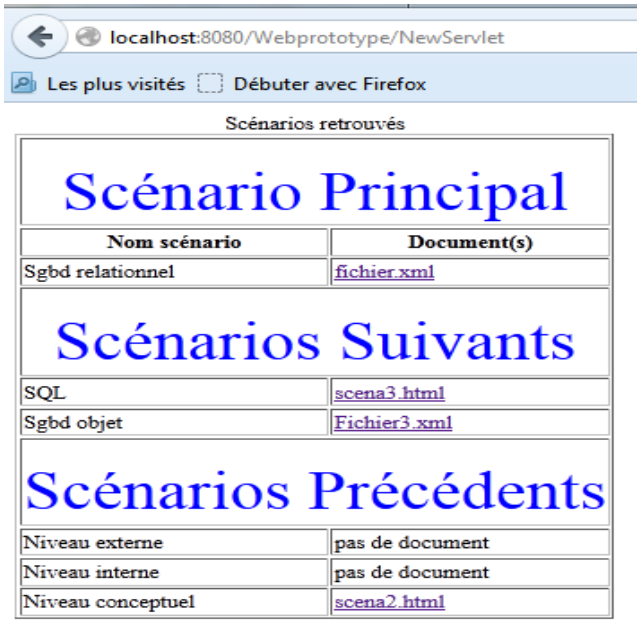

Figure 11 : Interface des documents retournés selon la requête et le profil apprenant.

7. CONCLUSION

L'approche présentée dans cet article apporte une certaine intelligence dans le processus de RI sur des documents pédagogiques semi structurés (domaine informatique). Cela passe notamment par l'emploi d'une ontologie de domaine à la collection de documents.

Nous avons proposé une représentation des nœuds de l'arbre d'un document et de la requête par des vecteurs sémantiques de concepts. La pondération des concepts est évaluée selon deux dimensions : la fréquence d'un concept dans un nœud et la fréquence inverse d'élément pour le concept. Les tests effectués montrent la faisabilité de l'approche proposée.

8. BIBLIOGRAPHIE

[Ahmed-Ouamer, 1996] Ahmed-Ouamer, R., « Développement de systèmes d'EIAO dans AGEDI », *Séminaire national d'informatique SNITO 96*, Tizi-Ouzou, 1996, p. 53-79.

[Ahmed-Ouamer et al., 2010] Ahmed-Ouamer, R., et Hammache, A. : Ontology-Based Information Retrieval for e-Learning of Computer Science. The First International Conference on Machine and Web Intelligence IEEE ICMWI 2010., October 2010, IEEE, **ISBN: 978-1-4244-8610-6**, p. 229-236.

[Apparao et al., 1998] Apparao,V., Byrne, S., Champion,M., Isaacs, S., Jacobs, I., Le Hors,A., Nicol,G., Robie,J., Sutor,R., Wilson,C., Wood,L. . Document Object Model (DOM). W3C recommendation, Technical Report REC DOM-Level-1-19981001, (1998).

[Berry et al., 1999] Berry, M. W., Z. Drmac, et E. R. Jessup : . Matrices, vector spaces, and information retrieval. SIAM Rev. 41(2), 335–362(1999).

[Cassin et al., 2003] Cassin, Eliot, C., Lesser, V., Rawlins, K., Woolf, B., « Ontology extraction for educational knowledge bases », Van Elst, P.L., Dignum, V., Abecker, A. (eds.), *AMKM* 2003, LNAI 2926, Springer-Verlag Berlin Heidelberg, 2003, p. 297–309.

[Cui& al, 2003] H.cui, J-R.Wen, J-R.Chua, "Hierarchical indexing and flexible element retrieval for structured document", april 2003

[Dietz, 1982] Dietz, PaulF. Maintaining order in a linked list. In Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, pages 122–127, San Francisco, California, May 1982.

[Harrathi et al., 2007] Harrathi R., Calabretto S. Un modèle pour l'interrogation visuelle des documents structurés. Actes de la Conférence CORIA'2007, Saint-Etienne, 28-30 mars 2007. pp. 291-302. ISBN 978-2_86272-452-2 2007.

[Harrathi et al, 2010] Harrathi R., CalabrettoS. Une approche de recherche sémantique dans les documents semi- structurés. Dans RISE (Recherche d'Information Sémantique) dans le cadre de la conférence INFORSID'2010, Marseille 2010.

[Harrathi, 2010] Harrathi R, " Recherche d'information conceptuelle dans les documents semi-structurés", *Lyon 2010.*

[Hernandez 05] Hernandez N., Ontologies de Domaine pour la Modélisation du Contexte en Recherche d'Information. Thèse de Doctorat en Informatique de l'Université Paul Sabatier de Toulouse (Sciences). Spécialité Informatique. Décembre 2005.

[Hwang, 2003] Hwang, G.J., « A conceptual map model for developing intelligent tutoringsystems », *Computers & Education*, 40, 2003, p. 217–235.

[Roche 05]. Roche. C. Terminologie et ontologie. Larousse, 2005.

[Salton & al., 1983] Salton G., Fox E. A., Wu H. Extended Boolean information retrieval system. CACM 26(11), pp. 1022-1036, 1983.

[Sauvagnat et al., 2006] Sauvagnat k, Boughanem.M.,. Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée. Dans : Actes de CORIA 2006, Lyon, 15-17 mars (2006).

[Woods, 1997] Woods, W. A. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, Mountain View,CA, April. www.sun.com/research/techrep/1997/abstrac-61.html.