Faculté de Technologie *Département d'Informatique*

Maintenance des applications orientées web par des ontologies de domaine : La rétro-ingénierie à base d'indexation sémantique.

THÈSE

Présentée et soutenue publiquement le : …..……..……

Pour l'obtention du

Doctorat en Sciences de l'Université de Djillali Liabes - Sidi Bel Abbes

(Spécialité Informatique)

Par

Abdeslem DENNAI

Composition du jury

Président:

Mimoun MALKI, *Professeur, Université de Sidi Bel Abbes, ALGERIE.*

Examinateurs:

Djelloul BOUCHIHA, *Maitre de conférences, Université de Nâama, ALGERIE.*

Khelifa BENAHMED, *Maitre de conférences, Université de Béchar, ALGERIE.*

Directeur:

Sidi Mohamed BENSLIMANE, *Professeur, Université de Sidi Bel Abbes, ALGERIE.*

Invité:

Amar Djamel BENSABER, *Maitre de conférences, Université de Sidi Bel Abbes, ALGERIE.*

A mes chers parents,
A ma chère femme,
A mes chers enfants : Hiba, Malak, Soufiane et Imad,
Et à toute la famille.

Remerciements

Je rends grâce, en premier lieu, à dieu, le tout puissant, pour m'avoir donné le courage et la patience afin de pouvoir accomplir ce travail.

Je tiens à témoigner mes très sincères reconnaissances à mon directeur de thèse Mr. Sidi Mohammed BENSLIMANE Professeur à l'université Djillali Liabes de SIDI BEL ABBES qui a dirigé cette thèse d'une main de maitre. Tout au long de ces six années, il a su orienter mes recherches aux bons moments, toujours dans les bonnes directions. Malgré l'éloignement, il a toujours été disponible pour prodiguer des conseils et des orientations pertinentes. Pour tout cela, pour m'avoir offert la chance d'en être là aujourd'hui, pour sa confiance et pour sa précieuse aide technique je le remercie du fond du cœur.

Mes plus vifs remerciements vont également à Mr. Mimoun MALKI professeur à l'université Djillali Liabes de SIDI BEL ABBES qui a accepté de chapeauter ce travail dans sa première année qu'il trouve ici l'expression de ma plus profonde reconnaissance.

J'aimerais exprimer ma reconnaissance envers les rapporteurs de cette thèse pour la lecture et la correction de mon manuscrit et pour l'intérêt qu'ils ont porté à mon travail. Je remercie également les autres membres du jury qui ont accepté de juger ce travail.

Enfin, Je remercie toute ma famille et en particulier mes parents et ma femme qui m'ont aidé d'une manière directe ou indirecte tout au long de mon travail.

Résumé

Les applications orientées web sont devenues les moyens de communication les plus importants pour les entreprises commerciales de toutes sortes. Cependant, la plupart de ces applications sont construites dans l'urgence. Pour écourter les délais de développement, la phase de conceptualisation est souvent sacrifiée et la documentation associée est négligée. En outre, en phase d'exploitation, ces applications sont modifiées au fil des besoins. Elles subissent diverses dégradations touchant aussi bien leur contenu informatif que leur structure de navigation.

L'objectif de ce travail est de proposer un processus de rétro-ingénierie des applications orientées Web à base d'une approche d'indexation sémantique. Cette approche qui, elle même, est à la base d'une ontologie de domaine avec l'utilisation de l'étiqueteur TreeTagger et le dictionnaire sémantique WordNet. Le processus que nous allons proposer, passe par quatre phases : (i). Modélisation : Cette phase permet d'avoir un ensemble de concepts représentants des informations non redondantes, extraites à partir des pages HTML ou des documents XML, (ii). Attachement sémantique : Le résultat de la phase précédente représente un élément candidat pour cette phase courante où on exploite l'ontologie de domaine pour la validation de ces concepts par application de la distance sémantique et en utilisant l'étiqueteur TreeTagger et WordNet. Un index initial est généré à la fin de cette phase, (iii). Enrichissement : Le contenu de l'index s'accroit par d'autres concepts de cette même ontologie en utilisant l'une des mesures de similarité sémantique entre les concepts de l'ontologie (Mesure de Wu et Palmer), (iv). Re-conceptualisation : L'index final, résultat de la phase précédente, sera transformé en dictionnaire de données en gardant que les concepts importants non redondants et en les complétant par d'autres informations telles que la définition du typage et la précision du format. Ainsi, une nouvelle conception du système ou de l'application peut commencer.

Mots clés : Ontologie de domaine, indexation sémantique, application web, document web, XML, HTML, rétro-ingénierie, distance sémantique et similarité sémantique.

Absract

Web-oriented applications have become the most important means of communication for business enterprises of all kinds. However, most of these applications are built in a hurry. To shorten development time, conceptualization phase is often sacrificed and associated documentation is neglected. In addition, during operation, these applications are modified over the needs. They undergo various degradations affecting both their information content navigation structure.

The objective of this work is to propose a process of reverse engineering web-oriented applications based on semantic indexing approach. Similarly, this approach based on domain ontology with the use of TreeTagger and semantic dictionary WordNet. The process we are going to propose four phases: (i). Modeling: This phase allows to have a set of concepts representatives non-redundant information, extracted from HTML pages or XML documents, (ii). Attachment semantics: The result of the previous phase is a candidate element for this current phase where it exploits the domain ontology to validate these concepts by applying the semantic distance and by using TreeTagger with WordNet. An initial index is generated at the end of this phase, (iii). Enrichment: The contents of the index increases by other concepts of the same ontology using one of semantic similarity measures between ontology concepts (Wu and Palmer measure), (iv). Reconceptualization: The final index, as a result of the previous phase, will be transformed into data dictionary keeping the important and no redundant concepts and supplementing them with other information such as the definition of typing and accuracy of the format. So, a new system design or application can begin.

Keywords: Domain ontology, semantic indexing, web application, web document, XML, HTML, reverse engineering, semantic distance, and semantic similarity.

ملخص

أصبحت تطبيقات شبكة الإنترنت أهم وسيلة اتصال بالنسبة للمؤسسات التجارية بجميع أنواعها. مع ذلك، فإن معظم هذه التطبيقات يتم تنفيذها في استعجال. لربح الوقت في إنتاج وتطوير هذه التطبيقات، غالبا ما يتم التضحية بمرحلة التصميم وإهمال إنجاز الوثائق المرفقة بها. بالإضافة إلى ذلك، وخلال عملية استغلالها، يتم تعديل هذه التطبيقات على حسب الاحتياجات، كما تتعرض إلى التدهور والذي يؤثر على كل من المعلومات المحتواة و هيكلتها.

الهدف من هذا العمل هو إقتراح هندسة عكسية لتطبيقات شبكة الإنترنت على أساس نهج الفهرسة الدلالي. وبالمثل، فإن هذا النهج يعتمد على أنطولوجيا المجالات مع استخدام TreeTagger والقاموس الدلالي WordNet. تتطلب العملية المقترحة أربع مراحل: (i). النمذجة: وهذه المرحلة تسمح بالحصول على مجموعة من المعطيات كلها معلومات غير مكررة، المستخرجة من صفحات HTML أو وثائق XML. (ii). الربط الدلالي: نتيجة المرحلة السابقة تعتبر عنصر مرشح لهذه المرحلة الحالية حيث يستغل الأنطولوجيا للتصديق على صحة هذه المعطيات من خلال تطبيق مفهوم المسافة الدلالي و باستعمال TreeTagger و WordNet. يتم إنشاء فهرس أولي في نهاية هذه المرحلة. (iii). إثراء: محتوى الفهرس يزيد بمعطيات أخرى من نفس الأنطولوجيا باستخدام أحد مقاييس التشابه الدلالي بين معطيات الأنطولوجيا (مقياس Wu-Palmer). (iv). إعادة التصميم: الفهرس النهائي، نتيجة المرحلة السابقة، يتم تحويله إلى قاموس للبيانات بالحفاظ فقط على المعطيات الهامة و الغير مكررة واستكمالها بمعلومات أخرى مثل تعريف النوعية وتدقيق الشكل. عندها يمكن تصميم نظام أو تطبيق جديد.

كلمات البحث: أنطولوجيا المجال، فهرسة الدلالات، تطبيقات الويب، وثيقة الويب، XML، HTML، الهندسة العكسية، مقياس المسافة الدلالي والتشابه الدلالي.

Table des Matières

Chapitre A.II. Ontologies et Mesures de Similarité

Chapitre A.III. Indexation Sémantique et Rétro-Ingénierie : Revue de Littérature

Partie B. De l'Ontologie vers l'Indexation Sémantique pour une Rétro-Ingénierie

Chapitre B.I. Indexation Sémantique et Rétro-Ingénierie : État de l'Art

Chapitre B.II. Rétro-Ingénierie des Applications Web à Base d'Indexation Sémantique : Conception et Implémentation

Introduction Générale

Sommaire

- **Contexte général**

La maintenance des applications orientées web a été considérée comme un sujet de recherche fortement préconisé dans le domaine de l'ingénierie des connaissances du web. Ce domaine qui fusionne d'autres domaines à savoir : La recherche d'informations (Des connaissances) dans le web, l'extraction, l'acquisition et la représentation de ces informations.

Les applications orientées web sont devenues les moyens de communication les plus importants pour les entreprises commerciales de toutes sortes. Elles fournissent les principaux moteurs qui améliorent non seulement l'image de marque de l'entreprise, mais agissent également en tant que ressources utiles pour augmenter le part de marché global d'une compagnie.

La majorité du contenu du web actuellement produit est conçu pour être lu par des êtres humains, et non pas pour être manipulé symboliquement par des programmes informatiques [Berners-Lee et al., 2001]. Certes les documents web, qui sont considérés comme des composants d'une application web, tels que les pages HTML ou les documents XML sont manipulés par un programme pour que la mise en page soit correcte, mais ce traitement se limite à interpréter les balises de présentation HTML ou XML présentes dans le document. Ces balises se limitent à décrire la manière dont le document doit être présenté. La signification du contenu du document reste implicite et le document ne peut donc pas être manipulé sur la base de cette signification. De manière générale les ordinateurs n'ont aucune méthode systématique pour traiter le contenu d'un document web sur la base de leur sémantique. La capacité à manipuler le contenu des documents web sur la base de leur sémantique permettrait à des programmes de réaliser des tâches qui doivent être actuellement réalisées à la main et ouvre la voie à de nouvelles possibilités d'automatisation et par la suite à des processus de rétro-ingénierie et de réingénierie.

- **Problématique**

La plupart des applications orientées Web sont construites dans l'urgence. Pour écourter les délais de développement, la phase de conceptualisation est souvent sacrifiée et la documentation associée est négligée. De plus, en phase d'exploitation, les applications orientées Web sont modifiées au fil des besoins. Elles subissent diverses dégradations, touchant aussi bien leur contenu informatif que leur structure de navigation. La nature hétérogène et dynamique des composants constituant une application orientée Web, le manque des mécanismes de programmation efficaces pour la production de ces applications, le développement très rapide de ces applications par des procédés qui ne respectent pas souvent les démarches traditionnelles de développement des systèmes d'information rendent la maintenance et l'évolution de ces applications une tâche complexe et couteuse.

Dans la pratique, la plupart des schémas conceptuels des systèmes d'information et bases de données sont développés essentiellement à partir de zéro. Cependant, au cours de cette dernière décennie, plusieurs approches ont vu le jour, ayant comme objectif la maintenance des applications orientées Web à base de processus de rétro-ingénierie [Tramontana, 2005 ; Vanderdonckt et al., 2001 ; Estiévenart et al., 2003 ; Gaeremynck et al., 2003 ; Di Lucca et al., 2002 ; Bellettini et al., 2004].

D'autre part, plusieurs chercheurs [Gómez-Pérez et Rojas-Amaya Ma, 1999 ; Fürst, 2004 ; Gandon, 2002] ont pu démontrer que le concept d'ontologie permet d'analyser le savoir dans un domaine en modélisant les concepts pertinents pour une ou plusieurs applications de ce domaine. Récemment, plusieurs approches tentent d'utiliser les ontologies comme source sémantique permettant la dérivation de schéma conceptuel [Peterson et al., 1998 ; Swartout et al., 1996 ; Gibson et Conheeney, 1995].

Cependant, la plus part de ces approches supposent l'existence d'information utile à cette extraction. De plus, si l'ontologie de domaine utilisée est assez large, le schéma conceptuel dérivé peut inclure plusieurs concepts et relations superflus [Conesa et Olivé, 2004 ; El-Ghalayini et al., 2006 ; Vasilecas et Bugaite, 2007].

- **Contributions**

Pour répondre à cette problématique, nous proposons un processus de rétro-ingénierie des applications orientées Web à base d'une approche d'indexation sémantique. Cette approche qui, elle même, est à la base d'une ontologie de domaine. Notre processus proposé, qui s'exécute en parallèle du déroulement de notre approche d'indexation sémantique, passe par quatre phases :

(i). Modélisation : Cette phase permet d'extraire les informations utiles à partir des pages HTML incluant les tableaux, les listes et les formulaires ou à partir de documents XML (eXtensible Markup Language) en se basant sur ses balises, (ii). Attachement sémantique : Les informations, résultat de la phase précédente, représentent les éléments candidats pour cette phase courante où on exploite l'ontologie de domaine comme source sémantique pour la validation des concepts cachés dans des pages HTML (HyperText Markup Language) ou dans des documents XML par l'application de la distance sémantique ou par l'utilisation de l'outil WordNet. Un index initial est généré à la fin de cette phase, (iii). Enrichissement : On accroit le contenu de l'index par d'autres concepts de l'ontologie de domaine en utilisant l'une des mesures de similarité sémantique (Dans notre approche la mesure de Wu et Palmer) entre ces concepts de l'ontologie. En outre, on peut enrichir en plus l'index, d'une part, par d'autres termes résultats du TreeTagger en étiquetant que les termes fréquents[1] des deux documents web et d'autre part par d'autres termes résultats de WordNet et qui sont sémantiquement liés aux termes fréquents dans les deux documents web. (iv). Re-conceptualisation : L'index final, résultat de la phase précédente, sera transformé en dictionnaire de données en gardant que les concepts importants, non redondants et en les complétant par d'autres informations telles que la définition du typage et la précision du format. Ainsi, une nouvelle conception du système ou de l'application peut commencer.

Autres principales contributions de cette thèse sont les suivantes :

- Un premier état de l'art détaillé présentant les différentes approches et techniques de calcul de mesure de similarité et en particulier la similarité sémantique. Une classification avec une synthèse, nous a permis de se décider sur la mesure de Wu et Palmer qui une mesure basée sur les arcs et simple à manipuler (cf. section A.II.2.).
- Un deuxième état de l'art survolant les différentes recherches effectuées dans le domaine de l'indexation sémantique avec un tableau récapitulatif, nous a permis de proposer une approche en prenant en considération le contexte général de notre recherche (cf. section B.I.1.).
- Un troisième état de l'art dévoilant le principe et les spécifications de certains travaux réalisés dans le domaine de la rétro-ingénierie et de la réingénierie avec un tableau récapitulatif, nous a permis de présenter notre processus de rétro-ingénierie basé sur l'approche d'indexation sémantique (cf. section B.I.2.).

[1] Sont détectés en appliquant les formules de calcul des fréquences d'un terme dans un document.

- **Communications et publications dans le cadre de la thèse**

- **Communications**

1- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Vers une mise à jour de la mesure de Wu et Palmer pour l'amélioration du calcul de la distance sémantique", 1st International Conference on Information Systems and Technologies ICIST'2011, Tébessa, Algérie, 24- 25 et 26 Avril 2011.

2- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Nouvelle version d'une mesure de similarité pour un meilleur calcul de la distance sémantique entre concepts d'une ontologie", 7ème Colloque sur l'Optimisation et les Systèmes d'Information COSI'2011, Guelma, Algérie, 24-28 Avril 2011.

3- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Extraction d'informations à partir de pages HTML ou de documents XML par une indexation sémantique en utilisant une ontologie de domaine", 2nd International Conference on Information Systems and Technologies ICIST'2012, Sousse, Tunisie, 24- 25 et 26 Mars 2012.

4- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Information extraction from HTML pages or XML documents by a semantic indexing, using domain ontology", 3rd International Conference on Multimedia Computing and Systems ICMCS'2012, IEEE conference, Tangier, Morocco, 10- 12 Mai 2012.

5- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Toward an Update of a Similarity Measurement for a Better Calculation of the Semantic Distance between Ontology Concepts", 2nd International Conference on Informatics Engineering & Information Science ICIEIS'2013, Kuala Lumpur, Malaysia, 12- 14 November 2013.

6- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Building a Semantic Index from HTML Pages or XML Documents", International Conference on Computing Technology and Information Management, ICCTIM 2014, Dubai, E.A.U, 09- 11 April 2014.

- **Publications**

1- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Vers une mise à jour de la mesure de Wu et Palmer pour l'amélioration du calcul de la distance sémantique", Proceedings Information Systems and Technologies, ISBN : 978-9931-9004-0-5.

2- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, "Extraction d'informations à partir de pages HTML ou de documents XML par une indexation sémantique en utilisant une ontologie de domaine", Proceedings Information Systems and Technologies, ISBN : 978-9938-9511-2-7.

3- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, “Toward an Update of a Similarity Measurement for a Better Calculation of the Semantic Distance between Ontology Concepts”, The Society of Digital Information and Wireless Communications (SDIWC) indexed in Research Bible, Nov-2013, ISBN N°: 978-0-9891305-2-3, http://paper.researchbib.com/?action=viewList&isbn=978-0-9891305-2-3.
4- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, “Building a Semantic Index from HTML Pages or XML Documents”, The Society of Digital Information and Wireless Communications (SDIWC) indexed in Research Bible, April-2014, ISBN N°: 978-0-9891305-5-4, http://paper.researchbib.com/?action=viewList&isbn=978-0-9891305-5-4&pid=14ou.
5- Abdeslem DENNAI & Sidi Mohammed BENSLIMANE, “Semantic Indexing of Web Documents Based on Domain Ontology”, International Journal of Information Technology and Computer Science (IJITCS), ISSN: 2074-9007 (Print), ISSN: 2074-9015 (Online), DOI: 10.5815/ijitcs, Published By: MECS Publisher, IJITCS Vol. 7, No. 2, January 2015.

- **Organisation du manuscrit**

Ce manuscrit est organisé en deux parties. La première partie (A) est composée de trois chapitres, le premier présente des concepts sur les applications et les documents web avec des exemples de documents tandis que le deuxième est dédié à la présentation des notions les plus importantes sur les ontologies et les mesures de similarité. Notre première contribution dans ce deuxième chapitre est un état de l’art sur les techniques de mesure de similarité. Le troisième chapitre est consacré pour une revue de littérature, d’une part, sur l’indexation en général et l’indexation sémantique en particulier, d’autre part, sur la rétro-ingénierie et proprement la rétro-ingénierie des applications web. La seconde partie (B) présente nos autres contributions. Elle est constituée de deux chapitres, dédiés respectivement, à la présentation d’un état de l’art sur l’indexation sémantique et la rétro-ingénierie des applications web, et à notre processus de rétro-ingénierie à base d’indexation sémantique.

Le détail de cette organisation est donné comme suit :

Le chapitre A.I est dédié à la présentation des applications web, ses composants, leur création et leur usage. Il décrit, aussi, les documents web non structurés et semi structurés avec quelques exemples tels que les HTML, XHTML, DTD, XML et XML schema.

Le chapitre A.II définit l’ontologie, ses constituants, ses dimensions de classification selon une typologie et son cycle de vie, il définit, également, les méthodes d’ingénierie ontologique, les outils et les langages de développement d’une ontologie. Enfin et dans le même chapitre, on

décrit la similarité en général que ce soit syntaxique ou sémantique avec ses différentes techniques.

Le chapitre A.III est consacré à présenter les différents types d'indexation y compris l'approche d'indexation sémantique en outre du processus de rétro-ingénierie.

Le chapitre B.I commence par présenter notre contribution récapitulant, d'une part un état de l'art sur les approches d'indexation sémantique suivi par notre apport par rapport à celles-ci, et d'autre part un état de l'art sur les processus de rétro-ingénierie suivi par notre apport par rapport à ceux-ci.

Le chapitre B.II propose notre apport à travers un nouveau processus de rétro-ingénierie des applications web à base d'indexation sémantique, constitué par quatre phases : La modélisation, l'attachement sémantique ou la validation, l'enrichissement et la re-conceptualisation.

Enfin, nous concluons ce travail par la présentation de ce qui a été fait et les perspectives envisageables à l'évolution de cette recherche, suivi par une partie d'annexes.

Partie A.

Considérations Théoriques.

CHAPITRE A.I.

Applications et Documents Web

Sommaire

Toute aborde d'un sujet de recherche nécessite des prés requis afin de le traiter convenablement et d'arriver, à la fin, à des résultats fiables et crédibles.

Ce premier chapitre de la première partie de ce travail est consacré pour présenter deux notions importantes à savoir les applications web et les documents web. Une application web, pour être exécutée, doit être hébergée dans un environnement internet (Web) et manipulée par le biais d'un navigateur web. Un document web est considéré comme l'un des composants d'une application orientée web, il est considéré comme étant l'un des formalismes de représentation des connaissances. À titre d'exemple nous allons décrire dans ce chapitre les pages HTML, XHTML, les documents DTD, XML et XML schema.

Pour notre sujet de recherche, nous avons pris comme exemple de cas à traiter, les documents XML et les pages HTML, deux sources d'information, respectivement, semi structurés et non structurés.

A.I.1. Applications web

A.I.1.1. Définition

Une application web est une application manipulable grâce à un navigateur web, de la même manière que les sites web. Une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau, etc.) [Conallen, 1999].

Les messageries web, les systèmes de gestion de contenu (CMS : Content Management System), les wikis et les blogs sont des applications web [LP_LBM_EISTI, 2006].

Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne et les logiciels de forum peuvent être sous forme d'application web.

Des appareils réseau tels que les routeurs sont parfois équipés d'une application web dans leur micro logiciel [LP_LBM_EISTI, 2006].

Les applications web font partie de l'évolution des usages et de la technologie du web.

A.I.1.2. Usage

La technologie des applications web permet de nombreux usages. Les plus populaires sont les moteurs de recherche, le webmail, le e-commerce et les jeux en ligne [Myers and al., 1996].

- Un moteur de recherche est une application web qui recherche des documents.
- Un webmail est une application web pour recevoir et envoyer du courrier électronique.
- Un CMS est une application web qui présente des documents. La présentation des documents est similaire à celle d'un site web, cependant les documents sont générés par le logiciel lors de chaque demande. Le CMS effectue les traitements nécessaires à la mise en forme et la présentation des documents.
- Un weblog est un CMS où des éléments de contenu sont présentés dans l'ordre chronologique de leur date de création.
- Un wiki (De l'hawaïen wikiwiki qui signifie vite) est un CMS qui vise à simplifier la création collaborative des documents. Il autorise plusieurs personnes à effectuer des modifications simultanées et est équipé d'espaces de discussion.
- Un site web marchand est un CMS où le contenu est des annonces concernant des produits. Il est utilisé pour la vente par correspondance. Les visites et les opérations d'achat sont enregistrées à des fins de marketing. Les sites web marchands sont utilisés aussi bien pour la vente des produits d'une société que pour des ventes entre particuliers ou des ventes aux enchères.

- Un jeu par navigateur est un jeu vidéo réalisé sous la forme d'une application web.
- Un logiciel de forum permet des discussions ouvertes entre des utilisateurs : Un utilisateur écrit un message et ce message peut être lu par tous les autres utilisateurs. Les logiciels de forum sont parfois réalisés sous forme d'application web.
- La messagerie instantanée permet l'échange instantané de messages texte entre différents utilisateurs. Les logiciels de messagerie instantanée sont parfois réalisés sous forme d'application web. Les messages peuvent être transmis à un autre utilisateur du logiciel ou un téléphone portable via le Short Message Service (SMS).
- Google Maps est une application web qui permet de consulter des cartes géographiques du monde entier.
- Facebook est une application web qui permet à chaque utilisateur de se constituer un réseau social (Amis, associés, personnes qui partagent les mêmes centres d'intérêt).

Ces applications sont caractérisées par : [Myers and al., 1996]

- L'accès universel des individus avec des qualifications limité ou même sans qualifications aux applications informatiques introduit le besoin de nouvelles interfaces homme-machine capables d'attirer l'attention du client et de faciliter l'accès à l'information.
- La disponibilité globale des émetteurs d'informations hétérogènes exige la gestion intégrée du contenu structuré et non structuré, probablement entreposé dans différents systèmes (Bases de données, systèmes de fichiers, dispositifs de stockage multimédia) et des sites multiples finis distribués.

A.I.1.3. Création

Les applications web sont souvent créées par des équipes composées à la fois de développeurs et de designers.

Le développement nécessite la connaissance des différents langages utilisés dans les technologies du Web : HTML pour la présentation des pages, CSS (Cascading Style Sheets) pour la charte graphique, JavaScript, Java ou ActionScript pour les automatismes exécutés par le client, ainsi qu'un langage tel que Java, PHP (Personal Home Page ou Hypertext Preprocessor), C# ou VBScript (Visual Basic Script) pour les automatismes exécutés par le serveur [LP_LBM_EISTI, 2006].

Les applications web sont faites d'un ensemble de composants logiciels et de pages « porteuses » ; les composants sont regroupés dans des bibliothèques logicielles.

Un logiciel serveur web prévu à cet effet (Serveur d'applications web) exécute un composant donné lors de la réception de chaque requête. ASP.NET (Active Server Pages de Microsoft), Websphere, JBoss ou Apache Tomcat sont des logiciels serveurs d'application web [LP_LBM_EISTI, 2006].

Une application web est typiquement utilisée simultanément par plusieurs usagers ; elle est équipée de mécanismes de contrôle d'accès logique, ceux-ci sont basés sur les mécanismes de contrôle d'accès propre au serveur d'application web et au système d'exploitation. Ils utilisent parfois des mécanismes existants tels que l'authentification unique (*Single sign-on*).

Pour les travaux de construction, les ingénieurs utilisent des environnements de développement intégré qui aident à la fois à la programmation informatique et à la conception de site Web tels que Visual Studio ou Eclipse [LP_LBM_EISTI, 2006].

A.I.1.4. Composants d'une application web

Les applications Web contiennent plusieurs composants qui sont liés ensemble pour fournir la fonctionnalité de l'application (cf. Figure n° 1). Ces composants sont écrits en divers langages de programmation et exécutés sur des machines multiples distribuées à travers le réseau. Chaque composant est écrit en un langage approprié pour implémenter sa fonctionnalité. Les langages de scripts sont utilisés pour assembler les différents composants, et les bases de données sont utilisées par les composants pour stocker et partager leurs données [Hassan, 2002].

Il y a un ensemble reconnaissable de composants qui constituent ce système. Cet ensemble se compose des navigateurs Web (Utilisés par les clients), des serveurs Web, des serveurs d'application et des composants suivants :

- Pages statiques : Celles-ci contiennent seulement le code HTML et le code exécutable qui fonctionne sur le navigateur Web. Elles sont servies par le serveur Web et elles n'ont pas besoin d'être prétraitées par le serveur d'application.
- Pages actives : comme ASP et JSP (Java Server Pages). Ces pages contiennent un mélange de balises HTML et de code exécutable. Quand une page active est demandée, le serveur d'application la prétraite et intègre des données de diverses ressources telles que les objets Web ou les bases de données, pour produire une page Web final HTML envoyé au navigateur.
- Les objets Web : ces derniers sont des morceaux de code compilé qui fournissent un service au reste du système logiciel par une interface définie. Ils sont supportés par des technologies distribuées telles que CORBA (Common Object Request Broker Architecture), EJB (Enterprise JavaBeans) et DCOM (Distributed Component Object Model). Ils ne sont pas des objets dans le sens de la programmation orienté-objet définis dans C++ ou Java.

- Objets multimédia : Comme les images et les vidéos.
- Bases de données : Celles-ci sont utilisées pour stocker les données communes entre les divers composants.

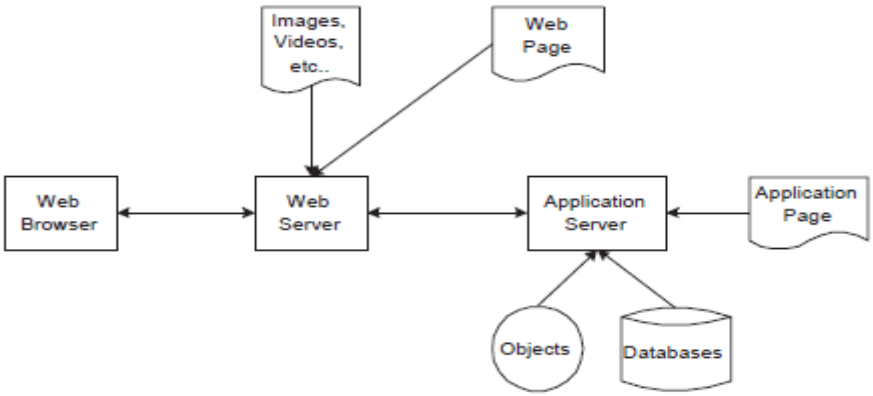

Fig. 1. Vue physique de l'architecture d'une application web (Non traduit) [Hassan, 2002].

La figure n° 1 montre le flux de données entre les divers composants d'une application Web. L'utilisateur de l'application utilise le navigateur Web pour accéder à la fonctionnalité de l'application Web. L'utilisateur se met en interaction avec le navigateur en cliquant sur des liens et en remplissant des champs du formulaire.

Le navigateur à son tour transmet les actions de l'utilisateur au serveur Web. Les demandes sont envoyées en utilisant le protocole http (HyperText Transfer Protocol). Quand il reçoit la demande, le serveur Web détermine s'il peut accomplir la demande directement ou si le serveur d'application doit être appelé. Le serveur Web sert directement le contenu de pages statiques et de multimédia HTML tel que les images, les vidéos, ou les fichiers audio; ou il passe la demande au serveur d'application. Le serveur d'application traite les pages actives et renvoie le résultat au serveur Web comme pages statiques HTML. Le serveur Web renvoie la page HTML au navigateur Web demandeur, qu'il l'affiche à l'utilisateur [Bouchiha, 2005].

A.I.2. Documents web

A travers leurs façons très fines de description des documents et des liens entre leurs différentes parties, les documents XML (eXtensible Markup Language) et les pages HTML sont utilisés dans la structuration du web ce qui offre des possibilités de combinaison entre la recherche d'information et l'interrogation des bases de données dans le web.

A.I.2.1. Les données web non structurées

Le développement rapide du World Wide Web (WWW) et le succès du langage HTML ont permis la construction de milliers de sites web générant une quantité importante d'information accessible sur Internet [Moussa et al., 2001].

Lors de la construction de la plupart de ces sites, l'approche la plus courante consiste à se focaliser beaucoup plus sur l'implantation d'une solution que sur le processus de développement.

Ces sites web présentent un ensemble de pages HTML, (i) statiques : Le contenu ne varie que lorsque l'administrateur du serveur les modifie ou (ii) interactives et dynamiques : Le contenu dépend soit des informations localisées sur le serveur (Connexion avec une base de données par exemple), soit de paramètres donnés de façon claire par le navigateur du client [Moussa et al., 2001].

Quelques outils de développement ont permis d'apporter une aide appréciable dans la génération et la mise en œuvre rapide d'applications web, à l'aide des technologies ASP, JSP, PHP, PL-SQL (Procedural Language-Structured Query Language) (Oracle-Web), etc. Ces technologies permettent d'extraire dynamiquement des informations à partir de diverses sources de données et de les inclure dans des modèles de pages HTML.

Dans ces applications, les concepteurs ont souvent privilégié l'aspect présentation au détriment de la structuration des données. C'est lors de l'exploitation de ces sites que cette approche montre ses limites. Les problèmes posés sont souvent dus à l'augmentation de la taille et de la complexité des sites, au besoin d'une interopérabilité avec d'autres applications, à la nécessité de modifications au fil du temps et au manque de possibilités de requête des pages HTML [Moussa et al., 2001].

A.I.2.2. Les données web semi structurées

La norme XML en tant que telle doit être vue comme un outil permettant de définir un langage (On dit alors qu'il s'agit d'un langage de structuration et de balisage ou tout simplement d'un métalangage), permettant de créer des documents structurés à l'aide de balises. Ce métalangage qui a favorisé l'expression des spécifications des standards et des normes de description, comme RDF, DC (Dublin Core), LOM (Learning Object Metadata) ou MPEG-7 (Motion Picture Expert Group 7), …, peut offrir la possibilité de créer des documents qui peuvent être traités comme une base de données intrinsèque. Ils sont auto-descriptifs, extensibles et surtout convertibles en plusieurs autres formats : HTML, XML, PDF (Portable Document Format), RTF (Rich Text File, etc. à l'aide des feuilles de styles, définies, elles mêmes par un langage en XML qui s'appelle XSL (eXtensible Stylsheet Language) [Moussa et al., 2001].

En plus ces documents peuvent être conformes à des structures, basées elles-mêmes sur le langage XML selon deux recommandations existantes qui sont DTD (Document Type Definition) et XML Schema [Moussa et al., 2001].

A.I.2.3. Quelques exemples de documents web

A.I.2.3.1. Pages HyperText Markup Language (HTML)

- **Définition**

Une page HTML est un simple fichier contenant du texte formaté avec des balises HTML. Elle peut être construite à partir du plus basique des éditeurs de texte (Une application bloc-notes par exemple), mais il existe des éditeurs beaucoup plus évolués [Tim Berners et Mark, 2000 ; James et Robert, 2000].

Les éditeurs WYSIWYG («What You See Is What You Get», littéralement «ce que vous voyez est ce que vous obtenez») sont des éditeurs graphiques permettant de travailler sur une page web telle qu'elle sera affichée sur un navigateur à quelques détails près. Grâce à ce genre d'éditeurs il est possible d'ajouter des balises par simple clic et d'en modifier les attributs en éditant leurs propriétés dans un formulaire. Pour autant, afin d'utiliser au mieux ce genre d'éditeur, une connaissance préalable du HTML est tout de même très utile.

Il existe également des éditeurs permettant d'éditer le code HTML en affichant les balises, les attributs et leurs valeurs avec différentes couleurs pour une meilleure lecture et proposant parfois des outils pour vérifier la validité du code HTML.

Par convention l'extension donnée à une page HTML est .htm ou .html mais une page web peut potentiellement porter n'importe quelle extension, en voici quelques unes :

- .asp pour une page générée dynamiquement en ASP ;
- .cgi pour une page générée dynamiquement avec des CGI (Common Gateway Interface) ;
- .php, .php3 ou .php4 pour une page générée dynamiquement en PHP ;
- .pl pour une page générée dynamiquement en Perl (Practical extraction and report language).

- **Structure d'une page HTML**

Une page HTML commence par la balise <HTML> et finit par la balise </HTML>. Il contient également un en-tête décrivant le titre de la page, puis un corps dans lequel se trouve le contenu de la page. L'en-tête est délimité par les balises <HEAD> et </HEAD>. Le corps est délimité par les balises <BODY> et </BODY>.

Ci-après un simple exemple d'une page HTML :

```
<HTML>
  <HEAD>
    <TITLE>Titre de la page</TITLE>
  </HEAD>
```

```
<BODY>
  Contenu de la page
</BODY>
</HTML>
```

- **Déclaration du type de document**

Il est conseillé d'indiquer dans la page HTML le prologue du type de document, c'est-à-dire une référence à la norme HTML utilisée, afin de spécifier le standard utilisé pour le codage de la page. Cette déclaration se fait par une ligne du type :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>...</HEAD>
    <BODY> Contenu de la page </BODY>
</HTML>
```

La déclaration du document indique la DTD (Document Type Definition) utilisée, c'est-à-dire la référence des caractéristiques du langage utilisé.

A.I.2.3.2. Pages eXtensible HyperText Markup Language (XHTML)

[W3C Recommendation, 2002]

- **Définition**

Est un langage de balisage servant à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML. XHTML se fonde sur la syntaxe définie par XML, plus récente, mais plus simple que celle définie par SGML sur laquelle repose HTML. Il s'agissait en effet à l'époque de tirer parti des gains techniques attendus de la simplification offerte par XML.

Comme de nombreux langages fondés sur XML, celui-ci commence par la lettre X, qui représente le mot eXtensible. Ainsi le premier document décrivant officiellement XHTML s'appelle XHTML™ 1.0 The Extensible HyperText Markup Language (« XHTML 1.0 Le langage de balisage hypertexte extensible »). C'est cependant l'abréviation XHTML qui est une marque du World Wide Web Consortium (W3C) et qui est seule utilisée dans les spécifications qui ont suivi la version 1.0.

- **Conversion de HTML en XHTML**

Cet exemple illustre les différences syntaxiques les plus courantes entre un document écrit en HTML 4 et en XHTML 1.0.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<title> Exemple HTML 4</title>
<ul>
<li> Des éléments comme HTML, HEAD et BODY sont implicites, leurs balises ouvrantes et fermantes sont optionnelles. </li>
<li> De nombreuses balises fermantes sont optionnelles, notamment pour P (Paragraphe) et LI (Entrée de liste).
<li> Les noms d'éléments et d'attributs peuvent <EM Class="important"> librement </Em> mélanger majuscules et minuscules. </li>
<li> Certains attributs ont une valeur par défaut <input type="checkbox" checked value="...">. </li>
<li> Les guillemets ne sont pas <em class=important>toujours </em> obligatoires autour des valeurs d'attribut. </li>
<li> Les éléments vides n'ont pas de syntaxe fermante <img src="i.png" alt="i">. </li>
</ul>
```

À l'inverse de la syntaxe HTML permissive ci-dessus, le même document doit être « bien formé » pour respecter les règles d'écriture du XHTML :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD /xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Exemple XHTML 1.0</title>
</head>
<body>
<ul>
<li> Tous les éléments doivent être explicitement balisés. </li>
<li> Les balises fermantes ne sont pas optionnelles. </li>
<li> Les noms d'éléments et d'attributs <em class="important">doivent </em> être en minuscules. </li>
<li> Tous les attributs doivent avoir une valeur explicite <input type="checkbox" checked
="checked" value ="..." />. </li>
<li> Les guillemets sont <em class="important">toujours</em> obligatoires autour des valeurs d'attribut.
</li>
<li> Les éléments vides doivent être fermés <img src="i.png" alt="i"/>. </li>
</ul>
</body>
</html>
```

- **Déclaration XML**

La déclaration XML n'est requise que dans quelques conditions d'encodage [Jeu de caractères autre qu'UTF-8 (Universal Character Set Transformation Format - 8 bits) en particulier] et lorsque le document est traité en tant que document XML (Type de contenu application/xhtml+xml). La plupart des documents XHTML 1.0 ne l'exigent donc pas. Il entraîne par ailleurs dans le navigateur web Internet Explorer 6.0 un mode d'interprétation problématique des Cascading Style Sheets et des scripts JavaScript. Néanmoins, la version Internet Explorer 8 tend à se rapprocher des autres navigateurs et à se conformer au CSS.

En fonction du jeu de caractères retenu, le document peut donc commencer par l'instruction suivante mise en première ligne :

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

Quelques exemples de Déclaration de Type de Documents en XHTML :

XHTML 1.0 Strict

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

XHTML 1.0 Transitional

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

XHTML 1.0 Frameset

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

A.I.2.3.3. Document Type Definition (DTD)

- **Définition**

La DTD ou Définition de Type de Document est un document permettant de décrire un modèle de document SGML (Standard Generalized Markup Language) ou XML [W3C Recommendation, 1998].

Le modèle est décrit comme une grammaire de classe de documents : grammaire parce qu'il décrit la position des termes les uns par rapport aux autres, classe parce qu'il forme une généralisation d'un domaine particulier, et document parce qu'on peut former avec un texte complet.

Une DTD décrit les documents à deux niveaux : la structure logique, que l'on peut assimiler à la syntaxe abstraite, et la structure physique, que l'on peut assimiler à la syntaxe concrète.

Au niveau de la structure logique, une DTD indique les noms des éléments pouvant apparaître et leur contenu, c'est-à-dire les sous-éléments et les attributs. En dehors des attributs, le contenu est spécifié en indiquant le nom, l'ordre et le nombre d'occurrences autorisées des sous-éléments. L'ensemble constitue la définition des hiérarchies valides d'éléments et de texte. En revanche, les DTD ne permettent pas de poser des contraintes sur la valeur du texte comme « le contenu de l'élément X est un entier en décimal », ou encore « dans l'élément Y, toutes les séquences de blancs sont équivalentes à un seul espace ».

Définir ce qui est valide est aussi le rôle des « schémas » comme Schéma XML, Relax NG (Regular Language for XML Next Generation) et Schematron mais ceux-ci sont préférentiellement exprimés en syntaxe XML alors que les DTD ont une syntaxe spécifique. Seule la DTD fait partie intégrante de la recommandation W3C du XML, et elle seule permet de valider un document XML du point de vue de cette recommandation.

La DTD d'un document peut être écrite à l'intérieur et à l'extérieur de ce document. La DTD finale est un regroupement des deux.

Au niveau de la structure physique, une DTD peut aussi définir des entités générales. Celles-ci ont l'un des rôles suivants :

- Une référence à un fragment de document externe, typiquement un autre fichier.
- Une abréviation pour un fragment de texte répétitif. Pour cette utilisation, la définition est plutôt dans le sous-ensemble interne.
- Un synonyme de caractère permettant des références par nom plutôt que par un code numérique.

- **Différences entre DTD-SGML et DTD-XML**

Il y a quelques différences entre les DTD pour SGML et celles pour XML. La plus significative du point de vue de la capacité d'expression est que les DTD pour XML ne permettent pas de restrictions sur l'imbrication des éléments et spécifient les arbres valides. Par exemple, dans la version SGML de HTML, un élément « A » (Pour les liens et ancres) ne peut contenir un autre élément « A » à n'importe quel niveau, même si la description générale du contenu mentionne indirectement « A » comme contenu possible de « A ». Cette restriction n'est pas exprimable dans la version XML des DTD.

Une autre restriction sur la capacité d'expression des DTD pour XML par rapport à SGML est la suppression des groupements non ordonnés : en SGML, écrire pour le contenu d'un élément X « A & B & C » signifie que X doit contenir les trois éléments A, B et C sans exigence sur l'ordre.

Dans les DTD pour XML, qui n'ont pas le connecteur « & », cela ne peut s'exprimer que par l'énumération explicite de tous les ordres possibles : La définition de contenu pour X s'écrit : « A,B,C ». L'explosion combinatoire que cela implique amène généralement à imposer dans les DTD pour XML un ordre qui n'est pas logiquement nécessaire pour le traitement de l'information du document.

- **Syntaxe**
  - PCDATA : Parsed Character DATA, représente un seul élément texte (Sans quantificateur).
  - | : permet d'ajouter des éléments prédéfinis au PCDATA.
  - " : défini la valeur par défaut du PCDATA.
  - * : quantificateur zéro ou plus.
  - + : quantificateur un ou plus.
  - ? : quantificateur zéro ou un.
- **Exemple**

```
< ?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE liste_de_gens [
 <!ELEMENT liste_de_gens (Personne)*>
 <!ELEMENT personne (Nom, date_de_naissance?, genre?, numero_de_secu?)>
 <!ELEMENT nom (#PCDATA)>
 <!ELEMENT date_de_naissance (#PCDATA)>
 <!ELEMENT genre (#PCDATA | masculin | féminin) "féminin">
 <!ELEMENT numero_de_secu (#PCDATA)>
]>
<liste_de_gens>
  <personne>
    <nom>Fred Bloggs</nom>
    <date_de_naissance>2008-11-27</date_de_naissance>
    <genre>masculin</genre>
  </personne>
</liste_de_gens>
```

A.I.2.3.4. Document eXtensible Markup Language (XML)

- **Définition**

La norme XML en tant que telle doit être vue comme un outil permettant de définir un langage (On dit alors métalangage), permettant de créer des documents structurés à l'aide de balises. Une balise est une chaîne de caractère du type : [W3C Recommendation, 2008]

<Balise>

Ainsi, un document XML, c'est-à-dire le fichier créé en suivant les spécifications de la norme XML pourra par exemple ressembler à ceci :

```
<Annuaire>
    <personne class = "étudiant">
        <Nom>Dennai</nom>
        <Prénoms>Abdeslem</Prénoms>
        <Téléphone>0771437242</Téléphone>
        <email>de_selam@yahoo.fr</email>
    </Personne>
    <Personne>
        ...
    </Personne>
</Annuaire>
```

Enfin il est possible d'ajouter des commentaires dans le document XML de la manière suivante :

<!-- Voici des commentaires XML -->

- **Structure d'un document XML**

En réalité un document XML est structuré en 3 parties :

La première partie, appelée prologue permet d'indiquer la version de la norme XML utilisée pour créer le document (Cette indication est obligatoire) ainsi que le jeu de caractères (En anglais encoding) utilisé dans le document (Attribut facultatif, ici on spécifie qu'il s'agit du jeu ISO-8859-1, jeu LATIN, pour permettre de prendre en compte les accents français). Ainsi le prologue est une ligne du type :

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

Le prologue se poursuit avec des informations facultatives sur des instructions de traitement à destination d'applications particulières. Leur syntaxe est la suivante :

<? instruction de traitement?>

La deuxième partie est une déclaration de type de document (A l'aide d'un fichier annexe appelé DTD - Document Type Definition).

Et enfin la dernière partie d'un fichier XML est l'arbre des éléments (Comme celui ci-dessus).

- **Syntaxe des éléments en XML**

L'arbre des éléments, c'est-à-dire le véritable contenu du document XML, est constitué d'une hiérarchie de balises comportant éventuellement des attributs.

Un attribut est une paire clé valeur écrit sous la forme Cle="Valeur", ainsi une balise affectée d'un attribut aura la syntaxe suivante :

<balise cle="valeur">

Toute donnée est ainsi encapsulée entre une balise ouvrante <balise> et une balise fermante </balise> (Sachant qu'une donnée peut éventuellement être un ensemble d'éléments XML). Ainsi un élément vide est uniquement constitué d'une balise spécifique dont la syntaxe est la suivante : <balise/>.

D'une part, il est interdit en XML de faire chevaucher des balises, c'est-à-dire d'avoir une succession de balises du type :

<balise1>
<balise2>
</balise1>
</balise2>

D'autre part, il est possible entre les balises (Donc pas à l'intérieur d'une balise) d'ajouter :

- Des espaces,
- Des tabulations,
- Des retours chariots.

Cela est très utile pour définir une indentation des balises (Ce qui est possible puisqu'elles ne se chevauchent pas).

```
<Annuaire>
  <personne class = "Etudiant">
    <Nom>Dennai</Nom>
    <Prénom>Abdeslem</Prénom>
    <Téléphone>0771437242</Téléphone>
    <email>de_selam@yahoo.fr</email>
  </Personne>
</Annuaire>
```

A.I.2.3.5. XML Schema

- **Définition**

XML Schema publié comme recommandation par le W3C en mai 2001 est un langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. Cette définition permet notamment de vérifier la validité de ce document.

Il est possible de décrire une organisation de vocabulaires d'origines différentes, par l'usage des espaces de noms. Il est possible de combiner les schémas eux-mêmes, et d'exprimer une combinaison pour le document contenu, comme quelqu'un qui parlerait de géographie et de sociologie dans un même texte [Hubert et Valérie, 2003].

Il est également possible, après une validation, de savoir avec quelle règle une information particulière a été testée : il s'agit du jeu de validation post-schema, ou PSVI (Post-Schema-Validation Infoset).

Une définition se compose d'un ou plusieurs documents XML, usuellement nommée (XML Schema Definition en anglais, ou fichier XSD).

Une instance d'un XML Schema est un peu l'équivalent d'une définition de type de document (DTD). XML Schema amène cependant plusieurs différences avec les DTD : Il permet par exemple de définir des domaines de validité pour la valeur d'un champ, alors que cela n'est pas possible dans une DTD ; en revanche, il ne permet pas de définir des entités ; XML Schema est lui-même un document XML, alors que les DTD sont des documents SGML.

Ce langage de description de contenu de documents XML est lui-même défini par un schéma, dont les balises de définition s'auto-définissent (C'est un exemple de définition récursive).

La recommandation du W3C 1.0 se compose d'un document de présentation (Non normatif), d'un document précisant comment définir la structure, et d'un document précisant comment définir les données. La dernière édition, de version 1.0, de cette recommandation, date de 2004. Le W3C travaille actuellement sur la version 1.1 dont l'objectif est de définir les notions de version de schéma, ainsi que des contraintes selon la présence de telle ou telle valeur [Hubert et Valérie, 2003].

- **Modèle de XML Schema**

La recommandation spécifie la validation des documents XML à partir d'un modèle abstrait ; elle en fournit le format XML.

Pour ce modèle abstrait, un schéma est un ensemble de composants, tels que :

- La déclaration d'éléments (La notion d'éléments vient de la recommandation XML),
- La déclaration d'attributs (Qui vient aussi de la recommandation XML),
- La définition de types simples (Valeurs constituées uniquement à partir d'une chaîne de caractères),
- La définition de types complexes (Valeurs constituées d'attributs et d'autres valeurs).

Elle introduit également le type anyType (N'importe quel type), base de tous les types utilisés.

Chaque composant est encadré par une unité d'information, au sens de la recommandation XML Information Set (Dite Infoset) du W3C.

Pour ce qui est de l'usage, XML Schema permet de définir éléments et types de valeur soit nommément, soit localement à un contexte nommé. La combinaison de ces deux modes permet de définir quatre techniques :

La première décrit les éléments en les emboîtant les uns dans les autres, la deuxième contient des définitions globales des éléments, mais locales des types, la troisième contient des définitions locales des éléments, mais globales des types et la quatrième contient la définition globale des éléments et des types.

XML Schema détermine l'unicité comme une combinaison de nœuds, relativement à un contexte, par rapport à leurs composants. Ainsi, on peut par exemple affirmer et vérifier qu'une personne est unique, dans le contexte d'un annuaire, par rapport à son nom et son prénom.

Un exemple de fichier XSD (Personne.xsd):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personne">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nom" type="xs:string" />
        <xs:element name="prenom" type="xs:string" />
        <xs:element name="date_naissance" type="xs:date" />
        <xs:element name="etablissement" type="xs:string" />
        <xs:element name="num_tel" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Suivi d'un fichier XML valide :

```
<?xml version="1.0" encoding="UTF-8"?>
<personne xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="personne.xsd">
  <nom>MBODJ</nom>
  <prenom>Babacar</prenom>
  <date_naissance>1996-10-06</date_naissance>
  <etablissement>NIIT</etablissement>
```

```
  <num_tel>764704140</num_tel>
</personne>
```

A.I.3. Conclusion

Dans ce chapitre, nous avons essayé d'exposer différents exemples de documents web non structurés ou semi structurés et à travers cette description nous avons pu détecter les points différents et les points communs entre ces documents.

Pour notre cas de recherche, nous allons prendre deux exemples de documents à savoir une page HTML pour les documents non structurés et un document XML pour les documents semi structurés en essayant d'appliquer le processus de rétro-ingénierie des applications orientées web à base d'indexation sémantique.

Le prochain chapitre va présenter des notions sur l'ontologie comme un moyen de validation et d'enrichissement des connaissances avec un état de l'art des techniques de mesure de similarité.

CHAPITRE A.II.

Ontologies et Mesures de Similarité

Sommaire

En poursuivant notre évocation des considérations théoriques requises pour notre sujet de recherche, ce deuxième chapitre de la première partie de notre travail est une revue de littérature d'autres notions théoriques.

Une première importante partie est consacrée à décrire la notion d'ontologie, qui sera la base de notre travail, ses constituants, ses dimensions de classification, ses méthodes et ses outils de construction ainsi que les langages ontologiques.

Une deuxième partie décrit le concept "mesure de similarité" qu'elle soit syntaxique ou sémantique avec quelques exemples de techniques de calcul, accomplie par une synthèse présentant quelques avantages et inconvénients de ces mesures et une récapitulation générale de ces techniques sans omettre de présenter à la fin la similarité sémantique ontologique.

Cette synthèse sur les techniques de mesure de similarité est un état de l'art et considéré comme notre première contribution dans ce travail.

Quelques parts dans ce chapitre, on aura l'occasion de rencontrer quelques définitions de concepts importants à savoir : La fréquence des termes et le calcul du poids des termes dans un corpus ou une section, en prenant comme exemple les deux formules *TF-IDF (Term Frequency - Inverse Document Frequency)* et *TF-ITDF (Term Frequency-Inverse Tag and Document Frequency).*

A.II.1. Notions d'ontologie

A.II.1.1. Qu'est ce qu'une ontologie ?

Elle est définit par les chercheurs comme suit :

- Dans le domaine de l'intelligence artificielle, Neches avec son équipe ont défini l'ontologie comme : « Les termes et les relations de base comportant le vocabulaire d'un domaine aussi bien que les règles pour combiner les termes et les relations afin de définir des extensions du vocabulaire » [Neches et al., 1991].
- Grüber et son équipe en 1993 à Stanford (Université de Stanford, USA) ont donné la définition suivante [Gruber, 1993] : « Une ontologie est une spécification explicite d'une conceptualisation ». Il est intéressant de noter que c'est la référence la plus consultée sur le web.
- La définition de Gruber a été modifiée légèrement par Borst en 1997 : « Spécification formelle d'une conceptualisation partagée » [Borst, 1997].
- Puis modifiée par Studer en 1998 « Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée » [Studer et al., 1998].

Une explication à ces définitions :

- ***Une conceptualisation*** fait référence à un modèle abstrait de certains phénomènes du monde, modèle qui identifie les concepts pertinents de ce phénomène.
- ***Explicite*** signifie que le type des concepts utilisés et les contraintes sur leur utilisation sont explicitement définis.
- ***Formelle*** se réfère au fait que l'ontologie doit être compréhensible par les machines.
- ***Partagée*** reflète la notion de connaissance consensuelle décrite par l'ontologie, c'est-à-dire qu'elle n'est pas restreinte au point de vue de certains individus seulement, mais reflète un point de vue plus général, partagé et accepté par un groupe.

- Une autre définition de Shreiber et ses coéquipiers, en 1995 « Une ontologie fournie le moyen pour décrire d'une manière explicite la conceptualisation des connaissances représentées dans les bases de connaissances » [Shreiber et al., 1995].
- Une ontologie définit les termes utilisés pour décrire et représenter un domaine de la connaissance, défini Par W3C (World Wide Web Consortium).

A.II.1.2. Constituants d'une Ontologie

Les connaissances traduites par une ontologie sont à véhiculer à l'aide des éléments suivants [Dahmani, 2010] :

- **Concept**

Appelé aussi terme ou classe de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (Le domaine du problème considéré), retenue en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Selon Gómez et Pérez en 1999, Les concepts peuvent être classifiés selon plusieurs dimensions :

- Niveau d'abstraction (Concret ou abstrait),
- Atomicité (Elémentaire ou composée),
- Niveau de réalité (Réel ou fictif).

Différents auteurs conviennent qu'un concept peut être représenté par différents termes.

- **Relation ou lien**

Traduit une association (Pertinente) existante entre les concepts présents dans le segment analysé de la réalité. Les relations incluent les associations suivantes:

- Sous-classe-de (Généralisation – spécialisation souvent appelé lien « I*s a* »),
- Partie-de (Agrégation ou composition),
- Associée-à,
- Instance de, etc.

Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres. Les relations taxonomiques ou de subsomption, à priori, vont permettre de construire des hiérarchies strictes entre les concepts.

Pour cela dans les différents travaux sur les ontologies, seule la relation « *Is a* » est généralement citée de taxonomique ou de subsomption. Cependant dans certains cas particuliers d'ontologies, d'autres relations peuvent être taxonomiques, c'est le cas de la relation de composition « *part of* ».

- **Fonction**

Constitue des cas particuliers de relations, dans laquelle un élément de la relation, le Nième (Extrant) est défini en fonction des (N-1) éléments précédents (Intrants).

Nous notons néanmoins que ce constituant d'ontologie est rarement évoqué dans la description d'ontologies. Nous pensons que cela est du au fait que les relations « fonctionnelles » entre concepts sont plutôt présentes dans certains domaines scientifiques comme la physique, la chimie, etc.

- **Axiome ou règle d'inférence**

Permettant de définir certaines propriétés des relations sous forme d'assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie. Les axiomes représentent les intensions des concepts et des relations du domaine et de manière générale, les connaissances n'ayant pas un caractère strictement terminologique. Ils spécifient la façon dont les primitives terminologiques du domaine (i.e. les concepts et relations) peuvent être utilisées. Ces axiomes sont spécifiques aux ontologies et les distinguent des thesaurus, qui ne représentent que des terminologies alors que les ontologies intègrent des connaissances au sens large. Certains axiomes se retrouvent dans de nombreuses ontologies et/ou sont communs à de nombreuses primitives de l'ontologie. On appelle ici ces axiomes particuliers des schémas d'axiome, ces derniers peuvent être :

Ces schémas d'axiomes, peuvent être :

- Les propriétés algébriques d'une relation (Symétrie, réflexivité, transitivité),
- La propriété de subsomption entre concepts ou entre relations (Relation « Is a »),
- La cardinalité d'une relation.

Certains schémas d'axiomes sont intégrés dans les formalismes de représentation de connaissances utilisés pour décrire des ontologies. Par exemple, la relation « Is a » apparaît bien dans les formalismes de type Entité-Relation et Graphes Conceptuels.

- **Instance**

Constitue la définition extensionnelle de l'ontologie, cet objet véhicule les connaissances (Statiques, factuelles) à propos du domaine du problème.

A.II.1.3. Dimensions de classification

Une ontologie, peut être classifiée selon l'objet de conceptualisation ou le niveau de détail ou le degré de formalisme et ainsi le niveau de complétude [Valéry et al., 2003].

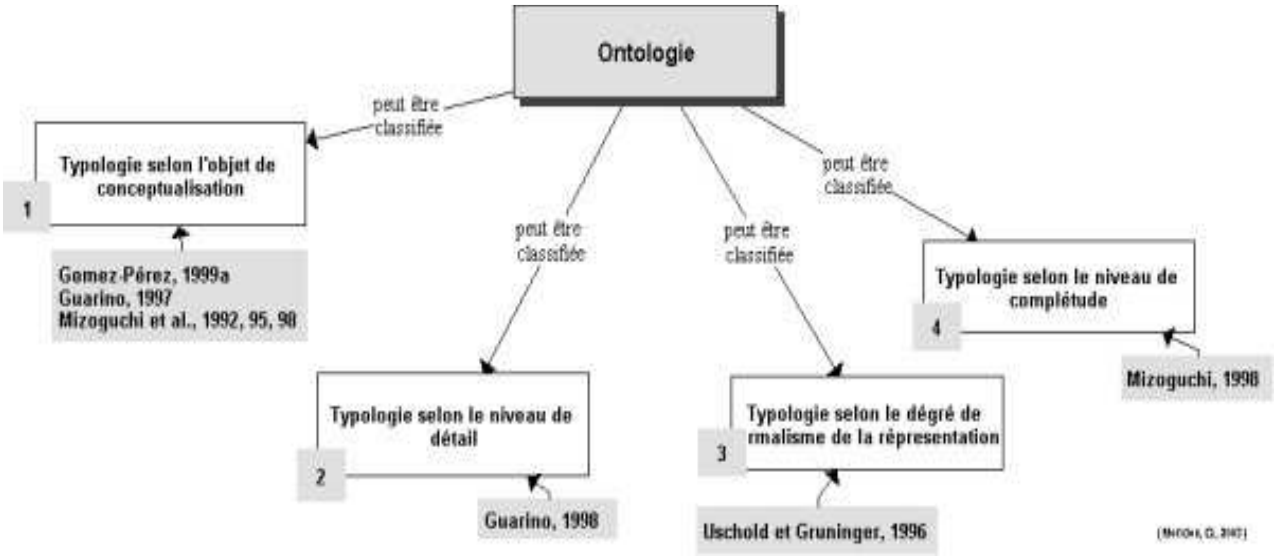

Fig. 2. Dimensions de classification de l'ontologie [Valéry et al., 2003].

A.II.1.3.1. Typologie selon l'objet de conceptualisation

Cette section présente les types d'ontologies les plus couramment utilisés afin de permettre au lecteur d'avoir une idée des connaissances à inclure dans chaque type d'ontologie. En gros, on identifie les catégories suivantes :

- **Ontologie de représentation de connaissances**

Ce type d'ontologies regroupe les concepts (Primitives de représentation) impliqués dans la formalisation des connaissances. Un exemple est l'ontologie de Frame qui intègre les primitives de représentation des langages à base de frames : Classes, instances, facettes, propriétés/slots, relations, restrictions, valeurs permises, etc. [Valéry et al., 2003 ; Valéry, 2007].

- **Ontologie supérieure ou de Haut niveau**

Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que : Les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations et les propriétés.

L'ontologie de haut de niveau est fondée sur : La théorie de l'identité, et la théorie de la dépendance ; la mariologie (La théorie comme ensemble est une tentative de fixer les principes qui sous-tendent les relations entre un ensemble entités et ses composantes, quelle que soit la nature de l'entité).

Guarino et Sowa ont poursuivi chacun indépendamment des recherches sur la théorie de l'ontologie. Tous deux intègrent les fondements philosophiques comme étant des principes à suivre pour concevoir l'ontologie supérieure. Sowa introduit deux concepts importants : Continuant et occurrent et obtient douze catégories supérieures en combinant sept propriétés primitives. L'ontologie supérieure de Guarino consiste en deux mondes : Une ontologie des Particuliers (Des choses qui existent) et une ontologie des universels comprenant les concepts nécessaires à décrire les Particuliers. La conformité aux principes de l'ontologie supérieure a son importance, lorsque le but est de standardiser la conception des ontologies [Valéry, 2007].

- **Ontologies Génériques (Generic ontology)**

Appelées Les méta-ontologies ou noyaux d'ontologies, sont réutilisables dans différents domaines. Les deux exemples les plus représentatifs sont une ontologie méréologique, qui inclut le terme « partie de » et l'ontologie topologique incluant le terme « associé-à » [Valéry et al., 2003].

- **Ontologies du domaine (Domain ontology)**

Cette ontologie régit un ensemble de vocabulaires et de concepts qui décrit un domaine d'application ou monde cible. Elle permet de créer des modèles d'objets du monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif. La plupart des ontologies existantes sont des ontologies du domaine [Valéry, 2007].

- **Ontologies de Taches (Task ontology)**

Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de tutorat, soit tout ce qui concerne la résolution de problèmes. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes inhérente aux tâches et indépendante du domaine [Valéry et al., 2003].

- **Ontologies d'application (Application ontology)**

Cette ontologie est la plus spécifique. Les concepts dans l'ontologie d'application correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité. Ce sont les plus spécifiques.

Dans cette classification, la notion d'ontologie d'application définit le contexte d'une application qui décrit la sémantique des informations et des services manipulés par une ou un ensemble d'applications sur un même domaine [Valéry et al., 2003 ; Valéry, 2007].

A.II.1.3.2. Typologie selon le niveau de détail de l'ontologie

Par rapport au niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé, deux catégories peuvent être identifiées [Valéry, 2007] :

- Granularité fine : correspondant à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche. Ce niveau de granularité peut s'avérer utile lorsqu'il s'agit d'établir un consensus entre les agents qui l'utiliseront.
- Granularité large : correspondant à un vocabulaire moins détaillé comme par exemple dans les scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés subséquemment dans d'autres ontologies de domaine ou d'application.

A.II.1.3.3. Typologie selon le niveau de complétude

On peut définir trois niveaux de complétude [Valéry, 2007] :

Niveau 1 – Sémantique (Interprétatif) : Tous les concepts, caractérisés par un terme/libellé, doivent respecter les quatre principes différentiels :

- Communauté avec l'ancêtre,
- Différence, spécification, par rapport à l'ancêtre,
- Communauté avec les concepts frères, situés au même niveau,
- Différence par rapport aux concepts frères.

Ces principes correspondent à l'engagement sémantique qui assure que chaque concept aura un sens univoque et non contextuel associé. Deux concepts sémantiques sont identiques si l'interprétation du terme/libellé à travers les quatre principes différentiels aboutit à un sens équivalent.

Niveau 2 – Référentiel (Formel) : Les concepts référentiels ou formels, se caractérisent par un terme/libellé dont la sémantique est définie par une extension d'objets. L'engagement ontologique spécifie les objets du domaine qui peuvent être associés au concept, conformément à sa signification formelle.

Deux concepts formels seront identiques s'ils possèdent la même extension. Par exemple, les concepts d'étoile du matin et d'étoile du soir associés à Vénus.

Niveau 3 – Opérationnel (Computationnel) : Les concepts du niveau opérationnel ou computationnel sont caractérisés par les opérations qu'il est possible de leur appliquer pour générer des inférences ou engagement computationnel.

A.II.1.3.4. Typologie selon le niveau du formalisme

Le degré de formalité par laquelle un vocabulaire est créé et le sens est précisé varie considérablement quatre points sont [Valéry et al., 2003] :

- Très informel : Exprimée librement dans la langue naturelle.
- Semi-informel : Exprimée sous une forme restreinte et structuré du langage naturel, augmentant considérablement la clarté en réduisant l'ambiguïté.
- Semi-formel : Exprimée dans un langage formellement défini artificielle.
- Formel : Utilisation d'un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves des propriétés telles la robustesse et l'exhaustivité.

A.II.1.4. Principes à suivre dans le cadre de l'élaboration d'une ontologie

Nous avons certains critères conceptuels et un ensemble de principes qui se sont avérés efficaces dans le domaine de la conception d'ontologies [Gruber, 1993].

- **Clarté et objectivité.**

  L'ontologie devrait fournir le sens des termes définis en offrant des définitions objectives ainsi que de la documentation en langage naturel.

- **Complétude et perfection.**

  Une définition exprimée par une condition nécessaire et suffisante est préférable à une définition exprimée seulement par une condition nécessaire ou par une condition suffisante.

- **Cohérence.**

  Afin de pouvoir formuler des inférences cohérentes avec les définitions.

- **Extensibilité monotone maximale.**

  Les nouveaux termes, qu'ils relèvent de la langue générale ou d'une langue de spécialité, devraient être inclus dans l'ontologie sans entraîner de modifications dans les définitions existantes.

- **Interventions ontologiques minimales.**

  Intervenir le moins possible sur le monde en phase de modélisation. L'ontologie devrait spécifier le moins possible le sens de ses termes, de façon à ce que les parties impliquées dans l'ontologie aient les mains libres pour spécialiser et instancier l'ontologie à leur guise.

- **Distinction ontologique.**

  Les classes d'une ontologie doivent être séparées. Le critère d'identité sera utilisé afin d'isoler le noyau des propriétés jugées invariables pour une instance d'une classe [Valéry, 2007].

- **Diversification des hiérarchies.**

  Afin d'optimaliser la puissance dérivant des mécanismes d'héritage multiple, Il est d'autant plus facile d'intégrer de nouveaux concepts (Dans la mesure où ils peuvent être définis sur la base des concepts et des critères de classification préexistants) et d'hériter de propriétés de différents points de vue que le volume de connaissances inclus dans l'ontologie est suffisant et que l'éventail de critères de classification est large [Valéry, 2007].

- **Modularité.**

  Afin de minimiser le couplage entre modules [Valéry, 2007].

- **Minimisation de la distance sémantique entre des concepts frères.**

  Les concepts proches sont regroupés et représentés dans des sous-classes d'une même classe et devraient être définis en ayant recours aux mêmes primitives, alors que les concepts plus éloignés sont éclatés dans la hiérarchie.

A.II.1.5. Cycle de vie d'une ontologie

Les activités liées à une ontologie peuvent être regroupées en trois catégories [Ahcene, 2005]:

- Activités de gestion de projet : Planification, contrôle, assurance qualité,
- Activités de développement : Spécification, conceptualisation, formalisation,
- Activités de support : Evaluation, documentation, gestion de la configuration.

La Figure n° 3 ci-après représente les différentes activités présentées par Fernandez et ses collègues En 1997 qui expliquent que le cycle de vie préconisé est un cycle par prototypes : « la vie d'une ontologie passe par les états suivants : Spécification, conceptualisation, formalisation, intégration, implantation et maintenance ».

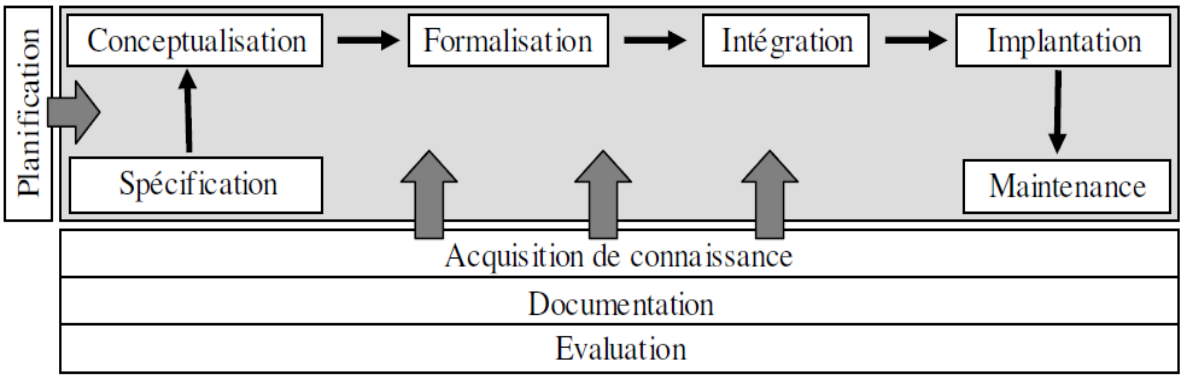

Fig. 3. Cycle de vie d'une ontologie selon Fernandez et ses collègues [Fernandez et al., 1997].

A.II.1.6. Méthodes d'ingénierie ontologique

Parmi les méthodes de construction des ontologies, on trouve :

- **Méthode d'Uschold et King**

La méthode d'Uschold et King en 1995 est la première méthode de construction d'ontologies. La figure n° 4 présente le processus de la méthode. Elle est basée sur l'expérience du développement d'Enterprise Ontology, qui est une ontologie pour les processus de modélisation d'une entreprise. Ce développement a été réalisé par l'Université de Edinburgh (Scotland). Elle se présente comme suit :

(1) Identification du but et de la portée de l'ontologie ;

(2) Construction de l'ontologie comprenant :

(a) Capture de l'ontologie,

(b) Programmation,

(c) Intégration des ontologies existantes ;

(3) Evaluation de l'ontologie ;

(4) Documentation de l'ontologie.

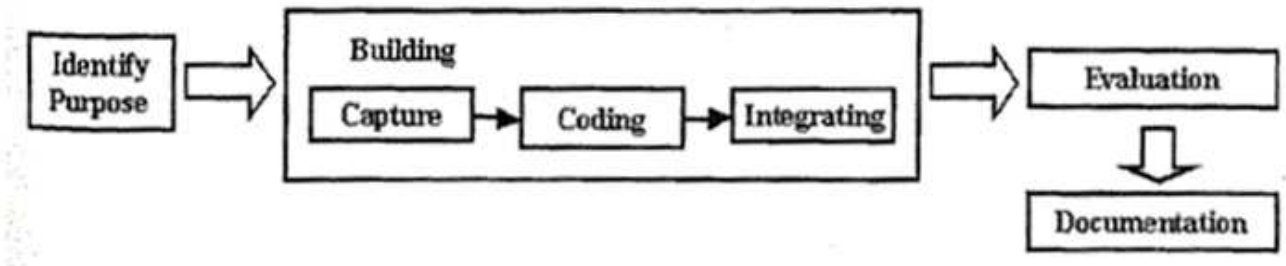

Fig. 4. Méthode d'Uschold et King [Uschold et King, 1997].

- **Methontology**

Proposée en 1997 par l'équipe d'intelligence artificielle de l'Université Polytechnique de Madrid, Espagne [Fernandez et al., 1997], permet de couvrir tout le cycle de vie d'une ontologie. Elle s'intéresse pratiquement à toutes les activités liées aux ontologies, c'est à dire aux activités de développement, de gestion de projet et de support.

Le processus de développement se base sur plusieurs représentations intermédiaires, obtenues le plus souvent par des mappings, permettant ainsi de changer de niveau. La phase principale est la conceptualisation [Ahcene, 2005].

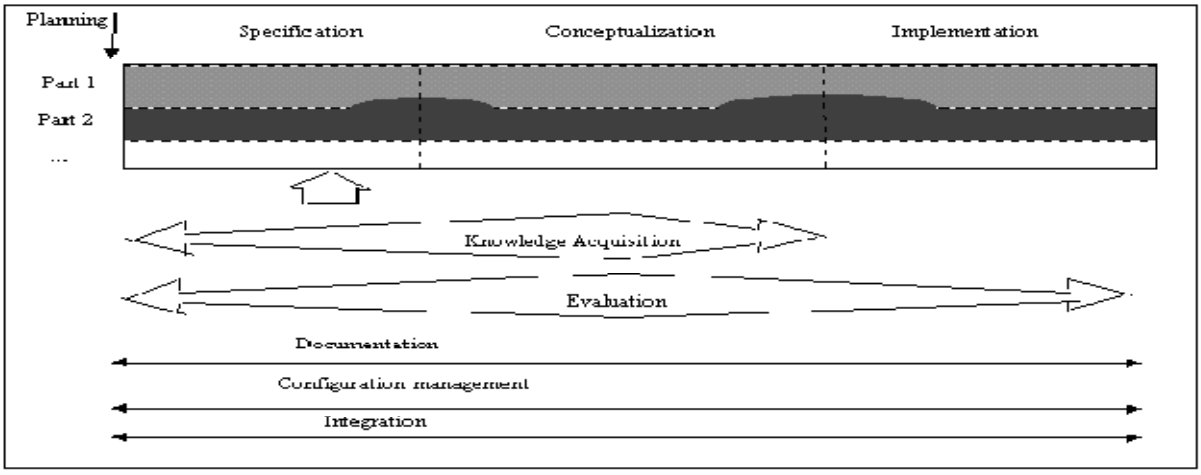

Fig. 5. Méthode de Methontology [Fernandez et al., 1997].

- **Spécification.**

  Le but de la spécification est de mettre sur pied un document qui couvre l'objectif principal et la portée de l'ontologie. Cette **spécification devrait être complète et concise.**

- **Conceptualisation**.

  Organise et structure la conceptualisation des connaissances acquises qui sont indépendants des langages et environnements de mise en œuvre. Plus précisément, cette phase organise et convertis les connaissances acquises et perçue d'un domaine dans une spécification semi-formelle, en utilisant ensemble de représentations intermédiaires que l'expert du domaine et l'ontologiste peuvent comprendre.

- **Acquisition de connaissances.**

  Il s'agit d'une activité indépendante dans le développement de l'ontologie.

- **Intégration.**

  Tout au long du développement de l'ontologie, nous avons identifié des termes qui pourraient inclure d'autres ontologies.

A.II.1.7. Quelques outils de construction d'une ontologie

Les outils peuvent être des outils dépendants ou indépendants du formalisme de représentation.

A.II.1.7.1. Outils dépendants de formalisme de représentation

Les outils dépendants de formalisme sont des outils qui présentent les ontologies de façon structurelle avec des formes de cette ontologie dépendants, parmi celle sont les suivant :

- **OntoSaurus.**

  Il est de l'Information Science Institute de l'Université de Southern California à l'USA. Il est composé de deux modules : Un serveur utilisant LOOM (Language for Object Oriented Methods) comme langage de représentation des connaissances, et en un serveur de navigation créant dynamiquement des pages HTML qui affichent la hiérarchie de l'ontologie; le serveur utilise des formulaires HTML pour permettre à l'usager d'éditer l'ontologie. Il utilise LOOM comme langage de représentation des connaissances. On peut représenter les concepts, la taxonomie des concepts, les relations entre les concepts, les fonctions, les axiomes et les instances [Swartout et al., 1997].

- **OilEd (Oil Editor).**

  supporte le développement d'ontologies de petites et moyennes tailles [Bechhofer et al., 2001].

- **Ontolingua.**

  Est un éditeur d'ontologies proposé par l'université de Stamford. Le terme « Ontolingua » renvoie en fait au langage utilisé pour construire les ontologies [Farqhuar et al., 1996].

- **WebOnto.**

  Est un outil basé sur le Web. Il a été développé par Knowledge Media Institute (KMI, laboratoire de technologie Open University, United Kingdom), donc WebOnto est principalement un outil graphique pour construire des ontologies qui rendent facile à utiliser et travailler en collaboration sur des ontologies [Domingue, 1998].

A.II.1.7.2. Outils indépendants de formalisme de représentation

- **Protégé 2000.**

  Est un outil de modélisation connaissance très populaire développé par Université de Stamford. Ontologies et bases de connaissances peuvent être modifiés interactivement dans Protégé et accessibles par une interface utilisateur graphique et de l'API (Application Programming Interface) Java [Musen et al., 2000].

Protégé peut être étendu avec des composants enfichables à ajouter de nouvelles fonctionnalités et de services. Il existe un nombre croissant de plugins offrant une variété de fonctionnalités en plus, telles que des outils supplémentaires de gestion de l'ontologie, support multimédia, l'interrogation et les moteurs de raisonnement, les méthodes de résolution de problèmes. Protégé apporte son soutien à la construction des ontologies qui sont à base de trames, en conformité avec le protocole Open Knowledge Base Connectivity (OKBC). Il existe différentes formes telles que RDF(s) (Resource Description Framework Schema), OWL (Web Ontology Language) et XML Schema dans lequel protégé ontologie peut être exporté [Ahcene, 2005].

- **OntoEdit.**

  Est également un environnement de construction d'ontologies indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généricité d'un concept [Maedche et al., 2003 ; Maedche, 2002].

  Un plug-in nommé ONTOKICK offre la possibilité de générer les spécifications de l'ontologie par l'intermédiaire de questions de compétences [Ahcene, 2005].

- **ODE (Ontology Design Environment).**

  Est l'environnement qui permet le développement d'ontologies au niveau des connaissances, Le but de ODE est de soutenir le fabricant de l'ontologie au cours du cycle de vie du processus de développement de l'ontologie, de la spécification des exigences, à travers les phases d'acquisition et de conceptualisation des connaissances [Ahcene, 2005].

- **WebOde.**

  Est une suite ontologie, ingénierie et extensible, basée sur un serveur d'applications dont le développement a débuté en 1998. Le noyau de WebODE était son service d'accès de l'ontologie, utilisé par tous les services et applications connectés au serveur. Ontologie de l'éditeur WebODE permis l'édition et la navigation [Blazquez et al., 1998].

A.II.1.8. Langages d'ontologies

Nous présentons ci-après (cf. figure n° 6) quelques langages de représentation des ontologies :

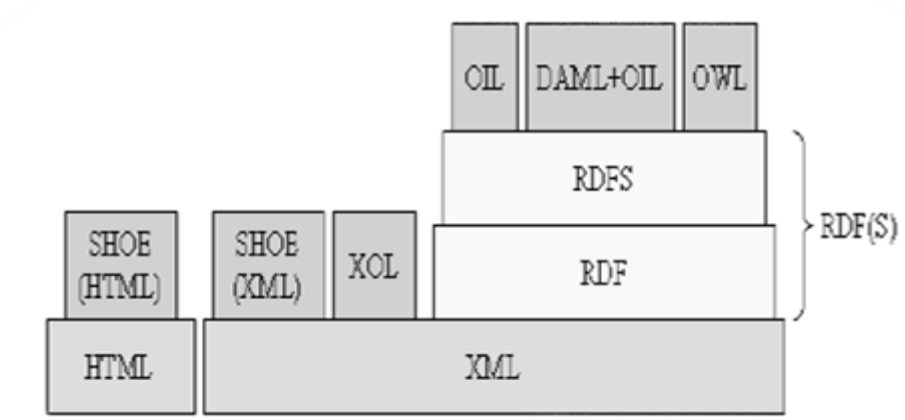

Fig. 6. Langages des ontologies [Maniraj et Sivakumar, 2010].

- **Cycl[2].**

  Est un Langage déclaratif basé sur la logique du premier ordre, avec des extensions pour les opérateurs de modèle et d'ordre supérieur quantification [Maniraj et Sivakumar, 2010].

- **KIF** (Knowledge Interchange Format).

  Est un langage pour l'échange de connaissances entre les programmes disparates. Il a une sémantique déclarative (i.e. le sens des expressions de la représentation peut être compris sans faire appel à un interprète pour les manipuler).

  Il est logique globale (i.e. qu'il permet l'expression de condamnations arbitraires dans le calcul des prédicats de premier ordre) [Maniraj et Sivakumar, 2010].

- **LOOM** (Language for Object Oriented Methods).

  Il n'est pas conçu pour le développement des ontologies mais pour les bases de connaissances. Il est basé sur la logique de description et les règles de production, il fournit une classification automatique des concepts [MacGregor, 1991].

- **OCML** (Operational Conceptual Modeling Language).

  Est un langage de modélisation de connaissances, qui permet la spécification et l'opérationnalisation des fonctions, des relations, des classes, des instances et des règles. Il comprend également des mécanismes pour définir des ontologies et des méthodes de résolution de problème [Motta, 1999].

- **F-Logic.**

  Il est un formalisme qui intègre la logique de la programmation orientée objet dans un mode propre et déclarative [Krötzsch, 2010].

[2] Http://www.cyc.com/cycdoc

- **SHOE (Simple HTML Ontology Extensions).**

  Est une extension du langage HTML qui permet aux auteurs de pages Web de générer une annotation des documents, compréhensible par la machine et le langage HTML (Hypertext Markup Language) est utilisé pour rendre la connaissance facilement lisible par un être humain [Heflin, 2001].

- **XML (eXtensible Markup Language).**

  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 [W3C Ubiquitous Web domain, 2014].

- **XOL (XML based Ontology exchange Language).**

  Est un langage d'échange d'ontologies. En langage d'échange, on entend que XOL est destiné à être utilisé comme un langage intermédiaire pour transférer des ontologies entre différents systèmes de base de données, des outils de développement ontologie ou des programmes d'application [Bechhofer, 2002].

- **OIL (Ontology Inference Layer).**

  Il est basé sur les propositions existantes, telles que XOL et RDF.

  La description de l'ontologie est divisée en trois couches : La couche objet (Instances concrètes), la couche de premier méta-niveau (Définition de l'ontologie) et la couche de second méta-niveau (Définition des caractéristiques de l'ontologie) [Valéry, 2007].

  OIL permet de définir des classes et des relations et un nombre limité d'axiomes. Les relations sont considérées comme des classes et peuvent être organisées hiérarchiquement [Bechhofer, 2002].

- **RDF.**

  Permet d'encoder, d'échanger et de réutiliser des métadonnées structurées. Il a été créé pour gérer les métadonnées de documents XML et peut aussi être utilisé pour des ontologies [Bechhofer, 2002].

- **DAML+OIL.**

  Est un langage de balisage sémantique pour les ressources Web. Il s'appuie sur les normes du W3C antérieures comme RDF et RDF Schema et étend ces langues avec les plus riches primitives de modélisation [Bechhofer, 2002].

- **OWL.**

  Le langage Web Ontology est conçu pour être utilisé par les applications qui ont besoin de traiter le contenu de l'information au lieu de simplement présenter des informations à l'homme.

OWL facilite une plus grande machine à l'intelligibilité du contenu Web que celui soutenu par XML, RDF, et RDF-S en fournissant un vocabulaire supplémentaire avec une sémantique formelle. OWL a trois sous-langages plus expressifs: OWL Lite, OWL DL et OWL Full [Bechhofer, 2002].

A.II.2. Mesures de similarité

Le choix d'une mesure de similarité est tout à fait crucial pour une bonne exécution du raisonnement [Bisson, 2000]. Il s'agit en effet de trouver la meilleure adéquation entre le but à atteindre et le type de connaissances manipulées. Elle a été considérée comme un sujet de recherche fortement recommandé dans les domaines du web sémantique, de l'intelligence artificielle et de la littérature linguistique [Dennai et Benslimane, 2011].

A.II.2.1. Similarité syntaxique

Une mesure de similarité syntaxique permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme très proches, alors que "voiture" et "automobile" pourront être considérées comme très différentes.

Dans cette section, nous présentons les mesures de similarité syntaxique les plus utilisées, en passant par la représentation vectorielle d'un document.

A.II.2.1.1. Modèle d'espace vectoriel

Afin de réduire la complexité des documents et de faciliter leur manipulation, il faut transformer chaque document, i.e. sa version textuelle intégrale, en un vecteur qui décrit le contenu du document. La représentation d'un ensemble de documents sous forme de vecteurs dans un espace vectoriel commun est connue sous le nom de modèle d'espace vectoriel (VSM : Vector Space Model) [Elsa, 2013].

En recherche d'information, dans un modèle d'espace vectoriel, les documents sont représentés comme des vecteurs de caractéristiques représentant les termes qui apparaissent dans la collection. On parle aussi de "sacs de mots" où les mots sont considérés comme indépendants et où l'ordre est sans importance. La valeur de chaque caractéristique est appelée le poids du terme et est en général une fonction de fréquence de termes dans le document. Par conséquent, en utilisant la fréquence de chaque terme comme un poids, les termes qui apparaissent le plus fréquemment sont plus importants et donc descriptifs du document [Elsa, 2013].

La représentation d'un document sous forme vectorielle se déroule en deux étapes : [Elsa, 2013]

- **Extraction des termes pertinents du document.**

Il s'agit de prétraiter le texte des documents textuels en supprimant les mots-vides[3], la ponctuation et les éventuels retours-chariots, de lemmatiser[4] le texte et de le segmenter.

- **Calcul du poids des termes.**

Le poids de chaque terme dans un document peut être obtenu de différentes manières : booléenne, fréquence des termes, *TF-IDF*, *TF-ITDF*, etc.

- **Méthode booléenne.**

De manière booléenne, si un terme existe dans un document alors la valeur qui lui correspond vaut 1, sinon 0. L'approche booléenne est utilisée lorsque chaque terme est d'égale importance et s'emploie uniquement lorsque les documents sont de petites tailles.

- **Fréquence des termes.**

La fréquence des termes est obtenue en comptant les occurrences du terme dans le document, c'est ce qu'on appelle le poids des termes : $TF_{i,j}$ représente donc la fréquence du terme *(Term Frequency)* i dans le document j.

- **La formule TF-IDF.**

Une technique répandue et efficace pour associer un poids à des termes consiste à mesurer leur rareté, telle que définie par la mesure *TF-IDF*. La mesure de la rareté d'un terme se calcule comme suit : [Olivier, 2013]

Supposons un ensemble de documents *D* et la fréquence d'un terme *i* dans un document *j*. La mesure de la rareté du terme sera déterminée par sa fréquence inverse dans l'ensemble de documents *D : IDF*. L'équation suivante représente ce calcul : [Olivier, 2013]

$$\boldsymbol{IDF(t,D) = \log\left(\frac{|D|}{|\{d \epsilon D : t \epsilon D\}|}\right)} \qquad \textbf{(1)}$$

$\boldsymbol{IDF(t,D)}$: Valeur IDF du terme t.

$\boldsymbol{D}$: Nombre de documents dans l'ensemble D.

$\boldsymbol{d \in D : t \in d}$: Nombre de documents de l'ensemble D contenant le terme t.

Par exemple, si un terme se retrouve dans tous les documents, alors la valeur *IDF* de ce terme sera égale à *log (1)* donc *0*. À l'inverse, si un terme ne se trouve que dans un seul document, sa valeur *IDF* sera égale à *log (|D|)*. Ce qui produit l'effet recherché pour amoindrir la valeur de mots qui n'ont pas de pouvoir de discernement entre les documents de l'ensemble *D*.

[3] Pour, le, la, les, l',..., un, une, des, ..., y, à, se, s', les dérivés du verbe être, les dérivés du verbe avoir, mon, ton, son, mes, tes, ..., qu', que, qui, ... sont considérés des mots vides.

[4] La lemmatisation du contenu d'un texte permet de regrouper les mots d'une même famille. Chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme (forme canonique). La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit : Le nom, le pluriel, le verbe à l'infinitif, ...

Pour la valeur *TF*, il suffit de compter la fréquence d'un terme *t* dans un document *d*. Une fois ces valeurs calculées, on produit l'équation suivante qui constitue la base de la mesure *TF-IDF* :

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \qquad (2)$$

Le produit final donne la valeur *TF-IDF* d'un terme *t* par rapport à un document particulier *d* dans un ensemble de documents *D*. Une fois ces valeurs obtenues, il faut déterminer quels documents sont les plus pertinents pour une requête donnée de l'utilisateur. L'utilisation des espaces vectoriels est une façon de calculer la corrélation entre la requête et chacun des documents de l'ensemble D [Olivier, 2013].

- **La formule TF-ITDF.**

Cette formule s'inspire de la formule *TF-IDF* et qu'on applique aux balises. Ainsi, la définition de *TF-ITDF* est donnée de la manière suivante : [Zargayouna et Saloti, 2004]

$$TF - ITDF(t, b, d) = TF(t, b, d) \times ITF(t, d) \times IDF(t, b) \qquad (3)$$

$$ITF(t, d) = \log\left(\frac{|D|_b}{DF(t,b)}\right) \qquad (4)$$

$|D|_b$: Nombre total de documents où le modèle de balise *b* est présent dans leur structure.

$DF(t, b)$: *(Document Frequency)*, nombre de documents qui contiennent la balise *b* et dans laquelle le terme *t* apparaît au moins une fois.

$$IDF(t, b) = \log\left(\frac{|B|_d}{TagF(t,b)}\right) \qquad (5)$$

$|B|_d$: Nombre total des balises dans le document *d*.

$TagF(t, b)$: *(Tag Frequency)*, Nombre de balises dans le document *d* et dont le terme *t* apparaît au moins une fois.

Pour la valeur *TF (t,b,d)*, il suffit de compter la fréquence d'un terme *t* dans une balise *b* dans un document *d*.

A.II.2.1.2. Quelques techniques de mesure de similarité syntaxique

Une mesure de similarité est, en général, une fonction qui quantifie le rapport entre deux objets, comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés doivent être de même domaine.

Toutes les mesures de similarité ne sont pas des métriques. Pour être une métrique, une mesure de distance *d* doit satisfaire les quatre conditions suivantes : [Elsa, 2013]

Soit *x*, *y* et *z* trois éléments d'un ensemble, et soit *d(x, y)* la distance entre x et y.

- Positivité : $d(x, y) \geq 0$.
- Principe d'identité des indiscernables : $d(x, y) = 0 \equiv x = y$.
- Symétrie : $d(x, y) = d(y, x)$.
- Inégalité triangulaire : $d(x, z) \leq d(x, y) + d(y, z)$.

- **Mesure Cosinus.**

Elle utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots) ; deux objets (documents : d_1 et d_2) sont similaires si leurs vecteurs sont confondus ; si deux objets ne sont pas similaires, leurs vecteurs forment un angle *(X, Y)* dont le cosinus représente la valeur de la similarité. Elle quantifie donc la similarité entre les deux vecteurs comme le cosinus de l'angle entre les deux vecteurs [Dennai et Benslimane, 2011 ; Slimani et al., 2008].

La formule est définie par le rapport du produit scalaire des vecteurs d_1 et d_2 et le produit de la norme de d_1 et de d_2, $Sim_{Cosinus} \in [0, 1]$ [Baeza-Yates et Ribeiro-Neto, 1999].

$$Sim_{Cosinus}(d_1, d_2) = \frac{\overrightarrow{d_1} \times \overrightarrow{d_2}}{\|\overrightarrow{d_1}\| \times \|\overrightarrow{d_2}\|} \qquad (6)$$

- **Coefficient de corrélation de Pearson.**

Le coefficient de corrélation de Pearson calcule la similarité entre deux documents d_1 et d_2 comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue $Sim_{Pearson}(d_1, d_2) \in [0, 1]$ [Elsa, 2013].

$$Sim_{Pearson}(d_1, d_2) = Sim_{Cosinus}(d_1 - \overline{d_1},\ d_2 - \overline{d_2}) \qquad (7)$$

Où $\overline{d_1}$ (respectivement $\overline{d_2}$) représente la moyenne de d_1 (respectivement d_2).

- **Distance Euclidienne.**

La distance euclidienne calcule la similarité entre deux documents d_1 et d_2 comme la distance entre leurs représentations vectorielles ramenées à un seul point [Slimani et al., 2008].

$$Sim_{Euclidienne}(d_1, d_2) = \left\|\overrightarrow{d_1} - \overrightarrow{d_2}\right\| = \sqrt{\sum_{i=1}^{n}(d_{1i} - d_{2i})^2} \qquad (8)$$

Où n est le nombre total de termes représentés, i.e. la taille des vecteurs.

- **Coefficient de Jaccard.**

L'indice de Jaccard ou coefficient de Jaccard [Jaccard, 1901] est le rapport entre la cardinalité (La taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents d_1 et d_2 sont donc représentés, non pas comme des vecteurs, mais comme des ensembles de termes. La similarité obtenue $Sim_{jaccard}(d_1, d_2) \in [0, 1]$ [Dennai et Benslimane, 2011 ; Dennai et Benslimane, 2013].

$$Sim_{Jaccard}(d1, d2) = \frac{\|d_1 \cap d_2\|}{\|d_1 \cup d_2\|} \qquad (9)$$

Il est aussi possible d'utiliser la représentation vectorielle :

$$Sim_{Jaccard}(d1, d2) = \frac{\overrightarrow{d_1}.\overrightarrow{d_2}}{\|\overrightarrow{d_1}\|.\|\overrightarrow{d_2}\| - \overrightarrow{d_1}.\overrightarrow{d_2}} \qquad (10)$$

- **Distance (d'édition) de Levenshtein.**

La distance de Levenshtein [Levenshtein, 1966] calcule la similarité entre les représentations sous forme de chaines de caractères des documents d_1 et d_2. Il s'agit du coût minimal, i.e. du nombre minimal d'opérations d'édition, pour transformer d_1 en d_2.

Les opérations sont les suivantes :

- Substitution d'un caractère de d_1 en un caractère de d_2,
- Ajout dans d_1 d'un caractère de d_2,
- Suppression d'un caractère de d_1.

Pour obtenir la distance de Levenshtein $Sim_{levenshtein}(d_1, d_2)$ entre les documents d_1 et d_2, il s'agit d'associer à chacune de ces opérations un coût. Le coût des opérations est toujours égal à 1, sauf dans le cas d'une substitution de caractères identiques. Notons que cette distance a été étendue pour prendre en compte la grammaire, la phonétique, ...

La distance de Levenshtein est une mesure permettant l'appariement approximatif de chaines de caractères (approximate string matching) [Navarro, 2001].

- **Indice de Dice.**

L'indice de Dice mesure la similarité entre deux documents d_1 et d_2 en se basant sur le nombre de termes communs à d_1 et d_2 [Dice, 1945].

$$Sim_{Dice}(d1, d2) = \frac{2N_c}{N_1 + N_2} \qquad (11)$$

Où N_c est le nombre de termes communs à d_1 et d_2 et N_1 (respectivement N_2) est le nombre de termes de d_1 (Respectivement d_2).

A.II.2.1.3. Synthèse

- Les techniques basées sur l'approche syntaxique sont facilement automatisables.
- Les techniques basées sur le modèle vectoriel ont le même format initial et sont faciles à développer, il s'agit uniquement de calcul vectoriel.
- La lemmatisation, l'élimination des mots-vides et l'application de *TF-IDF* et de *TF-ITDF* permettent de pallier au problème des mots identiques considérés comme peu pertinents qui peuvent parfois trop influer sur la valeur de la similarité.
- Par définition, les techniques basées sur l'approche syntaxique ne prennent pas en compte la sémantique. Par exemple, il est difficile de trouver une forte similarité entre "Je possède un ordinateur" et "J'ai un ordinateur". Par conséquent, la prise en compte de la sémantique semble importante.

[Huang, 2008] et [Strehl et al., 2000] ont tous les deux montré que les performances de la similarité cosinus, du coefficient de Jaccard et du coefficient de Pearson sont très proches et qu'elles sont significativement meilleures que celles de la distance euclidienne.

Cependant, [Bavi et al., 2010] fait apparaitre que plus le document est de petite taille, meilleurs sont les résultats obtenus avec la distance euclidienne, tandis qu'ils sont plus mauvais avec la similarité cosinus ou avec le coefficient de Jaccard.

Sachant que l'indice de Dice est en fonction du coefficient de Jaccard[5], nous pouvons penser qu'ils ont des performances similaires. La distance de Levenshtein est largement utilisée en linguistique et en bioinformatique ainsi que pour la reconnaissance de blocs de textes contenants des erreurs isolées. Malheureusement, le temps de calcul (complexité), lorsqu'on l'applique à deux séquences d'approximativement la même taille n, est $O(n^2)$. Cela est un obstacle dans de nombreuses applications pratiques [Baake et al., 2006].

A.II.2.2. Similarité sémantique

Une mesure de similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique [Elsa, 2013].

Il est à noter que la distance sémantique peut être de deux sortes : La similarité sémantique et la parenté sémantique. La première est un sous-ensemble de la seconde, mais les deux termes peuvent être utilisés indifféremment dans certains contextes, ce qui rend encore plus important d'être conscient de leur distinction [Elsa, 2013].

[5] Soit D l'indice de Dice et J le coefficient de Jaccard, nous avons $D = \frac{2J}{1+J}$

Deux concepts sont considérés comme sémantiquement similaires s'il y a une synonymie, hyponymie[6], antonymie, ou troponymie[7] entre eux (Exemples : MEDECIN-CHIRURGIEN, SOMBRE-CLAIR).

Deux sens de mots sont considérés comme sémantiquement liés s'il existe au moins une relation lexico-sémantique entre eux, classique ou non classique (Exemples : CHIRURGIEN-SCALPEL, ARBRE-OMBRE) [Mohammad et Hirst, 2012].

Les mesures de similarité de textes ont été utilisées dans de nombreux domaines. Par exemple, pour la classification de textes [Rocchio, 1971], la désambiguïsation du sens des mots [Lesk, 1986], la traduction automatique [Papineni et al., 2002]...

À quelques exceptions près, l'approche classique pour trouver la similarité entre deux segments de texte, est d'utiliser une méthode simple de concordance lexicale et de calculer un score de similarité basé sur le nombre d'unités lexicales qui se produisent dans les deux segments. Des améliorations ont été apportées à cette méthode simple qui consiste à retirer les mots-vides, à ne considérer que la plus longue sous-suite, ou encore à pondérer ou normaliser [Salton, 1997]. Ces méthodes de similarité lexicale ne peuvent pas toujours identifier la similarité sémantique des textes. Par exemple, il y a une similitude évidente entre les segments de texte “Je possède un ordinateur” et “J'ai un ordinateur”, mais la plupart des mesures de similarité textuelles vont échouer à identifier tout type de connexion entre ces textes [Elsa, 2013].

Il existe des mesures de similarité sémantique qui tentent de réussir en utilisant des approches qui sont, soit fondées sur la connaissance [Wu et Palmer, 1994 ; Leacock et Chodorow, 1998] ; ... ou sur un corpus [Deerwester et al., 1990], ... Ces mesures ont été appliquées avec succès à des tâches de traitement du langage comme, par exemple, la désambiguïsation du sens des mots [Patwardhan et al., 2003].

A.II.2.2.1. Quelques techniques de mesure de similarité sémantique

Dans cette section, nous dissocions les approches vectorielles, les approches topologiques et les approches statistiques.

[6] Relation sémantique hiérarchique d'un lexème à un autre selon laquelle l'extension du premier terme est incluse dans l'extension du second. Le premier terme est dit hyponyme de l'autre. Haut-de-forme est un hyponyme de chapeau et chapeau est un hyponyme de coiffure.

[7] Relation sémantique entre deux verbes, l'un décrivant de manière plus précise l'action de l'autre. Le premier verbe est dit troponyme du second.

A.II.2.2.1.1. Approches vectorielles (Vector-based)

- **Vecteurs sémantiques.**

L'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Une manière simple de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de similarité vectorielles (Comme pour la similarité syntaxique). Le plus difficile est d'obtenir de tels vecteurs. Il faut donc construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé. Les vecteurs sont définis dans un espace vectoriel orthogonal à *n* dimensions où chaque base se voit attribuer un mot de vocabulaire unique (Donc, chaque entrée du dictionnaire a une base dans l'espace vectoriel). Pour chaque mot du dictionnaire, on détermine un vecteur dans cet espace où la composante du vecteur pour chaque base est le nombre d'occurrences du mot dans la base qui le représente où il apparaît dans le contexte du mot pour lequel un vecteur a été construit [Elsa, 2013].

- **Bi-clustering.**

La classification double ou co-clustering ou bi-clustering est une technique d'exploration de données non supervisée permettant de segmenter simultanément les lignes et les colonnes d'une matrice. Étant donné un ensemble de L lignes à C colonnes (i.e. une matrice $L \times C$), l'algorithme de bi-clustering génère de bi-clusters : Un sous-ensemble de lignes qui présentent un comportement similaire sur un sous-ensemble de colonnes, ou vice versa [Elsa, 2013].

Le bi-clustering est utilisé dans le domaine de la fouille de texte où il est populairement connu en tant que co-clustering [Bisson et Hussain, 2008]. Les corpus de textes sont représentés sous une forme vectorielle : Comme une matrice D dont les lignes sont les documents et les colonnes sont les mots du dictionnaire. Les éléments D_{ij} de la matrice désignent l'occurrence du mot j dans le document i.

Les algorithmes de bi-clustering sont, ensuite, appliqués pour découvrir des blocs dans D qui correspondent à un groupe de documents (Lignes) caractérisé par un groupe de mots (Colonnes).

[Bisson et Hussain, 2008] ont proposé une approche qui utilise la similarité entre les mots et la similarité entre les documents pour segmenter la matrice. Leur méthode (Connue sous le nom χ-*Sim*, pour similarité croisée) est basée sur la recherche de similarité document-document et de similarité mot - mot, puis utilise les méthodes classiques de classification. Au lieu de regrouper explicitement les lignes et les colonnes alternativement, les auteurs considèrent des occurrences de mots d'ordre supérieur, i.e., en tenant compte des documents dans lesquels ils apparaissent.

Ainsi, la similarité entre deux mots est calculée sur la base des documents dans lesquels ils apparaissent ainsi que des documents dans lesquels des mots similaires apparaissent [Elsa, 2013].

L'idée est que deux documents sur le même sujet ne contiennent pas nécessairement le même jeu de mots, mais un sous-ensemble des mots et d'autres mots similaires qui sont caractéristiques de ce sujet.

Cette approche d'utilisation de similarités d'ordre supérieur prend en considération la structure sémantique latente de l'ensemble du corpus et génère ainsi une meilleure classification des documents et des mots.

Plus formellement, à partir de la matrice documents/termes *D* (Où la vectrice ligne $d_{i:}$ de taille *C* décrit le document *i* et la vectrice colonne $d_{:j}$ de taille *L* décrit le mot *j*), il s'agit de déterminer les matrices *SR* (Similarity Row : Matrice de similarité pour les documents) et *SC* (Similarity Column : Matrice de similarité pour les termes). Classiquement, la similarité entre deux documents est une fonction sur les termes communs. Ainsi, on a l'équation : [Elsa, 2013]

$$Sim\ (d_{i:}; d_{:j}) = F_s\ (d_{i1}; d_{j1}) + \ldots + F_s\ (d_{ic}; d_{jc}). \qquad (12)$$

Où F_s est une fonction de similarité. Initialement, on suppose que la matrice *SC* est initialisée à 1, i.e., $sc_{i;i} = 1$.

L'équation devient donc :

$$Sim\ (d_i; d_j) = F_s(d_{i1}; d_{j1}).sc_{11} + \ldots + F_s(d_{ic}; d_{jc}).sc_{cc}. \qquad (13)$$

Cette équation est ensuite généralisée pour prendre en compte toutes les paires de mots possibles. Réciproquement, la similarité entre deux mots est fonction des documents communs dans lesquels ils apparaissent. Ainsi, les équations pour calculer les similarités sont dépendantes : Pour obtenir SR et SC, il faut utiliser itérativement et alternativement chaque équation et mettre à jour les valeurs pour s'en servir dans les itérations suivantes.

A.II.2.2.1.2. Approches topologiques (Knowledge-based)

Les approches de similarité de mots basées sur la connaissance s'appuient sur un réseau sémantique de mots, tel que WordNet [Fellbaum, 1998]. Etant donnés deux mots, leur similarité peut être estimée à partir de leurs positions relatives dans la hiérarchie de la base de connaissances.

En effet, la structure de la base est un arbre où chaque nœud est un concept (Par exemple, un ordinateur), ses enfants sont les hyponymes du concept (i.e., “X” est un hyponyme de “Y” si “X est un Y” est vrai) et ses parents sont ses hyperonymes (i.e., “X” est un hyperonyme de “Y” si “Y est un X” est vrai). Les concepts peuvent être des noms, des verbes ou des adjectifs [Elsa, 2013].

Les mots ont des “synsets”, qui sont des ensembles de concepts pour lesquels le mot peut correspondre (i.e., les concepts desquels le mot peut être synonyme). Enfin, il faut noter que les concepts sont de plus en plus abstraits et généraux lorsqu'on va vers la racine et qu'ils sont plus spécifiques lorsqu'on va vers les feuilles [Elsa, 2013].

Wordnet est une base de connaissances ou taxonomie dont les concepts sont en anglais. Cependant, une base similaire a été créée pour la langue française : WOLF (WordNet Libre du Français) [Sagot et Fišer, 2008].

- **Techniques basées sur les arcs (Edge – based).**

L'approche basée sur les arcs est une manière naturelle et directe d'évaluer la similarité sémantique dans une taxonomie. Il s'agit d'estimer la distance (e.g. longueur des arcs) entre les nœuds correspondants aux concepts / classes à comparer. Compte tenu de l'espace multidimensionnel des concepts, la distance conceptuelle peut facilement être mesurée par la distance géométrique entre les nœuds représentants les concepts. Évidemment, plus le chemin d'un nœud à l'autre est court, plus ils sont similaires.

- **Mesure de Wu et Palmer.**

Elle a été utilisée par [Halkidi et al., 2003] pour organiser des documents web dans des clusters. Elle a aussi servi dans [Desmontils et Jacquin, 2001] pour évaluer la proximité sémantique de deux concepts d’une page HTML relativement à un thésaurus dans le cadre d’une indexation d’un site web par des ontologies [Dennai et Benslimane, 2013].

[Lin, 1998] a effectué une comparaison entre les méthodes de mesure de similarité, il en ressort que la mesure de [Wu et Palmer, 1994] a l’avantage d’être simple à calculer en plus des performances qu’elle présente, tout en restant aussi expressive que les autres.

Etant donnée une ontologie formée par un ensemble de nœuds et un nœud racine C_R (Root) (cf. figure 7). Soit C_1 et C_2 deux éléments de l’ontologie dont nous allons calculer la similarité. Le principe de calcul de similarité est basé sur les distances (D_1+D et D_2+D) qui séparent les nœuds C_1 et C_2 du nœud C_R et la distance (D) qui sépare le concept subsumant ou le PPG de C_1 et de C_2 du nœud C_R [Dennai et Benslimane, 2013].

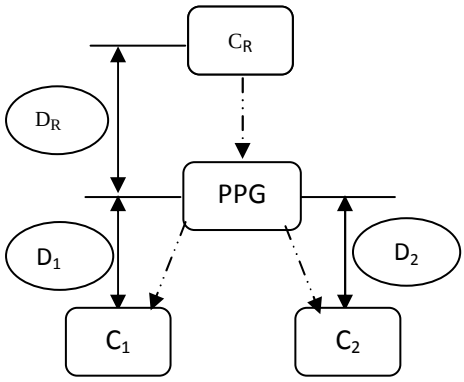

Fig. 7. Exemple d'un extrait d'une ontologie.

La mesure de Wu et Palmer est définie par la formule suivante :

$$Sim_{Wu\&Palmer}(C_1, C_2) = \frac{2 \times D_R}{D_1 + D_2 + 2 \times D_R} \qquad (14)$$

- **Mesure de Rada et al.**

Cette mesure est adoptée dans un réseau sémantique, elle est fondée sur le fait qu'on peut calculer la similarité en se basant sur les liens hiérarchiques "Is-a". Pour calculer la similarité de deux concepts dans une ontologie, on doit calculer le nombre des arcs minimums qui les séparent. Cette mesure, basée sur le calcul de la distance entre les nœuds par le chemin le plus court, présente un moyen pour évaluer la similarité sémantique [Rada et al., 1989].

Dans le domaine biomédical, [Pedersen et al., 2007] ont proposé la première mesure de similarité sémantique en utilisant la longueur de chemin entre les termes biomédicales dans l'ontologie MeSH[8] comme mesure sémantique [Dennai et Benslimane, 2013].

- **Mesure d'Ehrig et al.**

Ce travail introduit trois couches : Les données, l'ontologie et le contexte. La similarité des entités est mesurée au niveau des données en considérant les valeurs de données de type simple ou complexe (Entier, caractère). Les relations sémantiques entre les entités sont mesurées au niveau de la couche de l'ontologie. Finalement la couche du contexte spécifie comment les entités de l'ontologie sont utilisées dans un certain contexte externe, plus spécifiquement, le contexte de l'application [Ehrig et al., 2004].

- **Techniques basées sur les nœuds (Node-based ou information content-based).**

Une approche basée sur les nœuds pour déterminer la similarité conceptuelle est appelée une approche information content-based [Resnik, 1995].

[8]Medical Subject Heading : Http://www.nlm.nih.gov/mesh/

Étant donné un espace multidimensionnel où un nœud représente un concept unique composé d'un certain nombre d'informations et où un arc représente une association directe entre deux concepts, la similarité entre deux concepts est la mesure dans laquelle ils partagent des informations en commun. Compte tenu de cette notion de structure hiérarchique / espace de classes, ces informations communes peuvent être identifiées comme un nœud/concept spécifique qui englobe les deux dans la hiérarchie. Plus précisément, cette superclasse devrait être la première classe en haut de la hiérarchie qui englobe les deux classes. La valeur de similarité est définie comme la valeur du contenu de l'information de cette superclasse. La valeur du contenu de l'information d'une classe est ensuite obtenue en estimant la probabilité d'occurrence de cette classe dans un grand corpus de texte. Le contenu de l'information (IC) d'un concept / d'une classe *c* est :

$$IC(c) = -Log(P(c)) \qquad (15)$$

Où *P(c)* est la probabilité de rencontrer une instance du concept c.

- **Mesure de Resnik.**

La notion du Contenu Informationnel (IC) a été initialement introduite par Resnik. Il a prouvé qu'un objet (Mot) est défini par le nombre des classes spécifiées et que la similarité sémantique, entre deux concepts, est mesuré par la quantité d'information qu'ils partagent [Resnik, 1999].

Pour évaluer la pertinence d'un objet, il faut calculer le contenu informationnel. Le contenu informationnel est obtenu en calculant la fréquence de l'objet dans le corpus (En utilisant Wordnet par exemple) [Dennai et Benslimane, 2011].

$$Sim_{Resnik}(C_1, C_2) = Max[E(cs(c_1, c_2))] = Max\left[-log\left(P(cs(c_1, c_2))\right)\right] \qquad (16)$$

Où CS (C1, C2) représente le concept le plus spécifique (Qui maximise la valeur de similarité) qui subsume (Situé à un niveau hiérarchique plus élevé) les deux concepts C_1 et $C2$ dans l'ontologie. Cette mesure est un peu sommaire car elle ne dépend que du concept le plus spécifique.

- **Mesure de Lin.**

Cette mesure utilise une approche hybride qui combine deux sources de connaissances différentes (Thesaurus, corpus). En plus, elle représente la similarité comme degré probabiliste de chevauchement des concepts descendants de C_1 et C_2.

Les travaux de [Miller et al., 1993] ont évalué cette mesure [Lin, 1998] à travers une expérience qui utilise des sujets humains pour évaluer la similarité entre 30 paires de noms, il en ressort que cette méthode offre une amélioration significative [Dennai et Benslimane, 2011].

$$Sim_{Lin}(C_1, C_2) = \frac{2 \times log\left(P(AC(C_1,C_2))\right)}{Log(P(C_1)) + log(P(C_2))} \qquad (17)$$

- **Mesure de Hirst et St-Onge.**

L'idée de cette mesure [Hirst et St-Onge, 1998] est que deux concepts lexicalisés sont sémantiquement étroits si leurs ensembles synonymes (Synsets) de WordNet sont reliés par un chemin qui n'est pas trop long et qui ne change pas la direction trop souvent. Avec cette mesure, toutes les relations contenues dans un réseau WordNet sont prises en considération. Les auteurs de cette mesure ont classé la direction des liens en lien haut (Superclasse), lien bas (Sous-classe) et lien horizontal (Antonyme). Le calcul de la similarité, avec cette méthode, s'effectue entre objets (Mots) par le poids du plus court chemin allant d'un terme à un autre, en plus des classifications qui indiquent les changements de direction. [Zargayouna et Saloti, 2004 ; Slimani et al., 2008].

$$Sim_{Hirst} = T - SPD - K \times nd \qquad (18)$$

Où *T* et *K* sont des constantes, *SPD* (Shortest Path Distance) est la distance du plus court chemin en nombre d'arc et *nd* le nombre de changements de direction.

- **Techniques hybrides.**

Ces techniques sont fondées sur un modèle qui combine entre les approches basées sur les arcs (Distances) et le contenu informationnel qui est considéré comme facteur de décision. Parmi les travaux se basant sur cette technique, on peut citer : [Dennai et Benslimane, 2013]

- **Mesure de Jiang et Conrath.**

Pour remédier au problème présenté au niveau de la mesure de Resnik, Les auteurs de cette mesure [Jiang et conrath, 1997] ont apporté une nouvelle formule qui consiste à combiner l'entropie (Contenu Informationnel) du concept spécifique à ceux des concepts dont on cherche la similarité (combine entre les techniques basées sur les arcs et les techniques basées sur les nœuds qui consistent à compter les arcs afin d'améliorer les résultats par des calculs basés sur les nœuds).

La mesure adoptant cette méthode est basée sur la combinaison d'une source de connaissance riche (thesaurus) avec une source de connaissance pauvre (corpus) [Slimani et al., 2008].

$$Sim_{Jiang\ \&\ Conrath}(C_1, C_2) = \frac{1}{Distance(C_1, C_2)} \qquad (19)$$

Sachant que la distance entre C_1 et C_2 est calculée en utilisant la formule suivante:

$$Distance(C_1, C_2) = E(C_1) + E(C_2) - \left(2 \times E\left(CS(C_1, C_2)\right)\right) \qquad (20)$$

- **Mesure de Leacock et Chodorow.**

Les auteurs de cette mesure présentent une méthode qui combine entre la méthode de comptage des arcs et la méthode du contenu informationnel. La mesure, proposée par Leacock et Chodorow, est basée sur la longueur du plus court chemin entre deux synsets de WordNet.

Les auteurs ont limité leur attention à des liens hiérarchiques «Is-a» ainsi que la longueur de chemin par la profondeur globale P de la taxinomie [Leacock et Chodorow, 1998].

$$Sim_{Leacock\ \&\ Chodorow}(C_1, C_2) = -\log\left(\frac{cd(C_1, C_2)}{2 \times M}\right) \qquad (21)$$

Où *M* est la longueur du chemin le plus long qui sépare le concept racine, de l'ontologie, du concept le plus en bas. On dénote par *cd* (C_1, C_2) la longueur du chemin le plus court qui sépare C_1 de C_2.

A.II.2.2.1.3. Approches statistiques (Corpus-based)

Les mesures basées sur des corpus diffèrent des mesures présentées précédemment car elles ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte. Parmi de telles mesures de similarité sémantique, nous présentons l'analyse sémantique latente (LSA : Latent Semantic Analysis) [Deerwester et al., 1990]. Dans LSA, les occurrences des termes dans un corpus sont capturées au moyen d'une réduction de dimension réalisée par une décomposition en valeurs singulières (SVD : Singular Value Decomposition) sur la matrice termes/documents représentant le corpus ; l'analyse sémantique explicite (ESA : Explicit Semantic Analysis) [Gabrilovich et Markovitch, 2007] qui est une variation du modèle standard vectoriel où les dimensions du vecteur sont directement équivalentes à des concepts abstraits. D'autres mesures comme la distance normalisée de Google (NGD : Normalized Google Distance) [Cilibrasi et Vitanyi, 2007] et le n° de wikipédia (no. of Wikipedia (noW)) [Wong et al., 2006].

- **Mesure LSA / PLSA et LDA**.

[Deerwester et al., 1990] propose l'analyse sémantique latente (LSA), qui peut être utilisée pour déterminer la distance entre des mots ou entre des ensembles de mots.

Contrairement aux diverses approches décrites précédemment où une matrice de cooccurrence mots/mots est créée, la première étape de la LSA consiste à créer des matrices mots/paragraphes, mots/documents ou mots/passages, où un passage est un groupe de mots.

Une cellule pour un mot *w* et un passage *p* est, par exemple, rempli avec le nombre de fois qu'apparait *w* dans *p*. Ensuite, la dimension de cette matrice est réduite par l'application d'une décomposition en valeurs singulières (SVD), une technique de décomposition de matrice standard. Ce plus petit ensemble de dimensions représente un résumé (Inconnu) de concepts. Puis la matrice originale mot/passage est recréée, mais cette fois à partir des dimensions réduites.

Lorsque vous utilisez la modélisation latente, les documents d'une collection sont modélisés comme une combinaison pondérée des thèmes latents d'un ensemble $Z = \{z_1, \dots, z_{Nz}\}$. Dans l'analyse sémantique latente probabiliste (PLSA) [Hofmann, 1999], chaque thème latent possède un modèle de langage probabiliste $p(\omega||z)$ représentant la probabilité que le mot *w* puisse être généré par le thème *z*. Chaque document d_i de la collection de documents *D* est alors supposé avoir été généré par un mélange pondéré des modèles latents des thèmes.

Si un document est modélisé par une collection de mots $C = \{c_1, \dots, c_{Nv}\}$, le modèle génératif PLSA de *C* sachant d_i est :

$$\boldsymbol{P(C||d_i) = \prod_{\omega \in V}(\sum_{z \in Z} P(\omega||z)P(z||d_i))^{c_\omega}} \qquad \textbf{(22)}$$

L'analyse latente de Dirichlet (LDA : Latent Dirichlet analysis) [Blei et al., 2003], quant à elle, est une généralisation de PLSA dans laquelle l'estimation ponctuelle de $P(z||d_i)$ pour le document d_i dans PLSA est remplacée par une distribution probabiliste a priori de Dirichlet sur toutes les distributions possibles des thèmes latents au sein de *Z*.

Ces trois approches pâtissent de leur effet "boite noire".

- **Mesure ESA**.

L'analyse sémantique explicite (ESA) [Gabrilovich et Markovitch, 2007] est une représentation vectorielle de texte (Mots isolés ou documents) qui utilise Wikipédia comme une base de connaissances. Plus précisément, dans l'ESA, un mot est représenté par une vectrice colonne de la matrice TF-IDF du texte de l'article dans Wikipédia et un document (Chaîne de mots) est représenté comme le barycentre des vecteurs représentant ses mots.

L'ESA fait l'hypothèse que les articles de Wikipédia sont “orthogonaux”. Toutefois, il a été démontré que l'ESA améliore également les performances des systèmes de recherche d'information quand elle est fondée non pas sur Wikipédia, mais sur le corpus Reuters, qui ne satisfait pas la propriété d'orthogonalité.

A.II.2.2.2. Synthèse

- [Grefenstette, 2009] vante les mérites de l'approche basée sur les vecteurs sémantiques. D'après les auteurs, cette approche détecte aisément des documents similaires avec peu d'erreurs. [Bisson et Hussain, 2008] indique que le bi-clustering semble meilleur que le LSA.
- [Takale et Nandgaonkar, 2010 ; Corley et Mihalcea, 2005 ; Mihalcea et al., 2006] assument que les méthodes sémantiques ont des performances similaires, [Budanitsky et Hirst, 2006] a tendance à les ordonner en considérant que la méthode de Jiang et Conrath a des performances supérieures aux autres, suivi par celle de Lin et celle de Leacock et Chodorow puis par celle de Resnik et [Jiang et Conrath, 1997] indique que les méthodes basées sur les nœuds semblent avoir de meilleurs résultats que les méthodes basées sur les arcs. Il faut également noter que [Corley et Mihalcea, 2005] indique que les mesures de Jiang et Conrath, de Leacock et Chodorow, de Lin, de Wu et Palmer et de Resnik sont meilleures que les approches vectorielles.
- [Hazen, 2010] a montré que, dans certains cas, le LDA a des performances très décevantes par rapport aux approches vectorielles basées sur le TF-IDF. Les résultats de [Mohler et Mihalcea, 2009] indiquent que les résultats obtenus avec les mesures basées sur la connaissance (Approches topologiques) et celles basées sur un corpus (LSA et ESA) ont des performances comparables. L'avantage des approches basées sur un corpus par rapport à celles basées sur la connaissance réside dans leur indépendance de la langue et à leur facilité à créer des corpus spécifiques à un domaine contrairement à une taxonomie comme WordNet.
- Les approches sémantiques basées sur les corpus ou la connaissance posent des problèmes de stockage et de complexité et sont souvent spécifiques à un domaine donné.

A.II.2.3. Récapitulation

Mesure	Approche syntaxique			Approche sémantique				
	Espace Vectoriel	Edition	Termes Communs	Espace Vectoriel	A base d'Arcs	A base de Nœuds	Hybride : Arcs et Nœuds	A base de Corpus
Cosinus	■							
Coefficient de corrélation de Pearson	■							
Euclidienne	■							
Coefficient de Jaccard	■							
Levenshtein		■						
Indice de Dice			■					
Vecteurs sémantiques				■				
Bi-clustering				■				
Wu et Palmer					■			
Rada et al.					■			
Ehrig et al.					■			
Resnik						■		
Lin						■		
Hirst et St-Onge						■		
Jiang et Conrath							■	
Leacock et Chodorow							■	
LSA / PLSA / LDA								■
ESA								■

Tab. 1. Récapitulation des mesures de similarité.

A.II.2.4. Similarité sémantique ontologique

L'identification de la similarité entre les concepts d'une ontologie est déterminante pendant les phases de l'identification et l'enrichissement dans une approche de rétro-ingénierie d'une application orientée web et qui a été adoptée par plusieurs techniques telles que le regroupement, la fouille de données, la sémantique et en particulier les systèmes de recherche de l'information. Cette dernière est largement liée à des mesures pour identifier la similarité entre les documents [Baeza-Yates et Ribeiro-Neto, 1999 ; Salton et McGill, 1983].

La présence ou non de mots-clés de l'ontologie dans un document donne de bonnes informations sur le contexte du document, mais cette présence a des effets très variables sur la pertinence du document par rapport à un besoin d'information de l'utilisateur.

Une ontologie peut couvrir un domaine très vaste de connaissances mais le besoin de l'utilisateur est souvent très précis, touchant une infime partie des connaissances de l'ontologie. Pour calculer cette pertinence avec l'ontologie, il faut vérifier si le document contient d'autres termes de l'ontologie pour en déterminer la pertinence. Si c'est le cas, il faut vérifier quelle sorte de relation existe entre les termes trouvés dans le document [Olivier, 2013].

La similarité sémantique ontologique consiste en l'ordre des liens entre deux termes de l'ontologie. Une relation entre deux termes (Relation parent-enfant, une instance avec sa classe mère ou bien deux classes reliées par une relation) est d'ordre 1 alors qu'une relation fraternelle est d'ordre 2 entre deux instances avec la même classe mère.

Ci-après un exemple pris intégralement de [Olivier, 2013] qui explique d'une manière simple la notion de la similarité sémantique ontologique, sauf que l'auteur, ici, utilise l'appellation distance ontologique.

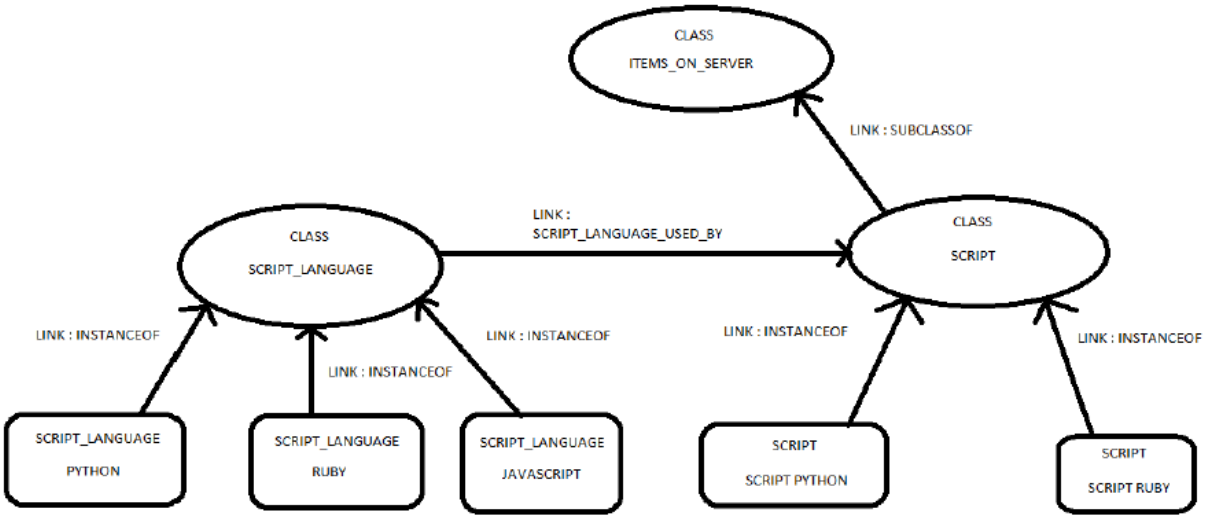

Fig. 8. Extrait d'une ontologie [Olivier, 2013].

Par exemple, si un document contient les mots « Script_Language », «Script », « Ruby », « HTML » et « Python » et que l'ontologie de la figure 8 est impliquée, on peut détecter que « Python » est une instance du concept « Script_Language ». Cela implique qu'une relation directe d'ordre 1 concerne ces deux termes. Donc, si un utilisateur fait une recherche sur l'un de ces deux termes, le document sera plus pertinent qu'un autre document n'ayant pas de relations directes entre les termes.

De la même façon, on peut détecter la relation directe entre les mots « Script » et « Script_Langage » par leur relation nommée « script_Language_used_By » représentée dans la figure 8. Il est tout à fait logique qu'un document ayant des termes liés par des relations soit plus pertinent pour les termes concernés.

Finalement, comme le terme « Ruby » est une instance de la classe « Script_Language », on peut détecter qu'une relation fraternelle (i.e. une relation enfant-parent-enfant) avec le terme « Python », une relation d'instance de classe avec « Script_language » et une relation d'ordre 2 avec « Script ».

Le résultat final est une sorte de toile liant les termes présents dans le document.

Pour la classe « Script_Language » : Présence de termes représentant la classe. Présence de termes représentant deux instances de la classe, soit « Python » et « Ruby ». Présence de termes représentant une classe reliée d'ordre 1 via la relation «Script_Language_Used_By». Présence du terme « HTML » représentant une instance avec une distance ontologique d'ordre 3 ou plus.

Pour la classe « Script » : Présence du terme représentant la classe. Présence de termes représentant des instances avec une distance ontologique d'ordre 2, soit « Python » et «Ruby». Présence de termes représentant la classe « Script_language » reliée d'ordre 1 via la relation « script_Language_used_By ». Présence du terme « HTML » représentant une instance avec une distance ontologique d'ordre 3 ou plus.

Pour l'instance « Python » : Présence du terme représentant la classe. Présence de termes représentant des instances liées par une relation fraternelle « Ruby ». Présence de termes représentant une classe avec une distance ontologique d'ordre 2 « Script ». Présence du terme « HTML » avec une distance ontologique d'ordre 3 ou plus.

Pour l'instance « Ruby » : Présence du terme représentant la classe. Présence de termes représentant des instances liées par une relation fraternelle « Python ». Présence de termes représentant une classe avec une distance ontologique d'ordre 2 « Script ». Présence du terme « HTML » représentant une instance avec une distance ontologique d'ordre 3 ou plus.

Pour l'instance « HTML » : Présence du terme représentant la classe. Présence des termes « Python », « Ruby », « Script_language », «Script » représentant des classes et des instances avec une distance ontologique d'ordre 3 ou plus.

Cela implique que pour chacun des 4 premiers termes, les liens avec trois autres termes augmentent le score de pertinence. Toutefois, la présence du terme «HTML», concept présent dans une ontologie du Web, n'augmentera pas de façon significative le score de pertinence des quatre autres termes car il n'est relié à aucun d'entre eux par une relation d'ordre 1 ou 2.

Les liens d'ordre 3 et plus sont jugés faibles dans le cadre du présent mémoire.

En abordant la situation de la façon inverse, un document qui contient le terme « HTML » une seule fois mais des dizaines de fois les termes reliés aux langages de script, le document n'aura probablement pas une forte pertinence pour quelqu'un recherchant des informations sur le langage HTML. L'information est considérée comme étant diluée par rapport aux autres concepts présents dans le document.

A.II.3. Conclusion

Les notions présentées dans ce chapitre ont permis d'attirer l'attention sur deux importantes parties dont on a besoin au cours de la réalisation de notre sujet de recherche :

- Une première partie a survolé la notion d'ontologie et tout en ce que la concerne ce qui nous a permis de choisir le type des ontologies du domaine, d'une part, elles permettent de décrire un domaine d'application et d'autre part, elles sont constituées de concepts et de propriétés de type déclaratif et en plus de tout ça, la plupart des ontologies existantes sont des ontologies du domaine. Elles régissent un ensemble de vocabulaires et de concepts qui décrivent un domaine d'application ou monde cible. L'ontologie du domaine est une méta-description d'une représentation des connaissances, c'est-à-dire une sorte de méta-modèle de connaissance dont les concepts et propriétés sont de type déclaratif. L'ontologie du domaine est à la base de notre indexation sémantique et cette dernière sera le cadre du prochain chapitre,
- Une deuxième partie a dévoilé des notions sur la similarité en général et en particulier les deux mesures de similarité syntaxique et sémantique entre termes (Concepts) et les différentes approches de calcul de cette similarité en prenant en considération la fréquence et le poids des termes ce qui nous a permis de se décider sur le choix de la mesure de Wu et Palmer qui est une mesure simple à implémenter et basée sur les arcs dans une ontologie hiérarchique.

Le prochain chapitre de cette partie dressera une revue de littérature sur les différentes techniques d'indexation et en particulier l'indexation sémantique et la notion de rétro-ingénierie.

CHAPITRE A.III.

Indexation Sémantique et Rétro-Ingénierie : Revue de Littérature

Sommaire

L'indexation sémantique est la base de notre rétro-ingénierie des applications web. Elle sera, d'une part, développée à notre manière et d'autre part utilisée dans notre approche proposée de rétro-ingénierie. L'approche d'indexation, que nous proposons, se base, particulièrement, sur l'indexation des pages HTML ou des documents XML pour lesquels nous avons une ontologie du même domaine. Cette dernière peut être construite à partir d'un corpus ou en utilisant différentes ressources. Le choix de traiter avec des corpus spécialisés, simplifie la tâche en limitant le vocabulaire, l'ambiguïté et la variabilité des formes syntaxiques.

Le présent chapitre est organisé en deux parties, la première partie dévoile des notions théoriques sur l'indexation en général, ses techniques y compris l'indexation sémantique et la deuxième partie est consacrée pour présenter la notion de rétro-ingénierie, sa position par rapport à la réingénierie des applications web.

A.III.1. L'indexation

A.III.1.1. Définition

Le but général de l'indexation est d'identifier l'information contenue dans tout texte et de la représenter au moyen d'un ensemble d'entités appelé index pour faciliter la comparaison entre la représentation d'un document et d'une requête. Plus exactement, le processus d'indexation est le transfert de l'information contenue dans le texte vers un autre espace de représentation traitable par un système informatique [Roussey et al., 1999].

L'utilisation des index remonte au XVème siècle peu après l'invention de l'imprimerie. Les index (Ou termes d'indexation) jouent un rôle important dans la recherche d'information dans la mesure où ils déterminent avec quels mots on peut retrouver un document [Nie, 2003].

L'indexation est une phase très importante pour un Système de Recherche d'Information (SRI), car de sa qualité dépend la qualité des réponses du système et donc les performances de ce dernier. Une bonne indexation doit permettre de retrouver tous les documents pertinents au besoin de l'utilisateur [Boubekeur, 2008].

FLUHER.C a défini l'indexation comme suit « Les documents sont lus par un documentaliste qui en déduit les thèmes principaux et les traduit en une liste de mots, dit descripteurs des documents. Cet ensemble de mots constitue l'index du document et représente la description du contenu sémantique de celui-ci » [Fluher, 1992].

En 1997, POM et SUTTER définissent l'indexation comme étant une opération ayant pour but de faciliter l'accès au contenu de documents ou d'un ensemble de documents à partir d'un sujet ou d'une combinaison de sujets ou toute autre entrée utile à la recherche [Pomart et Sutter, 1997].

A.III.1.2. Techniques d'indexation

L'indexation est la réduction du volume des données d'un document par le biais d'une représentation de ce document par des mots-clés. L'indexation peut être faite d'une manière manuelle (Indexation manuelle), automatique (Indexation automatique) et de façon assistée (Semi - automatique) ou bien par concepts (Indexation conceptuelle) [Hadj Henni, 2009].

A.III.1.2.1. Indexation manuelle.

Dans cette technique, c'est un opérateur humain, généralement expert du domaine, qui se charge de caractériser, selon ses connaissances propres, le contenu sémantique d'un document. Cette approche présente deux inconvénients :

1. Elle est subjective, puisque le choix des termes d'indexation dépend de l'indexeur et de ses connaissances du domaine,

2. Elle est pratiquement inapplicable aux corpus de textes volumineux.

Néanmoins, tel que rapporte dans [Savoy, 2005], elle est plus performante que l'indexation automatique en termes de précision moyenne des documents retrouvés en réponse à une requête utilisateur donnée [Boubekeur, 2008].

A.III.1.2.2. Indexation automatique

En indexation automatique [Luhn, 1957 ; Maron et Kuhns, 1960 ; Salton, 1968], le processus est complètement automatisé, il se charge d'extraire les termes caractéristiques du document.

L'intérêt d'une telle approche réside dans sa capacité à traiter les textes nettement plus rapidement que l'approche précédente, et de ce fait, elle est particulièrement adaptée aux corpus volumineux. L'indexation automatique est l'approche la plus étudiée en Recherche d'Information [Boubekeur, 2008].

A.III.1.2.3. Indexation semi-automatique

Les systèmes actuels remplacent l'homme pour une importante part de son expertise (Indexation semi-automatique) ; en fait, ils ne le remplacent pas complètement, car l'expression « Indexation automatique » suppose une intervention totale du système, ce qui est loin d'être le cas car l'intervention humaine est toujours nécessaire.

On peut citer comme exemple SINTEX et ALEXDOC comme logiciels d'indexation assistée par ordinateur [El Mabrouka, 2005].

A.III.1.2.4. Indexation conceptuelle

L'indexation conceptuelle se réfère à la construction de taxonomies conceptuelles à partir des textes. Cette approche est due à William Woods [William, 1997]. Le système conceptuel d'indexation et de recherche proposé extrait automatiquement des mots et des expressions de textes et les organise en un réseau sémantique (Taxonomie conceptuelle) qui intègre des relations syntaxiques, sémantiques et morphologiques. La construction d'une taxonomie de concepts à partir des textes est le plus souvent réalisée en parsant automatiquement chaque expression en une ou plusieurs structures conceptuelles qui représentent comment les éléments de l'expression sont réunis pour construire son sens [Boubekeur, 2008].

Remarque :

Notons que souvent l'indexation conceptuelle est définie comme une indexation sémantique puisque les concepts véhiculent la sémantique. Bien que nous adhérons à ce point de vue, nous avons choisi de suivre la classification donnée dans [Mihalcea et Moldovan, 2000] selon laquelle l'indexation conceptuelle réfère principalement à l'approche de Woods, tandis que toute indexation basée sur les sens des mots relève de l'indexation sémantique [Boubekeur, 2008].

A.III.1.2.5. Annotation

Dans le cas le plus général, annoter un document, c'est attacher à l'une de ses parties une description qui correspond à l'usage que l'on souhaitera en faire plus tard.

L'annotation savante est nécessaire au travail intellectuel sur les textes et ressort souvent du commentaire, de la mise en relation et de la construction d'un réseau d'intertextes [Hadj Henni, 2009].

A.III.1.2.6. Indexation sémantique

L'indexation sémantique a pour objectif la représentation des documents et des requêtes par le sens des termes (Ou des concepts) plutôt que par des mots de leur indexation classique. L'intérêt d'une telle approche est de lever l'ambiguïté des termes et de résoudre le problème de la divergence des termes.

A.III.1.3. Approche d'indexation sémantique

L'objectif des systèmes de recherche d'information (SRI) est de fournir aux utilisateurs les documents pertinents par rapport aux besoins qu'ils expriment. Les SRI utilisent des listes inversées qui rassemblent les différents termes d'indexation choisis pour représenter les contenus des documents et les liens vers ces documents. En complément, à chaque couple (Terme d'indexation, document) est associé un poids qui représente l'importance du terme dans un document.

Lorsqu'une requête est soumise au système, les termes qu'elle contient sont mis en correspondance avec les termes d'indexation extraits des documents pour en déduire les documents à restituer à l'utilisateur. La phase d'indexation est donc une phase primordiale dans le processus de recherche d'information.

Dans le chapitre B.I., un état de l'art des travaux similaires sur l'indexation sémantique, suivi par un tableau récapitulatif (Tableau n° 2) résumant le principe et les spécificités de chaque travail.

A.III.2. La rétro-ingénierie

A.III.2.1. Ingénierie, rétro-ingénierie et réingénierie

L'ingénierie ou plus précisément l'ingénierie directe (Forward Engineering) correspond au processus de développement traditionnel d'un système. On part d'un niveau de représentation conceptuelle élevé et on cherche à obtenir un système opérationnel par une série de développements/modifications progressifs descendants (Top-Down approach ou approches descendantes). Cette activité a pour objectif soit la mise au point d'un nouveau système (ingénierie de développement) soit l'évolution d'un système existant (Ingénierie de maintenance) [Benslimane, 2007].

La rétro-ingénierie ou ingénierie inverse (Reverse Engineering) consiste à repenser ce qui a été conçu dans une démarche d'ingénierie. En informatique, la rétro-ingénierie consiste à analyser un produit fini (Un système d'information, des processus, un logiciel ou des interfaces) pour déterminer la manière dont celui-ci a été conçu et identifier ses composants et leurs dépendances [Muller et al., 2000]. La rétro-ingénierie est le processus d'analyse d'un système permettant l'identification des entités et leurs corrélations en vue de passer d'une forme de représentation à une autre de niveau d'abstraction identique ou plus élevé. Cette activité consiste donc à examiner un existant, suivi éventuellement d'une remontée dans les niveaux d'abstraction, permettant ainsi une vue plus large du domaine et une meilleure compréhension du système d'information grâce à une re-documentation. Le processus de rétro-ingénierie comporte deux activités distinctes mais complémentaires :

- La re-documentation (Redocumentation) : Consiste à créer des représentations sémantiquement équivalentes et de même niveau d'abstraction. La re-documentation n'est pas une activité spécifique au processus de rétro-ingénierie, elle est implicitement présente à tous les niveaux de la réingénierie.
- La rétro-conception (Design recovery) : Consiste à utiliser des connaissances du domaine en vue d'obtenir une abstraction plus élevée pour permettre une meilleure compréhension du système.

La réingénierie (Reengineering) est le processus d'examen et de modification d'un système dans le but de le reconstituer dans une nouvelle forme puis l'implémenter. La réingénierie utilise donc des techniques de rétro-ingénierie et d'ingénierie directe. Le processus de réingénierie s'emploie, à partir d'un existant, plus ou moins bien documenté. Il consiste alors à remonter dans les niveaux d'abstraction afin d'approfondir les connaissances. La rétro-ingénierie représente la phase montante de la réingénierie [Benslimane, 2007].

Ainsi, à partir du niveau physique, il est possible de faire de la reconstruction, de la re-documentation, de la rétro-ingénierie avec de la rétro-conception. Les mêmes opérations peuvent être effectuées à partir du niveau conceptuel, alors qu'au niveau analyse, on ne peut effectuer que des opérations de restructuration.

A.III.2.2. La rétro-ingénierie pour une réingénierie

- Une première tentative de classification des trois notions : Ingénierie, rétro-ingénierie et réingénierie est présentée dans [Chikofsky et Cross, 1990]. Ils ont intégré une autre notion qui est la restructuration (cf. Figure n° 9).

  La restructuration (Restructuring) permet la transformation d'une représentation en une autre au même niveau d'abstraction, tout en conservant la sémantique et le comportement du système.

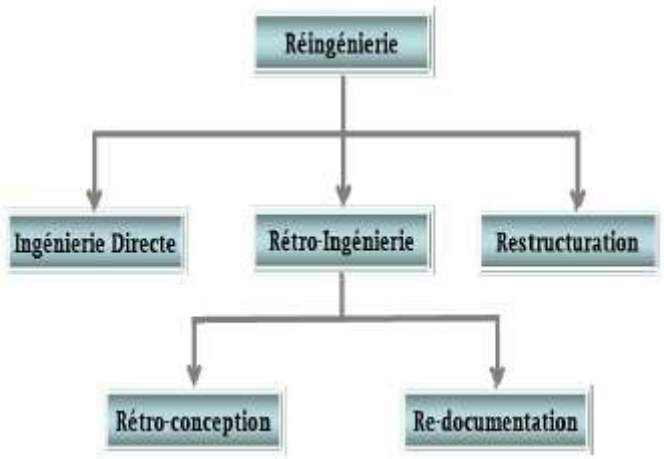

Fig. 9. Classification de Chikofsky et Cross [Benslimane, 2007].

- Une seconde classification est proposée par Chiang [Chiang, 1995]. Elle recentre la rétro-ingénierie dans le processus de réingénierie d'un système (cf. Figure n° 10). Ainsi, la rétro-ingénierie consiste à produire une abstraction d'un système existant, alors que la restructuration peut ne concerner que le passage direct de l'ancien au système restructuré. Si le système doit évoluer, il faut prendre en compte les nouveaux besoins au travers une phase complémentaire d'ingénierie directe.

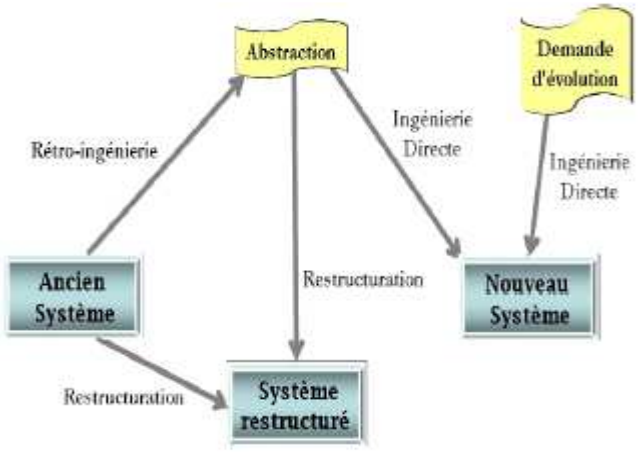

Fig. 10. Classification de Chiang [Benslimane, 2007].

- Une troisième taxonomie est proposée par Ebert et al. [Ebert et al., 1999]. Elle distingue la phase de compréhension du système de sa rénovation. La compréhension du système ou la rétro-ingénierie utilise un ensemble de techniques : Interrogation, navigation, mesure et récupération de la conception (Designe recovery). La rénovation, quant à elle, comprend les tâches de re-documentation, restructuration et re-modularisation des systèmes (cf. Figure n° 11).

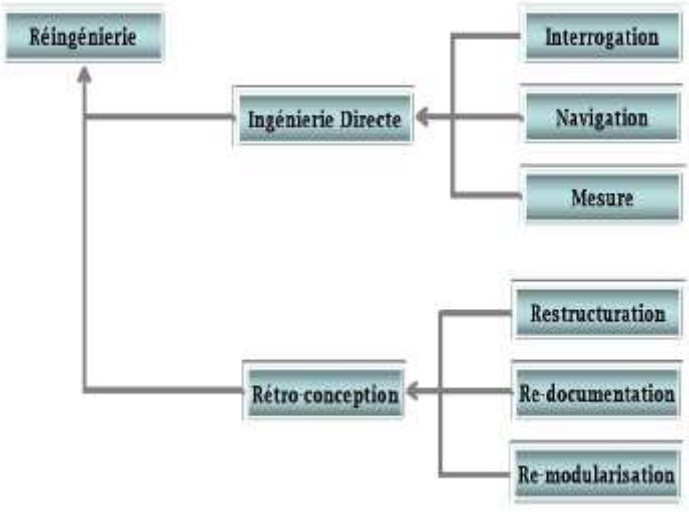

Fig. 11. Classification d'Ebert et al [Benslimane, 2007].

- Une quatrième classification d'Akoka et Comyn-Wattiau propose d'élargir et de compléter ces classifications [Akoka et Comyn-Wattiau, 2001]. (cf Figure n° 12).

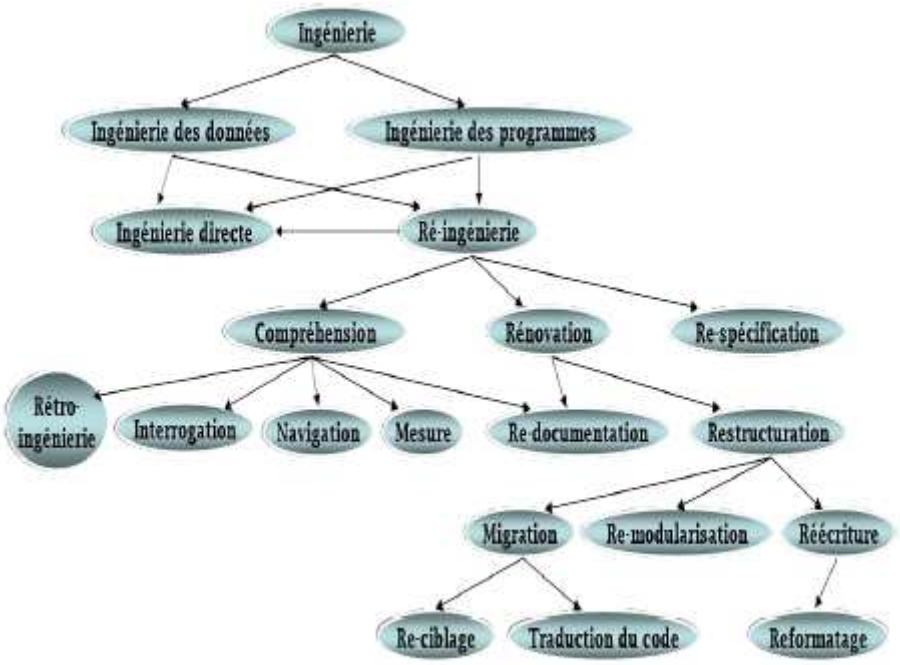

Fig. 12. Classification d'Akoka et Comyn-Wattiau [Benslimane, 2007].

A.III.2.3. Rétro-ingénierie des applications web

La rétro-ingénierie des applications web (Pages web, documents web, interfaces web, …) regroupe l'ensemble des techniques et des outils liés à la compréhension d'un système logiciel et/ou matériel existant, sans le bénéfice des spécifications originales. Réalisée généralement en deux phases, l'identification des composants du système et de leurs relations puis leurs représentations à un niveau d'abstraction plus élevé et plus compréhensible par les humains (Spécifications, documentations, modèles, schémas) [Chikofsky et Cross, 1990].

La rétro-ingénierie des applications web est un défi technique pour plusieurs raisons :

- L'intention du développeur initial n'est pas claire ;
- Le problème initial peut ne pas être résolu correctement ;
- La conception a été mal faite ou franchement non faite ;
- Le code a été altéré pour répondre à des besoins de maintenance ;
- L'application est développée sur un environnement différent ;
- La plateforme matérielle a changé ou évolué ;
- La sémantique de l'outil de développement et surtout le langage a changée.

C'est principalement pour cela que les mainteneurs des applications web doivent souvent faire des suppositions telles que «Que fait ce module ?», «Pourquoi cette opération est-elle réalisée de cette façon ?» [Erich et John, 1991] et «Pourquoi cette interface ?».

A.III.2.4. Objectifs de la rétro-ingénierie des applications web

La rétro-ingénierie des applications web a deux objectifs principaux : La redocumentation et la récupération du modèle de conception. La redocumentation a pour but de produire une vue alternative d'un modèle, au même niveau d'abstraction. La récupération du modèle de conception vise à recréer les modèles d'abstraction à partir du code source (Script des documents web par exemple), de la documentation existante, de la connaissance des experts ainsi que toute autre source d'informations [Gerardo et Massimiliano, 2007]. Le but de la rétro-ingénierie d'une application web est d'en améliorer sa compréhension globale, autant pour sa maintenance que pour son développement futur. Son analyse sera menée selon les axes suivants : [Chikofsky et Cross, 1990]

- Faire face à la complexité : Création de méthodes et outils, combinés aux environnements, afin de fournir un moyen d'extraire des informations pertinentes ;
- Générer des vues alternatives : Les représentations graphiques sont considérées comme des aides à la compréhension ;
- Récupérer l'information perdue : L'évolution d'un système de taille conséquente entraîne la perte d'information, les modifications ne sont généralement pas documentées ;
- Détecter les effets de bord : La conception initiale ou les modifications successives peuvent entraîner des effets de bords involontaires ;
- Créer des abstractions : La discipline nécessite des méthodes et outils afin de créer des abstractions à un niveau plus haut ;
- Faciliter la réutilisation : La rétro-ingénierie peut aider à trouver des logiciels potentiellement réutilisables.

Au delà de la maintenance logicielle, la rétro-ingénierie peut être utilisée dans d'autres domaines, notamment à des fins de tests ou d'audit de sécurité et de vulnérabilités.

A.III.3. Conclusion

Dans ce chapitre, nous nous sommes intéressés à exposer deux importants points : Le premier a concerné les différentes techniques d'indexation y compris l'indexation sémantique, le deuxième a touché le concept de rétro-ingénierie et sans doute nous avons pu différencier entre : Ingénierie, rétro-ingénierie et réingénierie ; en outre leur classification.

La prochaine partie de notre travail présentera deux chapitres qui résument, en quelques sortes, notre contribution qui commencera par dresser un état de l'art sur les deux notions : Indexation sémantique et rétro-ingénierie et s'achèvera par notre proposition de rétro-ingénierie.

Partie B.

De l’Ontologie vers l’Indexation Sémantique pour une Rétro-Ingénierie.

CHAPITRE B.I.

Indexation Sémantique et Rétro-Ingénierie : État de l'Art

Sommaire

Le contenu de cette deuxième partie de notre travail présentera notre contribution. En commençant par ce premier chapitre qui dressera deux états de l'art ; d'une part un état de l'art pour les approches d'indexation sémantique et d'autre part un autre état de l'art pour les processus de rétro-ingénierie. Chaque état de l'art s'achèvera par un tableau récapitulatif dressant le principe et les spécificités de chaque travail, et par nos contributions par rapport aux approches d'indexation sémantique et par rapport aux processus de rétro-ingénierie.

B.I.1. Approche d'indexation sémantique

B.I.1.1. Etat de l'art

WILKINSON R. dans [Wilkinson, 1994] a été le premier qui a proposé un système de recherche d'information basé sur la structure du document. Dans son système, les documents sont divisés en sections et la requête est appliquée à chaque section. La pertinence du document dépend de différents aspects : La fréquence du terme dans le contenu du document, la fréquence du terme dans le contenu de la section et le type de la section. Il applique la formule TF-IDF à la section du document au lieu de l'ensemble du document [Chagheri et al., 2009].

MYAENG S. H. et ses collègues dans [Myaeng et al., 1998] proposent un modèle de recherche d'informations dérivé du modèle du réseau d'inférence. Ils ont amélioré l'efficacité de recherche de document dans son modèle comparé à des systèmes traditionnels basés sur un document entier sans structure. Dans ce réseau, le document est représenté par une hiérarchie de nœuds où les feuilles contiennent la partie textuelle.

Cette approche est basée sur le réseau Bayesian[9] pour calculer la probabilité de l'occurrence du terme dans l'élément logique du document.

GAGNON O. dans [Gagnon, 2013] présente différentes expériences pour l'indexation de documents Web à l'aide d'ontologies, des graphes orientés constitués d'entités et de relations propres au domaine du Web Sémantique. Il a fait ensuite une expérience visant à comparer la qualité des résultats aux requêtes d'information à celle d'une technique d'indexation classique basée sur la présence des mots clés ou sur l'utilisation de la méthode de l'espace vectoriel.

A la fin, il a défini un ensemble d'expériences afin de confirmer ou d'infirmer l'efficacité de la méthode d'indexation par ontologies.

MASS Y. et MATAN M. dans [Mass et Matan, 2003] décrivent une méthode pour le classement composé dans des documents XML par la création des indices séparés pour plus d'éléments logiques dans la collection des documents. Dans [Mass et Matan, 2004], Ils ont rénové leur approche en proposant un document pivot pour compenser le problème des données en dehors de la portée de l'élément logique. Le document pivot mesure le rapport du résultat d'éléments logiques par le résultat de leurs articles contenants. La méthode est basée sur le modèle d'espace vectoriel et la formule TF-IDF [Chagheri et al., 2009].

Pour LALMAS M. [Lalmas, 2000], le document est représenté par un arbre avec des nœuds, branches et feuilles qui représentent, respectivement, les éléments logiques, les relations de composition et les données bruts dans la partie textuelle du document. Son approche est basée sur un raisonnement évident qui peut être appliqué à un modèle de représentation uniquement du contenu, uniquement la structure et à la fois le contenu et la structure du document.

[9] Est un modèle graphique probabiliste représentant des variables aléatoires sous la forme d'un graphe orienté acyclique. Il est à la fois : Un modèle de représentation des connaissances et une « machine à calculer » une probabilité conditionnelle.

Dans [Kazai et al., 2002], KAZAI G. avec son équipe utilisent les meilleurs points d'entrée (Best Entry Points, BEPs) qui correspondent aux éléments logiques du document où les utilisateurs peuvent parcourir pour accéder aux éléments pertinents du document. Cette approche représente l'élément document comme un rassemblement du contenu de tous ses sous éléments logiques.

KAREN S. P. et BOUGHANEM M. dans [Karen et Boughanem, 2006] ont combiné les facteurs différents pour calculer le poids des termes dans un document structuré. Ils ont utilisé TF pour prendre en considération l'importance locale du terme, IDF et IEF pour l'importance globale du terme dans le document et la collection. L'IEF^d estime l'importance semi-globale du terme dans la collection des éléments structurés dans le document.

Dans [Schileder et Holger, 2002], SCHLIEDER T. et HOLGER M. adoptent la mesure de similarité d'un modèle d'espace vectoriel, il incorpore la structure du document et supporte la structure des requêtes. Ils étendent le terme à un terme structuré qui inclut la structure de la requête et du document. Les notions de la fréquence du terme et la fréquence du document inversé sont adaptées à l'élément logique des documents.

En dehors de l'indexation structurée, récemment beaucoup de chercheurs se sont concentrés sur les systèmes de recherche d'informations conceptuels en utilisant des ressources sémantiques [Salton, 1968]. Voorhees E. dans [Voorhees, 1993] utilise WordNet pour désambiguïser des termes en prenant en considération la relation hyponyme entre les synsets de WordNet. Ses expérimentations sont appliquées, seulement, à des synsets de noms.

GUARINO N. et ses collègues dans [Guarino, 1999] présentent OntoSeek un système de recherche, spécifiquement balisé pour les pages jaunes on-line et produit des catalogues en utilisant des ressources sémantiques comme WordNet pour supporter le matching du contenu. Dans ce système, un formalisme de base de graphe conceptuel est utilisé.

KHAN L. avec son équipe dans [Khan, 2004] propose un modèle basé sur les concepts en utilisant des ontologies de domaine dépendant. Dans cette méthode, il utilise un algorithme de désambiguïsation automatique qui balaye les concepts non pertinents. Seuls les concepts pertinents sont associés aux documents et donc ils participent à la génération des requêtes [Chagheri et al., 2009].

ZARGAYOUNA H. et SALOTI S. Dans [Zargayouna et Saloti, 2004], utilisent le calcul des poids du terme qui est influencé par le contexte (Unité d'indexation) dans lequel il apparait. Le calcul du poids en se basant sur la formule TF-IDF est appliqué aux balises. Ainsi, les auteurs proposent la formule TF-ITDF, qui estime la capacité de discrimination d'un terme t pour une balise b dans un document d. Ce travail utilise le concept et la structure du document conjointement.

CHAGHERI S. et ses coéquipiers dans [Chagheri et al., 2009] proposent un modèle d'indexation sémantique qui exploite à la fois les structures logiques et le contenu sémantique des documents. Cette méthode est une extension du modèle de vecteur de Salton [Salton, 1968] en ajustant le calcul du TF-IDF en prenant en considération l'élément structure à la place de tout le document.

B.I.1.2. Récapitulation.

Approche de ...	Principe	Spécificités
Voorhees E. *[Voorhees, 1993]*	Basé sur WordNet.	- L'hyponyme entre les synsets de noms permet de désambiguïser des termes.
Wilkinson R. *[Wilkinson, 1994]*	Document structuré en sections.	- Requête de recherche / Section. - Application de : TF-IDF. - Fréquence du terme / Document. - Fréquence du terme / Section.
Myaeng S. H. et al. *[Myaeng et al., 1998]*	Document représenté par des nœuds et des feuilles qui contiennent la partie textuelle.	- Modèle de RI dérivé du modèle du réseau d'inférence. - Basé sur le réseau Bayesian.
Guarino N. et al. *[Guarino et al., 1999]*	Il a réalisé OntoSeek (Système de Recherche balisé).	- Production de catalogues en utilisant Wordnet. - Utilise le graphe conceptuel.
Lalmas M. *[Lalmas, 2000]*	Document représenté par un arbre.	- Modèle représentant le contenu ou la structure ou les deux à la fois.
Kazai G. et al. *[Kazai et al., 2002]*	Basé sur les BEPs	- Le document est un rassemblement de tout le contenu de ses BEPs (Eléments logiques).
Schileder T. et Holger M. *[Schileder et Holger, 2002]*	Document structuré et requête structurée.	- Modèle d'espace vectoriel. - Utilisation de la formule TF-IDF à l'élément logique des documents.
Mass et Matan *[Mass et Matan, 2003]*	Classement composé dans des documents XML.	- Eléments logiques déduits par la création des indices séparés.
Khan L. et al. *[Khan et al., 2004]*	Basé sur les concepts d'une ontologie.	- Algorithme de désambiguïsation automatique. - Déduction de concepts pertinents pour la génération de requêtes.
Mass Y. et Matan M. (Approche Màj) *[Mass et Matan, 2004]*	Proposition d'un document pivot (DP).	- Modèle espace vectoriel. - Utilisation de la formule TF-IDF.
Zargayouna H. et Salotti S. *[Zargayouna et Saloti, 2004]*	Utilise l'unité d'indexation qui est un ensemble de termes hiérarchiques au lieu d'un document.	- Utilisent les deux formules TF-IDF et TF-ITDF. - Utilisent le concept et la structure du document en même temps.
Karen S. P. et Boughanem M. *[Karen et Boughanem, 2006]*	Document structuré.	- Calcul des TF, IDF, IEF et IEF^d pour déduire l'importance locale, globale et semi globale des termes
Chagheri S. et al. *[Chagheri et al., 2009]*	Exploite en même temps la structure logique et le contenu sémantique des documents.	- Modèle étendu du modèle de vecteur de Salton. - Formule TF-IDF ajusté (Structure à la place du document).
Gagnon O. *[Gagnon, 2013]*	Basé sur les ontologies et les graphes orientés.	- Comparaison entre l'indexation à l'aide des ontologies et celle basée sur des mots clés.

Tab. 2. Récapitulation des travaux sur l'indexation sémantique.

B.I.1.3. Contributions de notre approche

Avant d'exposer l'apport de notre approche d'indexation sémantique par apport à celles décrites dans la section B.I.1.1 et récapitulées dans le tableau n° 2, il est à signaler, en premier lieu, ***que la nôtre, nous l'avons utilisée pour réaliser le processus de rétro-ingénierie des applications orientées web et par la suite maintenir ces applications, alors que les travaux connexes l'ont utilisé pour indexer seulement***. En deuxième lieu, plusieurs points communs existent entre notre approche et celles des auteurs des travaux cités précédemment, à savoir : Indexer des documents dites structurés (En réalité ce sont des documents semi-structurés), utilisation des formules de calcul de la fréquence des termes, en général, dans un document pour déterminer les termes fréquents, Utilisation de WordNet et représentation graphique du document en question.

Cependant, pour décrire la contribution de notre approche, il est important de signaler que nous avons [Dennai et Benslimane, 2015] :

1. Utilisé deux types de documents : Document non structuré (Page HTML) et document semi structuré (Document XML),
2. Utilisé une ontologie de domaine, d'une part, pour identifier et valider les termes extraits à partir des documents (Application de la distance sémantique entre terme-document et concept-ontologie) et d'autre part, pour enrichir l'index résultat (Application d'une mesure de similarité sémantique entre une paire de concepts d'une ontologie),
3. Utilisé, pour les termes fréquents, le dictionnaire sémantique WordNet et l'étiqueteur TreeTagger afin de déduire des termes dérivés.

B.I.2. Processus de rétro-ingénierie

B.I.2.1. Etat de l'art

BENCHEIKH A. et ses coopérateurs dans [Bencheikh et al., 2009] proposent une méthode de rétro-ingénierie des processus métiers dont l'un des points forts est de prendre en compte les motivations pour guider le processus de modélisation. Cette méthode a été caractérisée entre autres, par sa généricité puisqu'elle s'est basée sur un méta-modèle générique qui capture tous les éléments intervenant dans un processus dans cinq vues complémentaires (Intentionnelle, fonctionnelle, organisationnelle, interactionnelle et comportementale).

La démarche proposée, qui est composée de quatre phases à savoir : Définition de l'objectif, collecte des informations puis identification des vues des processus et modélisation, est illustrée par des exemples de l'application PostBac permettant de gérer les procédures d'admission des élèves de terminale dans les formations de l'enseignement supérieur.

L'apport des auteurs est l'utilisation de la phase « Identification des vues des processus » qui permet d'organiser les informations collectées, de rechercher toute information manquante et d'étudier tous les aspects d'un processus en utilisant les différentes vues du méta-modèle.

BESBES S. dans son livre [Besbes, 2008] présente RetroWeb, une démarche de reconstruction des sites web non documentés. Elle donne une description du contenu informatif du site à différents niveaux d'abstraction (Physique, logique et conceptuel) et utilise à chaque niveau d'abstraction un méta-modèle. RetroWeb se déroule en trois étapes : L'extraction, la désoptimisation et la conceptualisation. La première étape extrait les données semi structurées de chaque page HTML du site et d'instancier le méta-modèle des vues physiques (Une vue physique par page HTML).

La deuxième étape traduit les vues physiques en vues logiques et instancier le méta-modèle des vues logiques en utilisant des règles de rétro-conception.

La troisième étape traduit les vues logiques en vues conceptuelles (Schèmes EAE : Entité Association Entité) en utilisant d'autres règles de rétro-conception et ensuite fusionne les différents schémas EAE obtenus en un schéma global qui donne une description du contenu informatif de la totalité du site.

BOUCHIHA D., dans [Bouchiha, 2005], a présenté un processus de rétro-ingénierie des applications Web à base d'ontologie. Il a commencé par l'extraction d'un sous schéma riche et réduit décrivant l'application Web à base d'une ontologie du domaine. Ce processus est en deux phases :

- L'extraction, à partir des pages HTML, des informations utiles pour les comparer par rapport aux informations présentées dans l'ontologie ; Pour lui, les informations utiles sont celles présentées sous forme de tableaux et de listes.
- La reconnaissance des concepts de l'ontologie cachés dans les pages Web à l'aide des techniques de distance sémantique, puis inférer de nouveaux concepts et relations. A la fin, génération du schéma conceptuel UML.

BENSLIMANE S. M. a proposé dans [Benslimane, 2007] un processus de réingénierie des applications Web vers le Web sémantique. L'une des deux étapes de son approche est une présentation d'un processus de rétro-ingénierie pour l'extraction de la sémantique encapsulée dans la structure et les instances des formulaires HTML de l'application Web.

Il a débuté par une phase d'analyse des pages HTML pour l'identification du schéma XML du formulaire, suivie d'une phase de découverte des dépendances fonctionnelles, des dépendances d'inclusion et des contraintes. L'autre étape de son approche est une étape d'ingénierie pour la construction d'une ontologie.

BOUILLON L. et VANDERDONCKT J. dans [Bouillon et Vanderdonckt, 2004] proposent un modèle, une méthode et un outil permettant d'effectuer une rétro-ingénierie de pages Web écrites en HTML (L'interface), de manière à obtenir un ou plusieurs modèles de présentation. Ce modèle spécifie les différents objets de présentation contenus dans une page Web en les organisant en niveaux d'abstraction progressivement croissants : Les Objets d'Interaction Concrets (OIC), les Objets d'Interaction Abstraits (OIA), les Fenêtres Logiques (FL) et les Unités de Présentation (UP). Différents modèles avec variantes peuvent être produits en fonction d'une série d'options et d'heuristiques de rétro-ingénierie portant sur les différents niveaux. A partir de ces modèles de présentation, une ingénierie progressive permet de produire une nouvelle IHM (Interface Homme Machine) adaptée à un autre contexte d'utilisation.

DECOURSELLE J. expose dans [Decourselle, 2013] un projet des étudiants de master informatique encadrés par Lumineau M. et Duchateau M. qui est l'étude d'une solution concrète de rétro-conception d'une base de données.

Pour ce projet, elle consiste en la recherche d'une solution d'extraction de métadonnées depuis un modèle relationnel et la traduction de ces informations dans un schéma entité/association. Un algorithme a été élaboré et implémenté dans un environnement web.

B.I.2.2. Récapitulation

Processus de ...	Principe	Spécificités
Bouillon L. [Bouillon et Vanderdonckt, 2004]	Un modèle, une méthode et un outil pour effectuer une rétro-ingénierie de pages Web HTML (Interface).	- Obtention d'un ou plusieurs modèles de présentation de différentes variantes. - Spécification de différents objets : OIC, OIA, FL et UP pour une nouvelle IHM.
Bouchiha D. *[Bouchiha, 2005]*	Rétro-ingénierie des applications Web à base d'ontologie (Pages HTML)	- Processus sur deux étapes : L'extraction des informations et reconnaissance à base d'ontologie. - Utilisation de la distance sémantique.
Benslimane S. M. *[Benslimane, 2007]*	Réingénierie des applications Web (L'une des phases est la rétro-ingénierie) pour construire une ontologie.	- Analyse des pages HTML pour identifier un schéma XML. - Découvrir les dépendances fonctionnelles, d'inclusion et des contraintes.
Besbes S. *[Besbes, 2008]*	Reconstruction des sites web (Pages HTML) non documentés. (Démarche appelée RetroWeb)	- Utilisation à chaque niveau d'abstraction (Physique, logique et conceptuel) un méta-modèle. - Processus sur trois étapes : L'extraction, la désoptimisation et la conceptualisation.
Bencheikh A. *[Bencheikh et al., 2009]*	Rétro-ingénierie des processus métiers.	- Basée sur un méta-modèle générique capturant des éléments d'un processus dans 5 vues complémentaires. - Composée de 4 phases : Définition de l'objectif, collecte des informations puis identification des vues des processus et modélisation.
Decourselle J. *[Decourselle, 2013]*	Rétro-conception d'une base de données.	- Extraction de métadonnées depuis un modèle relationnel. - Traduction de ces informations dans un schéma EA. - Déduction d'un algorithme implémenté sur Web.

Tab. 3. Récapitulation des travaux sur la rétro-ingénierie des applications web.

B.I.2.3. Contributions de notre processus

Le point primordial de notre contribution dans le processus de la rétro-ingénierie des applications orientées web est qu'il est à la base d'indexation sémantique. Nous avons réalisé et en même temps exploité notre approche d'indexation sémantique parallèlement avec l'exécution du processus. L'index, résultat final de l'approche, est analogue à un dictionnaire de données pour un système d'information, amélioré par une couche sémantique [Dennai et Benslimane, 2015].

Les auteurs des travaux suscités ont divisé la rétro-ingénierie en étapes basées sur des modèles ou méta-modèles, par contre, notre processus a appliqué les phases de notre indexation sémantique en créant un processus de rétro-ingénierie sur quatre phases à savoir : La modélisation, l'attachement sémantique, l'enrichissement et la re-conceptualisation [Dennai et Benslimane, 2015].

B.I.3. Conclusion

Le cadre primordial de ce chapitre était de présenter, dans une importante partie, un ensemble des travaux réalisés dans les domaines de l'indexation sémantique des documents en général et la rétro-ingénierie des applications web. La plupart de des travaux réalisés sur l'indexation sémantique prennent en considération la structuration des documents (Déjà structurés en sections par exemple ou à structurer) avant de lancer le processus d'indexation ; des auteurs sont allés, même plus loin, en structurant les requêtes appliquées aux sections. D'autres auteurs ont indexés sur la base d'ontologie, WordNet, des éléments logiques du contenu du document, etc. Les auteurs des travaux réalisés sur la rétro-ingénierie se sont, quasiment, basés sur les modèles et les méta-modèles.

Les travaux réalisés sur la rétro-ingénierie s'articulent, en général, sur trois types : La rétro-ingénierie des interfaces des pages web, la rétro-ingénierie des applications web (pages HTML) et la rétro-ingénierie (Rétro-conception) des bases de données.

Le principe et les caractéristiques de chacun de ces travaux nous ont permis de revoir le processus d'indexation sémantique en l'améliorant et d'essayer de l'appliquer dans le déroulement des phases de notre processus de rétro-ingénierie des applications orientées web proposé, en prenant comme exemple des documents XML ou des pages HTML.

CHAPITRE B.II.

Rétro-Ingénierie des Applications Web À Base d'Indexation Sémantique : Conception et Implémentation

Sommaire

Le domaine du développement des applications orientées web nécessite, actuellement, la prise en considération le passage du web traditionnel vers le web sémantique, qui est un sujet de recherche d'actualité fortement abordé par les développeurs du web. Les technologies HTML et XML restent très importantes dans ce domaine et qui apparaissent comme ressources intéressantes pour tout ce qui peut constituer de véritables réservoirs de documents numériques.

L'usage de plus en plus du XML et de degrés moins du HTML dans la structuration du web offre des possibilités de combinaison entre la recherche d'information et l'interrogation des bases de données dans le web et ce à travers leurs façons très fines de description des documents et des liens entre leurs différentes parties.

Quelques méthodologies de conception ont été proposées pour des applications Web basées sur HTML. Mais les limites imposées par ce langage, notamment durant le processus de recherche d'information, et l'émergence de XML comme format de données amène tout naturellement à utiliser XML pour la construction de sites web importants.

Cette utilisation permet d'exploiter les énormes possibilités de représentation et d'interopérabilité offertes par ce langage. Elle permet d'une part d'effectuer une séparation nette et distincte entre le contenu du site (Données) et la présentation durant le processus de conception du site et, d'autre part, d'exploiter les données du site après sa réalisation.

Dans ce deuxième chapitre de cette deuxième partie, nous allons présenter notre approche de retro-ingénierie des applications orientées web à base d'indexation sémantique. Pendant notre aborde, nous allons essayer d'exploiter des documents web non structurés ou semi structurés, tels que, respectivement, les pages HTML et les documents XML et qui peuvent être des constituants d'une application web (Système) [Dennai et Benslimane, 2015].

L'approche proposée, qui est à la base d'indexation sémantique, utilise une ontologie de domaine. Elle prend en considération la structure et le contenu de ces deux types de documents et inclut les phases suivantes [Dennai et Benslimane, 2015] :

1. Extraction des concepts cachés dans des pages HTML incluant des tables, des listes et des formulaires ; ou marqués par des balises dans des documents XML en se basant sur la modélisation de ces deux types de documents web.
2. Attachement pour validation des concepts extraits à des concepts d'une ontologie du même domaine que ces documents web.
3. Enrichissement des concepts rattachés par d'autres concepts de la même ontologie en utilisant la mesure de similarité sémantique de Wu et Palmer.
4. Re-conceptualisation du système en utilisant tous ces concepts qui peuvent construire un dictionnaire de données d'un système d'information.

B.II.1. Rétro-ingénierie à base d'indexation sémantique

Notre approche d''indexation sémantique utilise, d'une part, une ontologie de domaine pour valider les concepts extraits à partir des deux sources d'information (Page HTML ou document XML) et d'autre part, une mesure de similarité sémantique entre concepts de cette ontologie dans le but d'enrichir l'index. Elle sera conçue en trois étapes : Modélisation, attachement sémantique pour validation et enfin enrichissement. L'index résultat permet de déclencher une nouvelle Re-conception d'un système d'information (cf. figure n° 13).

B.II.1.1. Approche générale d'indexation

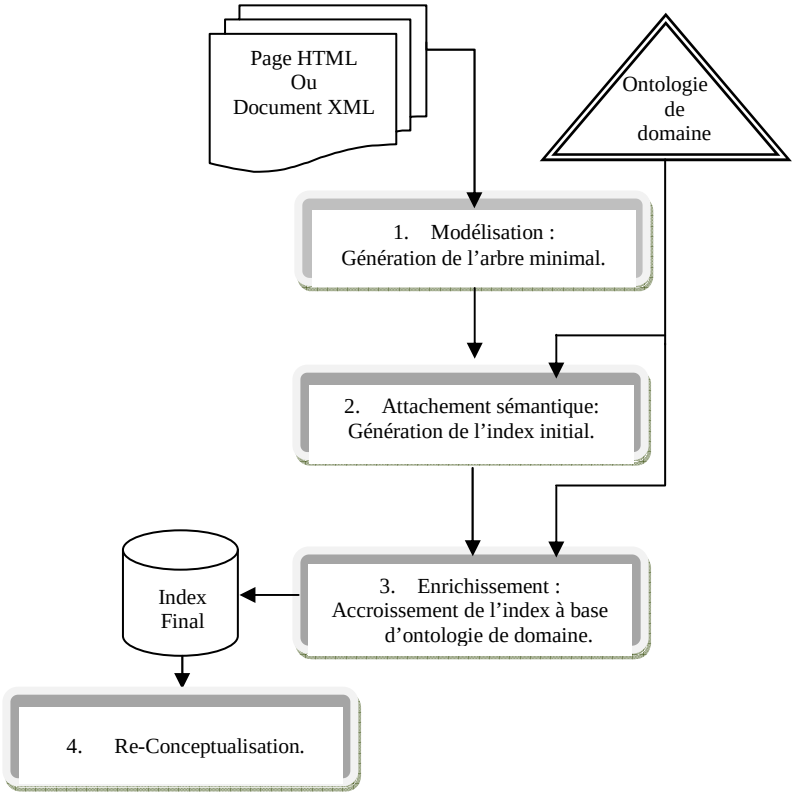

Fig. 13. Approche générale d'indexation [Dennai et Benslimane, 2015].

B.II.1.1.1. Modélisation

Durant cette phase, nous commençons par modéliser les pages HTML ou les documents XML en utilisant un module parseur et par la suite nous extractions la structure donnée par les balises dans ces deux sources d'information [Dennai et Benslimane, 2012].

Nous représentons la structure HTML ou XML avec un arbre libellé dont chaque élément (Ou attribut) correspond à un nœud de l'arbre. Enfin, nous générons la structure de l'arbre minimal trouvé en éliminant les chemins doubles où chaque unité sémantique représente une unité d'information (Chemin unique) [Dennai et Benslimane, 2012 ; Dennai et Benslimane, 2015].

Les principales étapes de la génération de l'arbre minimal sont décrites dans la figure n° 14 ci-après.

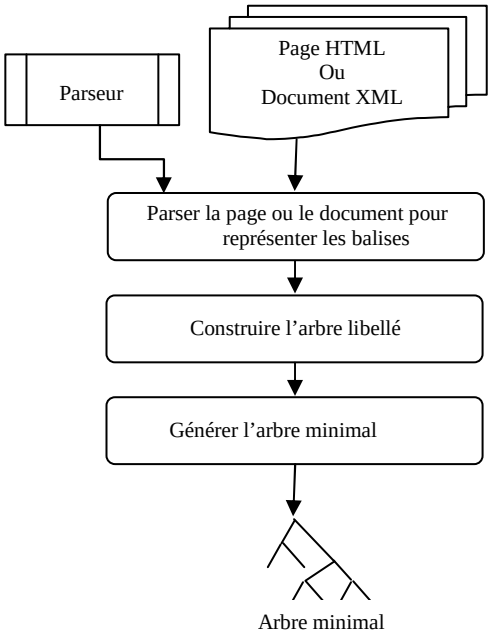

Fig. 14. Phase de modélisation [Dennai et Benslimane, 2015].

Ci-après l'algorithme de la génération de l'arbre minimal :

```
Input:        HTML Page or XML Document.
Output:       Minimal Tree.
-------------------------------------------------------------------------------
Load WEB file                                                  /*.HTML or *.XML */
To parser the document
Pointer on beginning file        /* Création de la liste chainée LIST1 et de l'arbre libellé TREE1 */
WHILE NOT END OF File DO
    Fill LIST1 by the TAGS
    Creating the TREE1
    Each NODE represents a TAG
    Next in the File
END WHILE                                          /* Fin de création de l'arbre libellé */
I Pointer on the ROOT of TREE1
WHILE TREE1I ≠ NIL-1 DO
    J Pointer on the I+1 of TREE1
    WHILE TREE1J ≠ NIL DO                          /* Elimination des chemins doubles */
        IF TREE1I = TREE1J THEN to delete TREE1J
        Next TREE1J
    END WHILE
    Next in TREE1I
END WHILE
Pointer on the ROOT of TREE1                                   /*Charger TREE1 */
WHILE TREE1 ≠ NIL DO
    Fill the table TABLE1 by nodes of the tree
    Next in TREE1
END WHILE
```

En appliquant l'algorithme de la génération de l'arbre minimal (Comme entrée : Document XML, cf. figure n° 15.), nous obtenons l'arbre libellé et l'arbre minimal représentés respectivement par les figures 16 et 17.

```
<Sector Name = "TOURISM">
    <S-Sector> Hosting </S-Sector>
    <Residences>
        <Residence> Hotel </Residence>
        <Residence> Inn</Residence>
        <Cells>
          <Cell> Hotel Rooms </Cell>
          <Cell> Cabins </Cell>
        </Cells>
    </Residences>
    <Localities>
        <Locality>Béchar (Algeria) </Locality>
        <Locality> Oran (Algeria) </Locality>
    </Localities>
    <Cells>
    <Cell>Tourist residences </Cell>
    </Cells>
    </Sector>
```

Fig. 15. Document XML [Dennai et Benslimane, 2012].

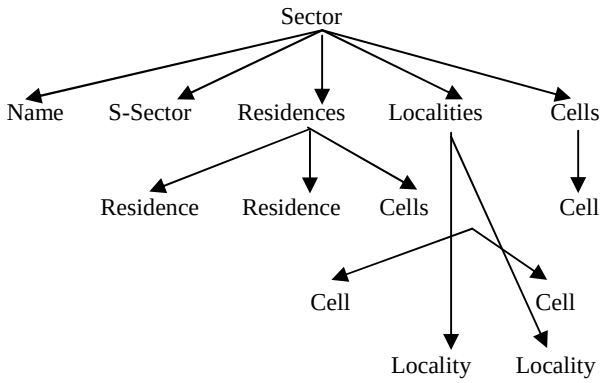

Fig. 16. Arbre libellé [Dennai et Benslimane, 2012].

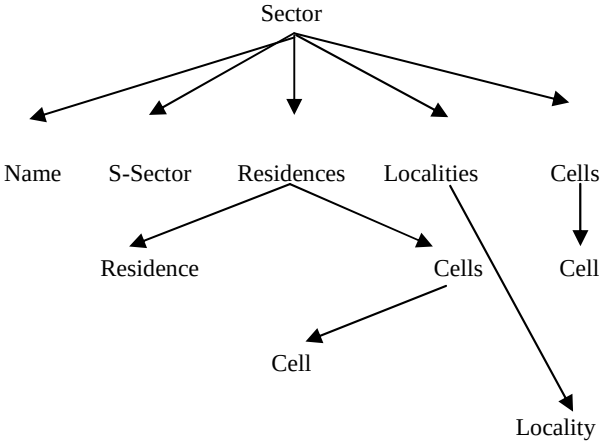

Fig. 17. Arbre minimal [Dennai et Benslimane, 2012].

B.II.1.1.2. Attachement sémantique (Pour validation)

Au cours de cette phase, un index initial est généré par l'attachement des termes (Des nœuds) de l'arbre minimal avec les concepts de l'ontologie de domaine. Un nœud de chaque chemin unique, connu sous le nom d'une unité d'information ou d'une unité sémantique, est attaché avec un concept de l'ontologie, auquel il fait référence et ce par un calcul de la distance sémantique entre les termes de l'arbre et les concepts de l'ontologie [Dennai et Benslimane, 2015].

L'attachement sémantique est accompli par l'utilisation d'une mesure de similarité sémantique qui sert à retrouver des structures similaires sémantiquement et ayant des labels différents ou par l'utilisation d'une ressource externe tel que la base de données lexicale WordNet [Dennai et Benslimane, 2014].

Nous continuons dans cette phase par l'intégration des termes attachés, initialement, dans une structure appelée index et nous achevons par l'enrichissement de ces termes par d'autres (Résultats de WordNet) en les connectons encore une fois avec les concepts de l'arbre minimal, puis les intégrer dans l'index. La génération de l'index initial est décrite dans la figure n° 18.

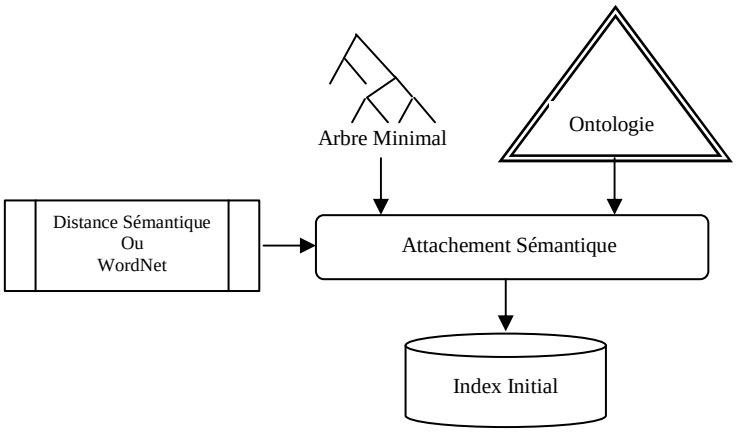

Fig. 18. Phase de génération de l'index initial [Dennai et Benslimane, 2015].

Ci-après l'algorithme de la génération de l'index initial :

```
Input:    Domain Ontology File + Minimal Tree.
Output:  Initial Index.
-----------------------------------------------------------------------------------------
                    /* Charger le fichier ONTOLOGIE file [*.OWL] du même domaine que HTML ou XML */
Pointer on beginning file
WHILE NOT END OF File DO
    Fill the table TABLE2 by the ontology concepts
    Next in file
END WHILE
                                                                    /* Création de la liste chainée LIST_INDEX */
FOR i=1 TO n                                                        /* n est la taille de TABLE1 */
    FOR j=1 TO m                                                    /* m est la taille de TABLE2 */
        Call WordNet
    IF TABLE1 [i] ≈ TABLE2 [j] THEN
        Fill LIST_INDEX with the elements of TABLE1 [i] /* Utilisation de WordNet */
    END FOR
END FOR
```

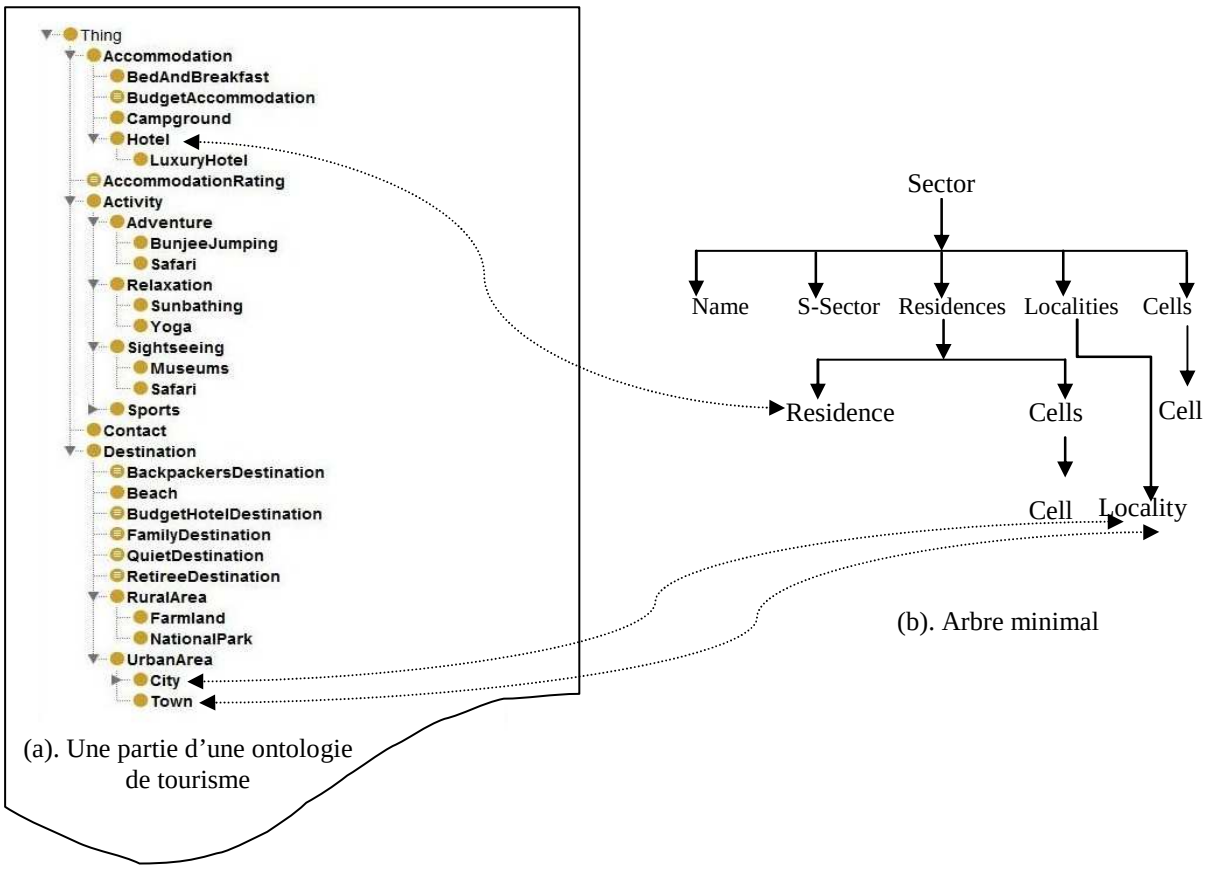

Fig. 19. Attachement sémantique des termes avec des concepts de l'ontologie [Dennai et Benslimane, 2015].

La figure n° 19 montre l'attachement sémantique des termes d'une unité sémantique avec des concepts d'une ontologie de domaine. Pour ce but, nous utilisons une ontologie de domaine de tourisme[10], elle est considérée comme étant un tutorial pour le web sémantique.

B.II.1.1.3. Enrichissement

En utilisant l'étiqueteur TreeTagger[11], nous pouvons produire la catégorie grammaticale et le lemme de chaque terme fréquent de l'unité sémantique résultat de calcul des deux fréquences pondérées des termes en utilisant les deux formules : *TF-IDF* et *TF-ITDF*. Les deux formules utilisent le nombre d'occurrences du terme dans une page HTML ou un document XML en outre la deuxième formule prend en considération le nombre d'occurrences du même terme dans une section (cf. chapitre n° A.II.) [Dennai et Benslimane, 2015].

Ce calcul nous permet de sélectionner d'autres termes résultats du TreeTagger (Sélectionner quelques catégories : noms, verbes, adjectifs, extraire le lemme des termes [Volk et al., 2002 ; Volk et al., 2003], enlever les termes longs et inverser les variantes des termes –filtrage et normalisation-) et les intégrer dans l'index.

Nous calculons les similarités entre les concepts relatifs aux termes avec ceux des autres termes co-occurrents dans la même unité sémantique puis nous enrichissons les fréquences des termes avec ces similarités. Enfin, nous intégrons les concepts de l'ontologie qui ne sont pas attachés dans l'index et qui sont sémantiquement similaires aux concepts attachés de la même ontologie en utilisant la mesure de similarité de Wu et Palmer [Wu et Palmer, 1994] (cf. chapitre n° A.II.) [Dennai et Benslimane, 2014].

En utilisant WordNet et TreeTagger, nous pouvons déterminer les termes dérivés de chaque terme fréquemment utilisé dans le document Web (HTML ou XML) et puis les intégrer dans l'index.

Les principales étapes de l'enrichissement de l'index à base d'ontologie de domaine sont présentées dans la figure n° 20.

[10] http://protege.stanford.edu/plugins/owl/owl-library/travel.owl

[11] Est un outil qui rend possible d'annoter un texte avec l'information sur les parties du discours (Nature des mots: noms, verbes, infinitifs et particules) et de l'information de lemmatisation

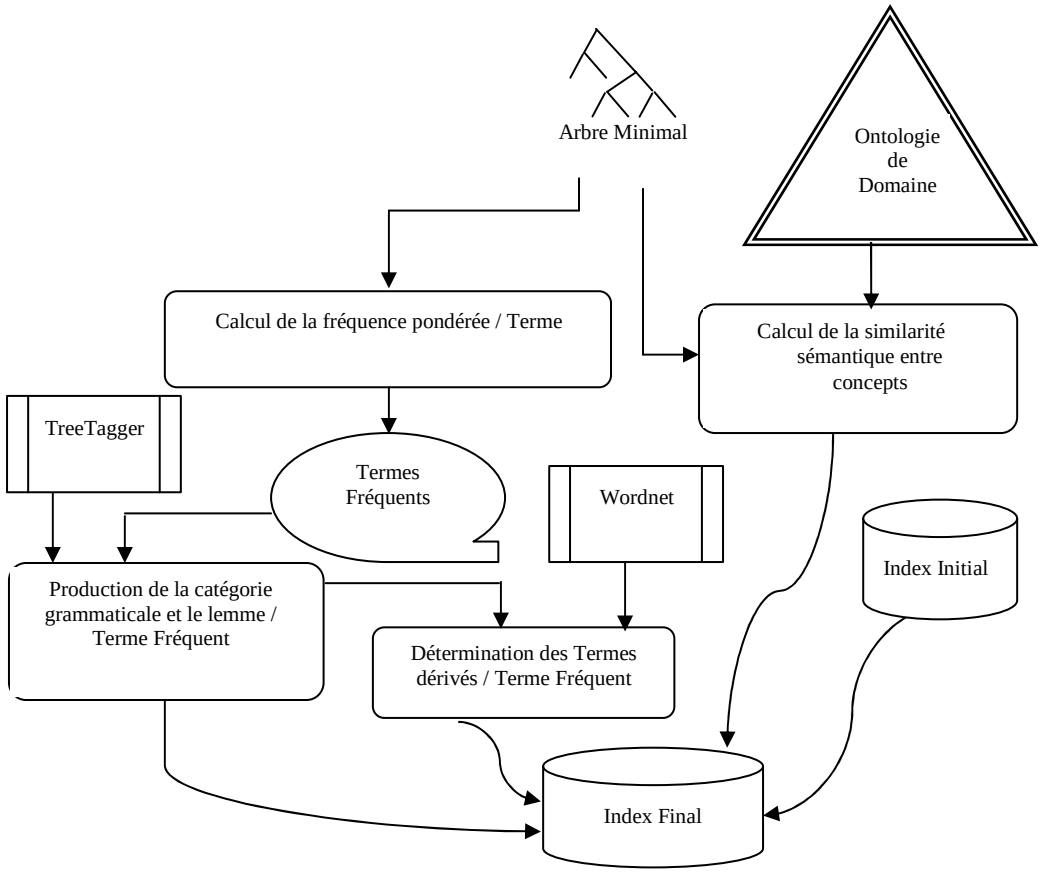

Fig. 20. Enrichissement de l'index à base d'ontologie de domaine, de Treetagger et de WordNet [Dennai et Benslimane, 2015].

Remarque :

Cette étape a besoin de l'achèvement des deux étapes précédentes. Elle utilise l'arbre minimal, résultat de la première étape, et l'index initial, résultat de la deuxième étape.

Ci-après l'algorithme de la génération de l'index final :

```
Input:    Initial Index + Minimal Tree + Domain Ontology.
Output:  Final Index.
-----------------------------------------------------------------------------------
FOR j=1 TO m-1                                                    /* m est la taille de TABLE2 */
    IF (SimWP (TABLE2 [j], TABLE2 [j+1]) ≈ X) AND (TABLE2 [j] ≈ element of LIST_INDEX) THEN
    /* SimWP() est une fonction de mesure de similarité sémantique de Wu et Palmer, X → (tend vers) 1 */
        To fill LIST_INDEX with elements of TABLE2 [j+1]
    ENDIF
END FOR
WHILE  LIST_INDEX ≠ NIL DO
    Call TREETAGGER
    Creating the list chained LIST2
    /* LIST2 contient la catégorie grammatical et le lemm de chaque élément de  LIST_INDEX  */
    Next in LIST_INDEX
END WHILE
WHILE LIST2 ≠ NIL DO
    IF (TF*IDF≠ 0) THEN
/* TF= Nombre d'occurences pour un élément de LIST2 dans HTML-page or XML-doc */
/* IDF= Nombre d'occurences pour un élément de LIST2 dans une unité sémantique,   ≈1: tend vers 1 */
        FOR j=1 TO m                                              /* m est la taille de TABLE2 */
          Call WordNet                                            /* Utilisation de WordNet */
          IF element of LIST2 ≈ TABLE2 [j] THEN
                Fill LIST_INDEX with element of LIST2 /* Ajouter dans la liste chainée LIST_INDEX */
          ENDIF
        END FOR
    ENDIF
    Next in LIST2
END WHILE
```

Par application de l'algorithme de génération de l'index final sur l'arbre minimal représenté par la figure n° 17 et en utilisant l'ontologie de tourisme, l'index final deviendra : {Residence, Locality, Hotel, City, Town, Cities, Hotels, Urban Area, Destination, Rural Area, …}.

B.II.1.1.4. Re-conceptualisation

À partir de l'index final crée à la fin de l'étape précédente, nous pouvons déduire ce que nous appelons un dictionnaire de données où ces dernières sont le contenu de cet index final. Il suffit de compléter ce contenu par d'autres informations telles que la définition du typage et la précision du format. De ce fait, une nouvelle conception du système peut commencer. La figure ci-après interprète cette phase.

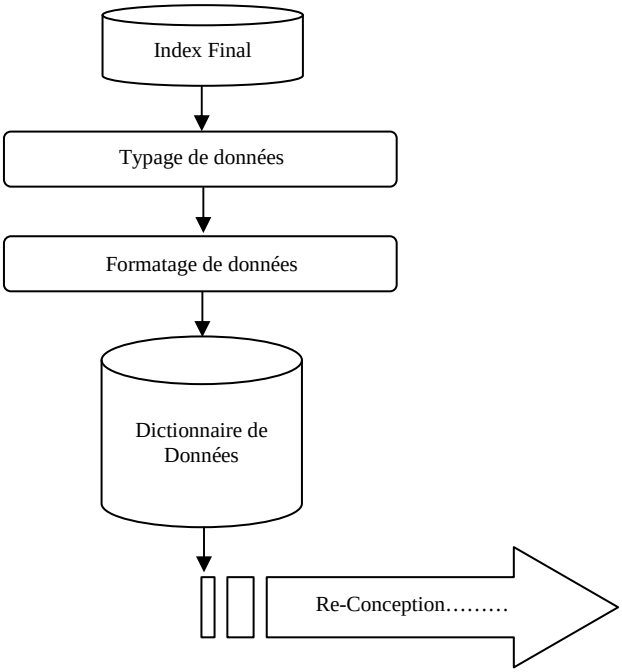

Fig. 21. Phase de Re-conceptualisation.

B.II.2. Application (Cas d'un document XML)

Nous proposons dans cette section, une mise en œuvre de notre approche de rétro-ingénierie des applications orientées web à base d'indexation sémantique. Pour une évaluation empirique, nous avons développé un outil en utilisant EMBARCADERO DELPHI 2010 qui met en œuvre toutes les fonctionnalités de l'approche présentée ci-dessus.

Dans ce qui suit, nous allons présenter les différentes captures d'écran de l'application qui permettent la description des différentes phases de création de l'index sémantique contenant des concepts extraits à partir d'un document XML ou une page HTML, enrichi par d'autres concepts déduits à partir des outils WordNet, TreeTagger et une ontologie de domaine [Dennai et Benslimane, 2015].

B.II.2.1. Phase de modélisation

Cette approche de rétro-ingénierie à base d'indexation sémantique commence par la phase de modélisation. Comme exemple, nous allons prendre un document XML, nous le chargeons en cliquant sur le bouton « Load XML file » (cf. figure n° 22).

Nous générons, en premier lieu, l'arbre libellé, puis nous déduisons l'arbre minimal en cliquant respectivement sur « Generate labeled tree » et « Deduct minimal tree ». Nous pourrons visualiser les résultats des deux actions à côté (cf. figure n° 23).

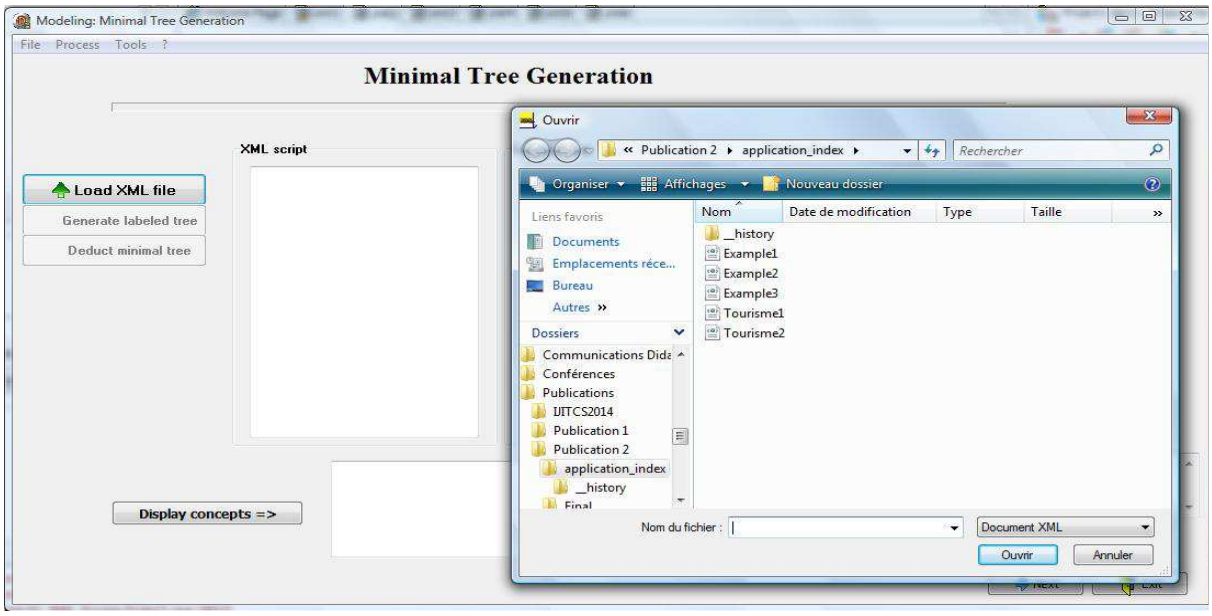

Fig. 22. Chargement d'un document XML pour modélisation.

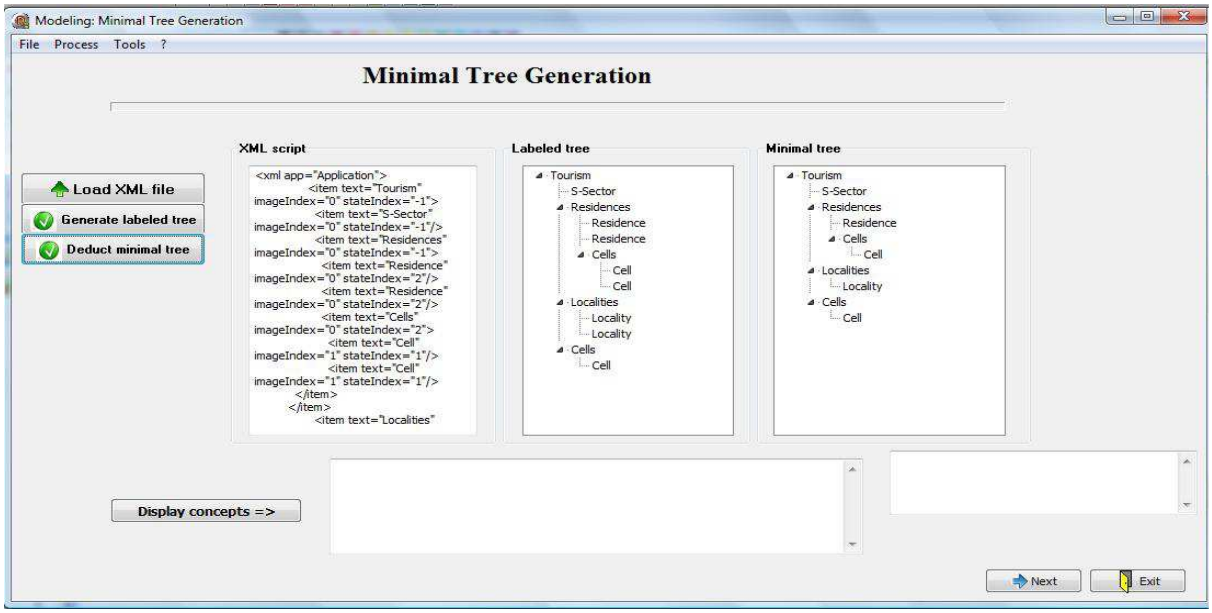

Fig. 23. Arbre minimal généré.

Comme le montre la figure n° 23, il y a des redondances de chemins (Chemins doubles) dans le champ « Labeled tree » : Tourism-Residences-Residence, Tourism-Residences-Cells-Cell et Tourism-Localities-Locality. Nous les avons éliminés, en cliquant sur « Deduct minimal tree », pour obtenir des chemins uniques appelés unités d'information ou bien unités sémantiques (Voir le champ « Minimal tree »).

À la fin, nous pouvons afficher les concepts non redondants du document XML (Tous les nœuds de l'arbre minimal) en cliquant sur le bouton « Display concepts » (cf. figure n° 24).

;

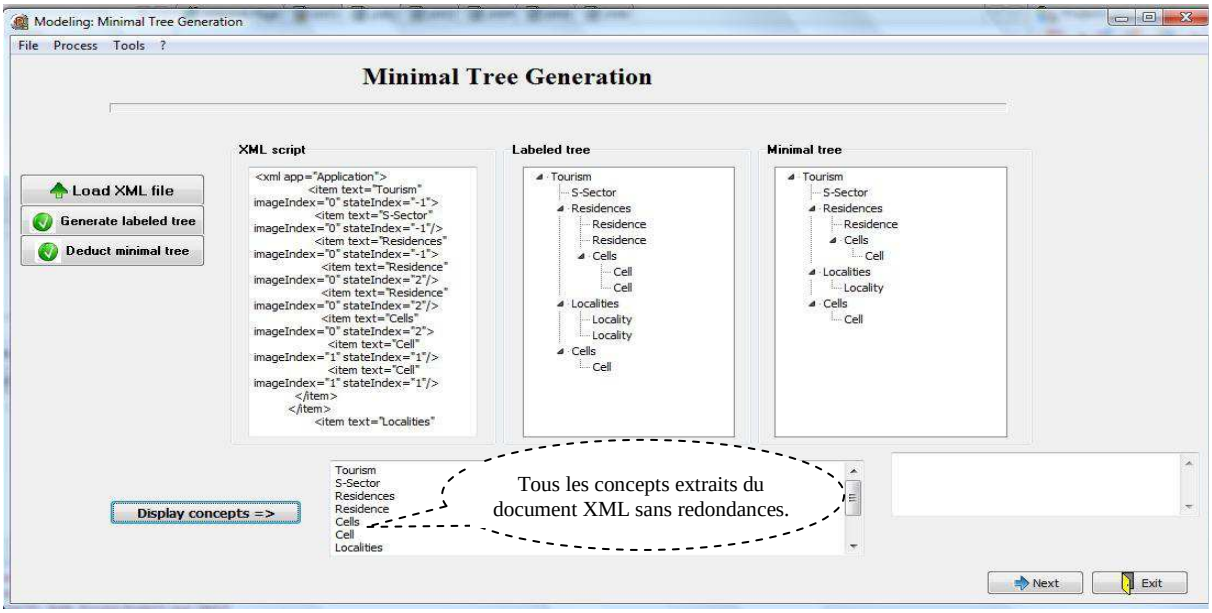

Fig. 24. Résultat final de la modélisation.

B.II.2.2. Phase d'attachement sémantique pour validation

Pour l'accomplissement de cette phase, nous avons besoin d'une ontologie du même domaine que le document XML en question. Nous commençons cette phase par le chargement de l'ontologie (*.owl) en cliquant sur le bouton « Load » (cf. figure n° 25).

Remarque :

Nous avons la possibilité d'utiliser les opérations courantes d'édition à savoir : Cut, Copy et Paste.

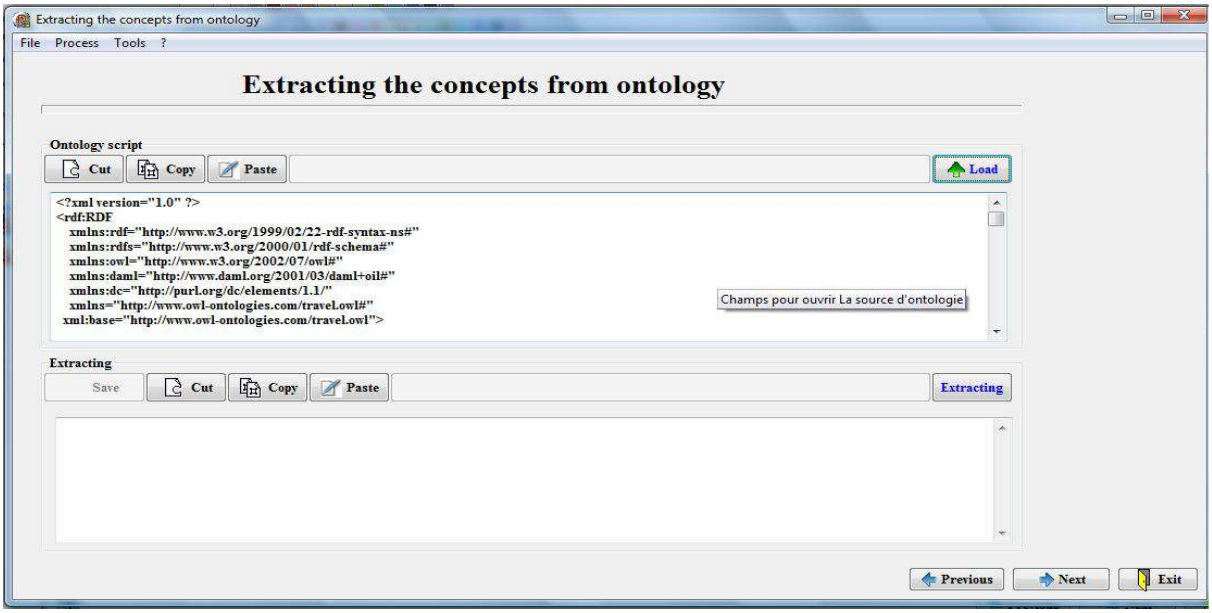

Fig. 25. Chargement de l'ontologie de domaine.

Par la suite, nous lançons l'opération d'extraction de tous les concepts de l'ontologie en cliquant sur le bouton « Extracting » et nous obtenons le résultat tel qu'il est représenté dans la figure n° 26.

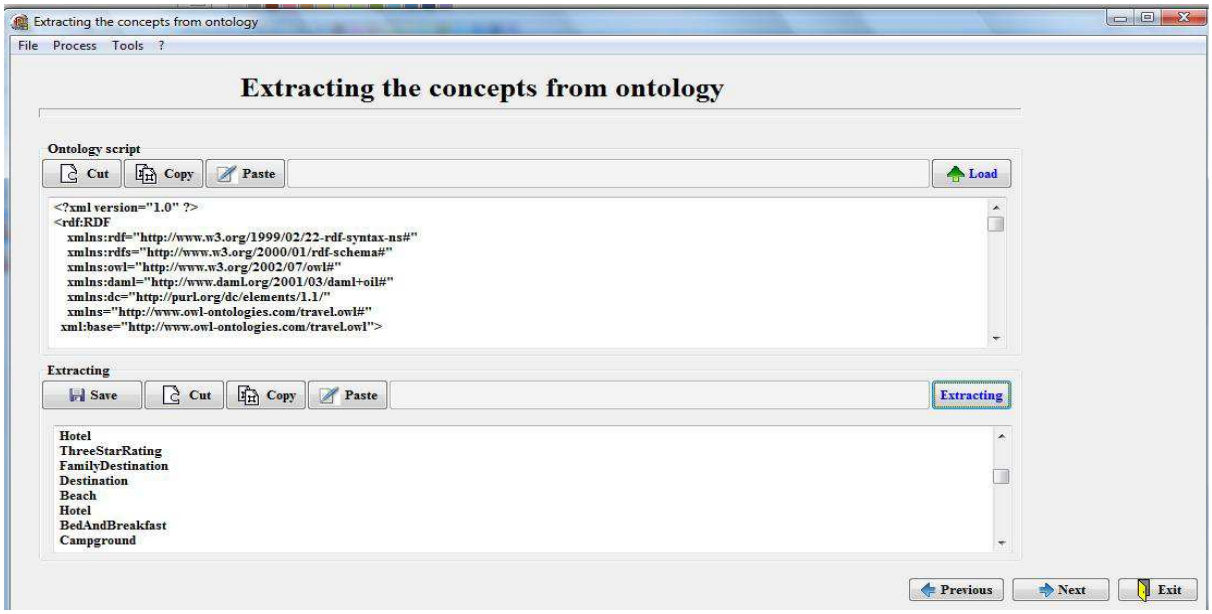

Fig. 26. Extraction des concepts de l'ontologie.

Le cadre de cette phase est d'effectuer un attachement sémantique entre les concepts extraits du document XML et ceux de l'ontologie en utilisation le calcul de la mesure de similarité sémantique. Cette action est exécutée en cliquant sur « Next » dans la figure n° 26).

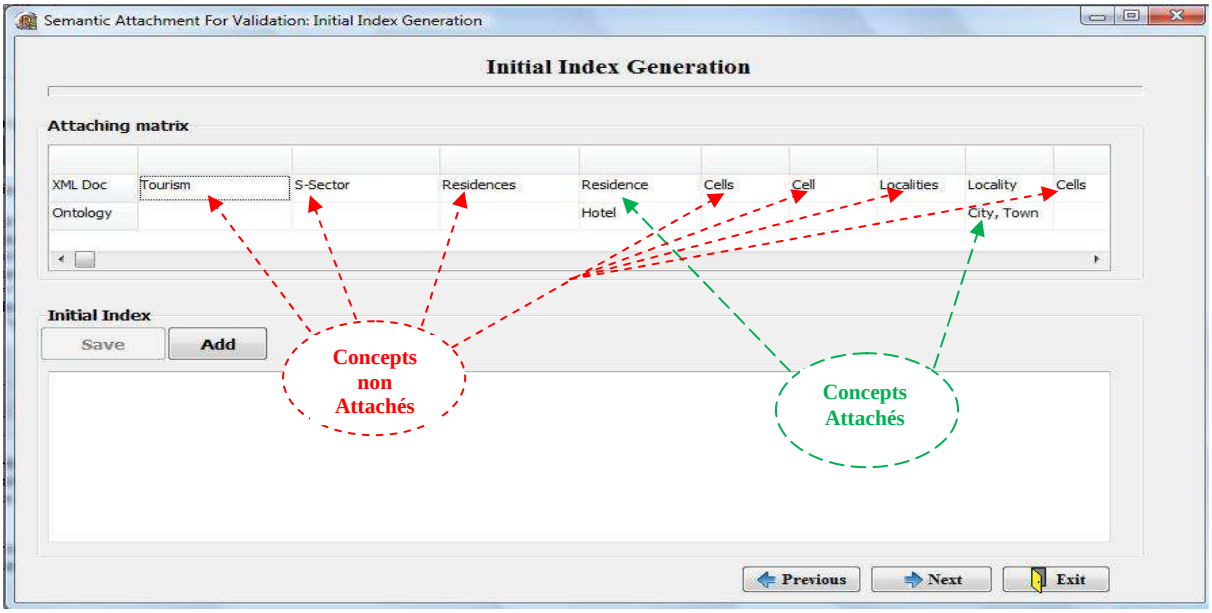

Fig. 27. Attachement sémantique des concepts Document XML - Ontologie.

La figure n° 27 montre les concepts attachés et ceux qui ne sont pas attachés en se basant sur la distance sémantique entre eux. Un index initial contenant les concepts attachés sera créé en cliquant sur « Add » et « Save » (cf. figure n° 28).

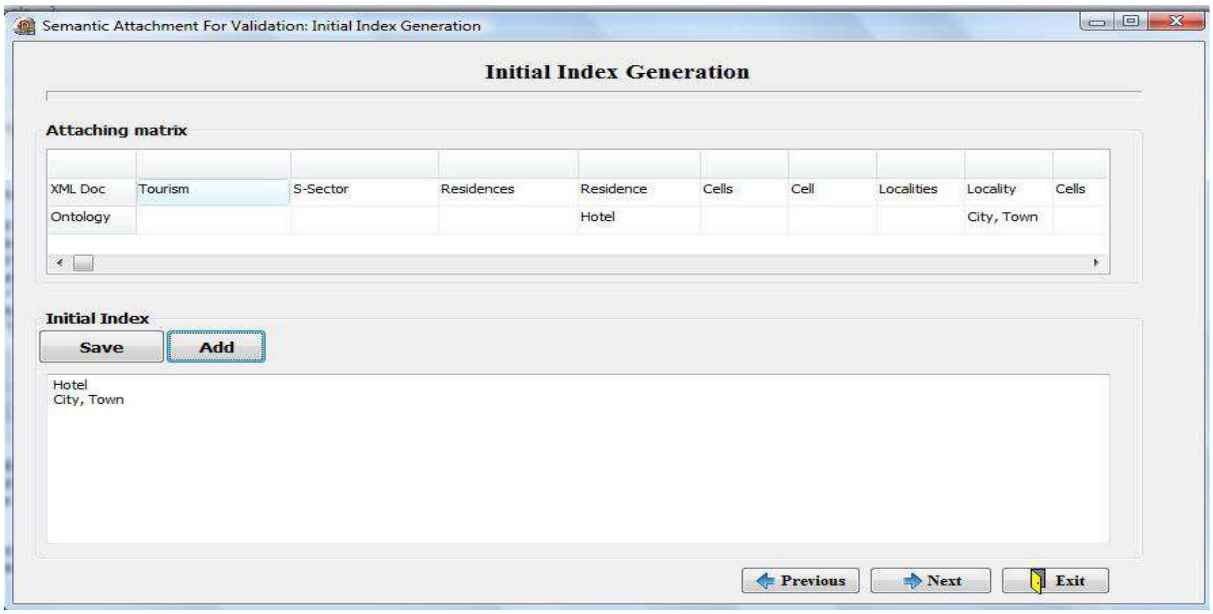

Fig. 28. Création de l'index initial.

Pour les concepts non attachés, nous allons faire une recherche dans WordNet afin d'avoir leurs similarités puis les attacher avec les termes de l'arbre minimal et enfin les intégrer dans l'index (cf. figure n° 29).

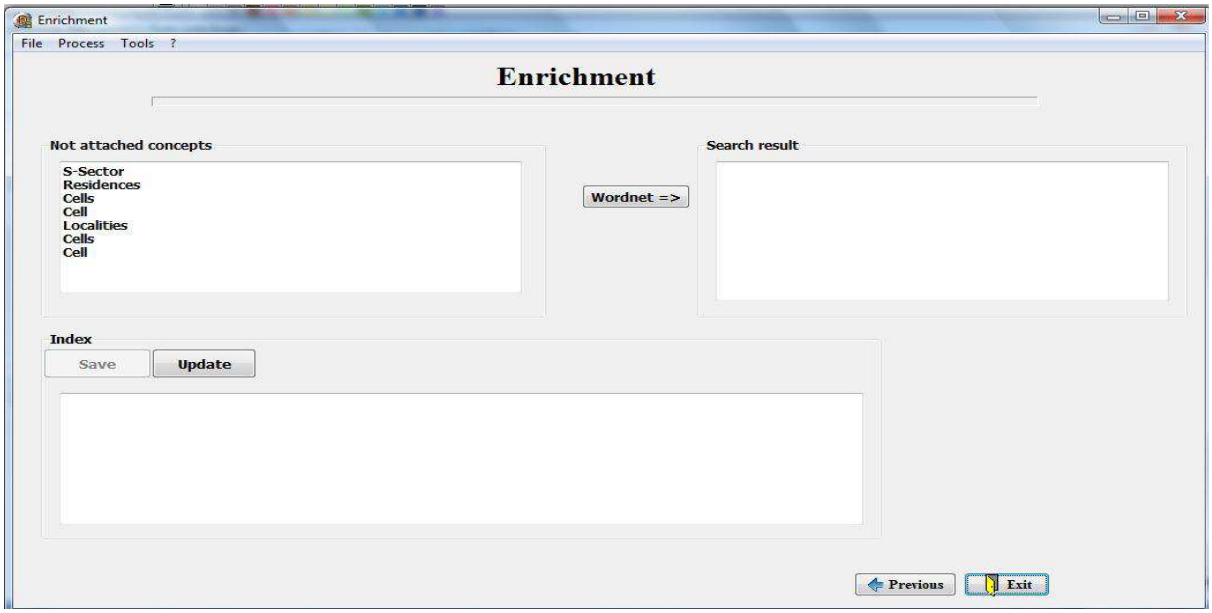

Fig. 29. Recherche manuelle dans WordNet.

Remarque :

La recherche dans WordNet n'a pas abouti pour les concepts non attachés, dans ce cas, nous gardons que le résultat du 1er attachement (cf. figure n° 30).

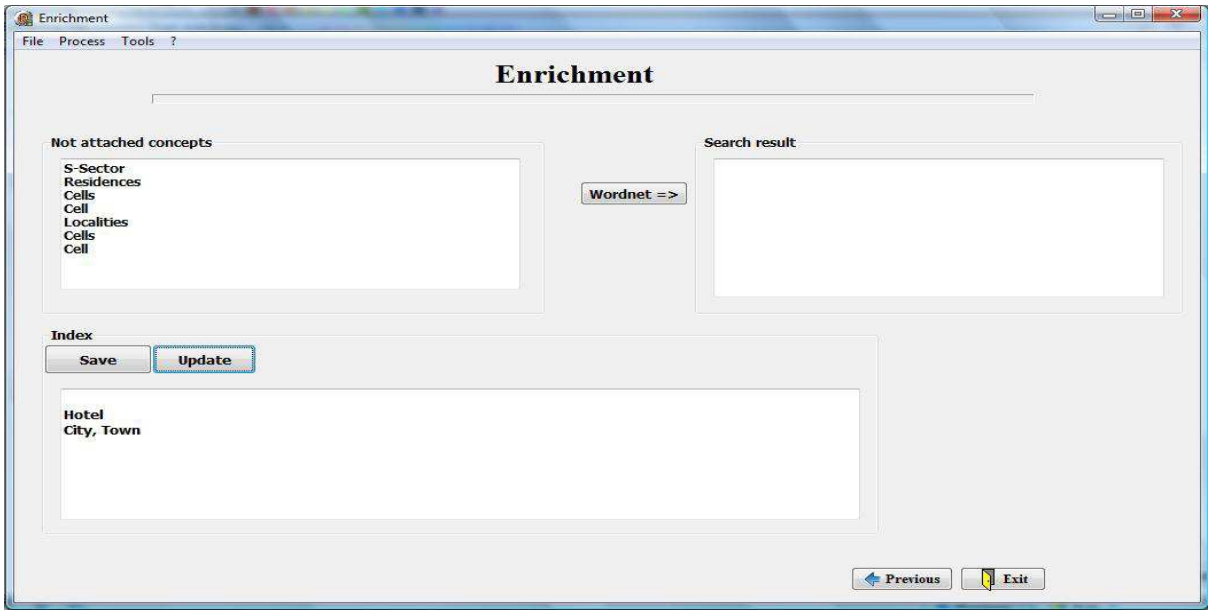

Fig. 30. Contenu de l'index après attachement.

B.II.3. Évaluation

Afin de démontrer la fiabilité de l'approche proposée, nous allons mener un ensemble d'expérimentations. Nous exécutons l'indexation sémantique pour six documents web pris du domaine du tourisme (Trois documents XML et trois pages HTML). Les phases d'attachement sémantique et d'enrichissement sont performantes en utilisant une ontologie tutorial de tourisme pour le web sémantique[12] [Dennai et Benslimane, 2015].

Taille de l'index (Nombrede concepts)			
1ère Expérience (Fichiers XML)			
ExempleN°	Après Modélisation	Après Attachement Sémantique	Après Enrichissement
1	--	**05**	**10**
2	--	**04**	**07**
3	--	**04**	**09**
2ème Expérience (Fichiers HTML)			
Exemple N°	Après Modélisation	Après Attachement Sémantique	Après Enrichissement
1	--	**07**	**11**
2	--	**06**	**08**
3	--	**04**	**07**

Tab. 4. Le contenu de l'index est en croissance.

[12] http://protege.cim3.net/file/pub/ontologies/travel/travel.owl

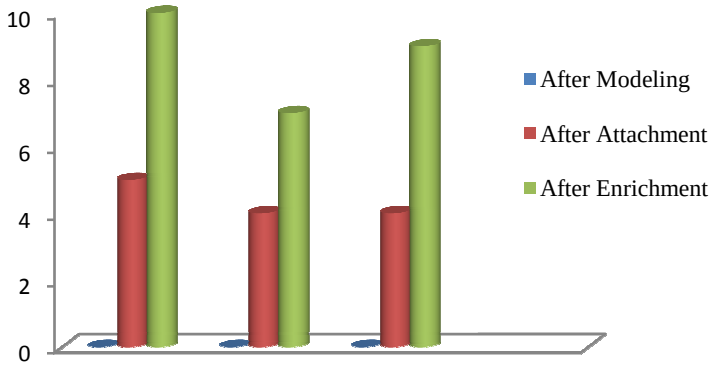

Fig. 31. Résultats de la 1ère expérience.

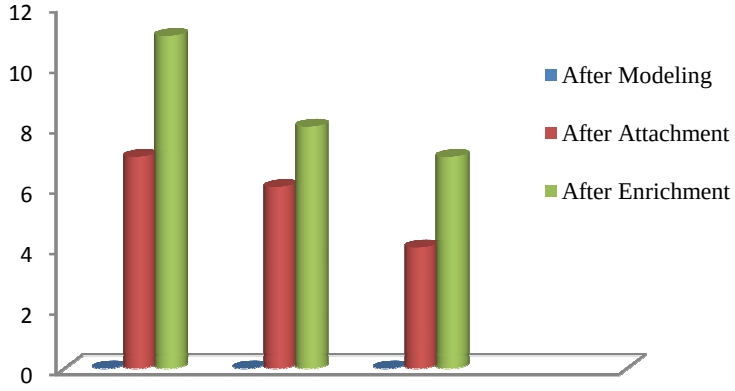

Fig. 32. Résultats de la 2ème expérience.

En lisant les résultats des deux graphes (Figures n° 31 et 32), nous pouvons déduire que le contenu de l'index augmente de plus en plus quand on :

1. Exécute successivement Les différentes phases de l'approche,
2. Effectue une bonne extraction des concepts à partir des pages HTML ou des documents XML,
3. Utilise une ontologie riche de concepts (Du même domaine que le document XML ou la page HTML),
4. Utilise une des mesures de similarité sémantique basée sur les arcs entre les concepts dans la même ontologie (Phase d'enrichissement),
5. Utilise le calcul de la distance sémantique entre un concept d'un document XML ou une page HTML avec un autre d'une ontologie de domaine (Phase d'attachement sémantique).

B.II.4. Conclusion

Ce chapitre était le dernier dans notre démarche de travail, il a présenté, en premier lieu théoriquement les différentes phases du processus de rétro-ingénierie des applications orientées web à base d'une approche d'indexation sémantique en utilisant une ontologie de domaine, en deuxième lieu l'implémentation du dit processus. Ce processus que nous avons proposé permet, une fois exécuté et achevé, de démarrer une nouvelle conception d'un système existant basé sur des pages HTML et des documents XML.

Notre démarche est semi-automatique. En effet, l'utilisateur a la possibilité d'intervenir d'une part sur l'enrichissement des concepts de l'index en interrogeant l'outil WordNet et d'autre part sur la sélection d'autres termes en consultant l'outil TreeTagger.

Conclusion Générale et Perspectives

Sommaire

- **Conclusion Générale**

Les applications orientées web sont devenues les moyens de communication les plus importants pour les entreprises commerciales de toutes sortes. Cependant, la plupart de ces applications sont construites dans l'urgence. Pour écourter les délais de développement, la phase de conceptualisation est souvent sacrifiée et la documentation associée est négligée. En outre, en phase d'exploitation, ces applications sont modifiées au fil des besoins. Elles subissent diverses dégradations touchant aussi bien leur contenu informatif que leur structure de navigation.

Dans ce travail, nous avons présenté un processus de rétro-ingénierie des applications orientées web à base d'indexation sémantique, cette dernière qui, à son rôle, est basée sur une ontologie de domaine. Notre approche d'indexation sémantique débute par l'extraction des concepts utiles à partir d'une page HTML ou d'un document XML afin d'avoir une modélisation de ces deux sources d'informations web. L'ontologie de domaine s'intègre dans le processus, d'une part, pour valider les concepts extraits et d'autre part pour enrichir l'index final. Ce parcours qui, à partir d'une page HTML ou d'un document XML (Considérés comme l'un des constituants d'une application web), aura un index sémantique transformé en dictionnaire de données qui permet à la fin d'avoir une meilleure réingénierie de ces applications et une facilitation dans leur maintenance par la suite.

La pertinence de ce processus s'accroît, en plus, en réalisant une meilleure modélisation des documents web et en utilisant une ontologie riche de concepts.

- **Perspectives**

Nos perspectives et celles de ceux qui s'intéressent à cette recherche se résument en quatre points, qui nous paraissent, importants dans la génération de l'index final et qui peut être transformé en un dictionnaire de données pour le démarrage d'une nouvelle conception d'un système d'information :

1. Utilisation de plusieurs documents web en même temps et par la suite la désignation de documents pertinents,
2. Mise à jour de la mesure de similarité sémantique de Wu et Palmer [Wu et Palmer, 1994], utilisée lors de la phase d'enrichissement de notre processus. La mesure de Wu et Palmer est intéressante mais présente une limite car elle vise essentiellement à détecter la similarité entre deux concepts par rapport à leur distance de leur PPG. Plus ce subsumant est général, moins ils sont similaires (et inversement). Cependant, elle ne capte pas les mêmes similarités que la similarité conceptuelle symbolique. Ainsi on peut avoir Sim(A, f) < Sim(A,B), f étant un des fils de A et B un des frères de A. Ce qui est à notre sens inadéquat dans le cadre de recherche d'information où il faut ramener tous les fils d'un concept (i.e requête) avant son voisinage. Cette mesure présente l'avantage de la rapidité du temps d'exécution, mais l'inconvénient de la production d'une valeur de similarité de deux concepts voisins qui dépassent la valeur de deux concepts dans la même hiérarchie.
3. Fusionnement ou plutôt alignement de plusieurs ontologies de domaine afin d'avoir une ontologie riche de concepts ce qui influence positivement sur le contenu final de l'index,
4. Intégration d'autres sources d'informations et de données dans le processus de rétro-ingénierie, si c'est possible, comme les tables de base de données par exemple.

Références Bibliographiques

[Adda et al., 1999] : Adda G., Mariani J., Paroubek P., Rajman M. et Lecomte J., "L'action GRACE d'évaluation de l'assignation des parties du discours pour le français", Langues, Vol. 2 Issue 1, p. 119-129, 1999.

[Ahcene, 2005] : Ahcene B., "Construction d'une mémoire organisationnelle de formation et évaluation dans un contexte E-Learning : Le projet MEMORAE", thèse doctorat, Université de technologie de Compiègne France, 2005.

[Akoka et Comyn-Wattiau, 2001] : Akoka J. et Comyn-Wattiau I., "La rétro-conception des bases de données et des systèmes de fichiers : Un état de l'art", Ingénierie des Systèmes d'Information, Vol. 6, 2001.

[Baake et al., 2006] : Baake M., Grimm U. et Giegerich R., "Surprises in approximating levenshtein distances", Journal of Theoretical Biology, p. 279-282, 2006.

[Baeza-Yates et Ribeiro-Neto, 1999] : Baeza-Yates R. et Ribeiro-Neto B., "Modern Information Retrieval", ACM Press, Addison-Wesley: New York, Harlow, England Reading, Mass., 1999.

[Bavi et al., 2010] : Bavi V., Beirne T., Bone N., Mohr J. et Neal B., "Comparison of document similarity metrics", Computer Science Department, Western Washington University, Information Retrieval, Winter 2010.

[Bechhofer, 2002] : Bechhofer S., "Ontology language standardization efforts", IST Project IST-2000-29243 OntoWeb, Université de Manchester, UK, 2002.

[Bechhofer et al., 2001] : Bechhofer S., Horrocks 1., Gobie C. et Stevens R., "OilEd: A Reason-able Ontology Editor for the Semantic Web". K12001, Joint German!Austrian conference on Artificial Intelligence, 2001.

[Belletini et al., 2004] : Bellettini C., Marchetto A. et Trentini A., "Webuml: Reverse engineering of web applications", 19th ACM Symposium on Applied Computing SAC2004, Nicosie, Chypre, p. 1662–1669, 2004.

[Bencheikh et al., 2009] : Bencheikh A., Rieu D. et Front A., "Une méthode de rétro-ingénierie des processus métier basée sur un méta-modèle multi-vues", Actes du XXVII° congrès INFORSID, Toulouse, France, Mai 2009.

[Benslimane, 2007] : Benslimane S. M., "Réingénierie des applications web vers le web sémantique : Approche dirigée par l'analyse de formulaires HTML", thèse doctorat, Université Djillali Liabes Sidi Bel Abbes Algérie, 2007.

[Berners-Lee et al., 2001] : Berners-Lee T., Hendler J. A. et Lassila O., "The semantic web", Scientific American, Vol. 5, p. 35-43, 2001.

[Berners-Lee et Fischetti, 2000] : Berners-Lee T. et Fischetti M., "Weaving the web: the past, present and future of the World Wide Web by its inventor", p. 45-46 (ISBN 978-1-58799-018-2), Londres, UK, Texere, 2000.

[Besbes, 2008] : Besbes S., "RetroWeb : Une approche de rétro-conception des sites web", Google livres, 2008.

[Bisson, 2000] : Bisson G., "La similarité : Une notion symbolique/numérique. Apprentissage symbolique - numérique", Tome 2, Editions CEPADUES, 2000.

[Bisson et Hussain, 2008] : Bisson G. et Hussain S. F. "Chi-sim : A new similarity measure for the co-clustering task", 7th International Conference on Machine Learning and Applications ICMLA, IEEE Computer Society, p. 211-217, 2008.

[Blazquez et al., 1998] : Blazquez M., Fernandez-Lopez M., Garcia-Pinar j. M. et Gomez-Pérez A., "Building Ontologies at the Knowledge Level using the Ontology Design Environment", Proceedings 11th Kaw, 1998.

[Blei et al., 2003] : Blei D. M., Ng A. Y. et Jordan M. I., "Latent dirichlet allocation", Journal of Machine Learning Research, Vol. 3, p. 993-1022, 2003.

[Borst, 1997] : Borst W. N., "Construction of engineering ontologies", Center for Telematica and Information Technology, University of Twenty, Enschede, Pays Bas, 1997.

[Boubekeur, 2008] : Boubekeur A. F., "Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets", thèse doctorat, université Toulouse III FRANCE, 2008.

[Bouchiha, 2005] : Bouchiha D., "Rétro-ingénierie des applications web à base d'ontologie", Mémoire magister informatique, Université Djillali Liabes, Sidi Bel Abbes, Algérie, 2005.

[Bouillon et Vanderdonckt, 2004] : Bouillon L. et Vanderdonckt J., "Rétro-ingénierie du modèle de présentation pour les pages Web", Revue d'Interaction Homme Machine (RIHM), Vol. 5 Issue 2, 2004.

[Budanitsky et Hirst, 2006] : Budanitsky A. et Hirst G., "Evaluating WordNet-based measures of lexical semantic relatedness", Computational Linguistics, MIT Press Journals, Vol. 32 Issue 1, p. 13-47, 2006.

[Chagheri et al., 2009] : Chagheri S., Roussey C., Calabretto S. et Dumoulin C., "Semantic indexing of technical documentation", LIRIS, 2009.

[Chaumartin, 2007] : Chaumartin F. R., "WordNet et son écosystème : Un ensemble de ressources linguistiques de large couverture", Colloque BD Lexicales, Université Montréal, Canada, 2007.

[Chiang, 1995] : Chiang R. H. L., "A knowledge-based system for performing reverse engineering of relational databases", Decis. Support Syst., Vol. 13 Issue 3-4, p. 295-312, 1995.

[Chikofsky et Cross, 1990] : Chikofsky E. et Cross J.I., “Reverse engineering and design recovery: A taxonomy”, IEEE software, 1990.

[Cilibrasi et Vitanyi, 2007] : Cilibrasi R. L. et Vitanyi P. M. B. “The google similarity distance”, IEEE Transactions, Knowledge and Data Engineering, Vol. 19 Issue 3, p. 370-383, 2007.

[Conallen, 1999] : Conallen J., “Building Web Applications with UML: Object technology”, Addison-Wesley Longman, Reading, Massachusetts, USA, 1st edition, Dec. 1999.

[Conesa et Olivé, 2004] : Conesa J. et Olivé A., “Pruning ontologies in the development of conceptual schemas of information systems”, ER’2004, LNCS 3288, p. 122–135, 2004.

[Corley et Mihalcea, 2005] : Corley C. et Mihalcea R., “Measuring the semantic similarity of texts”, Proceedings ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, EMSEE '05, p. 13-18, Stroudsburg PA USA, 2005.

[Dahmani, 2010] : Dahmani. F., “Modélisation basée ontologies pour l’apprentissage interactif - application à l’évaluation des connaissances de l’apprenant”, thèse doctorat, université mouloud Mammeri de Tizi Ouzou, Algérie, 2010.

[Decourselle, 2013] : Decourselle J., “Application de rétro-ingénierie pour la consAtruction de schémas Entité/Association à partir d’un modèle relationnel”, UCBL Master Informatique, 2013.

[Dennai et Benslimane, 2011] : DENNAI A. et BENSLIMANE S. M., “Nouvelle version d’une mesure de similarité pour un meilleur calcul de la distance sémantique entre concepts d’une ontologie”, 7ème Colloque sur l’Optimisation et les Systèmes d’Information COSI’2011, Guelma, Algérie, 24-28 Avril 2011.

[Dennai et Benslimane, 2012] : DENNAI A. et BENSLIMANE S. M., “Information extraction from HTML pages or XML documents by a semantic indexing, using domain ontology”, 3rd International Conference on Multimedia Computing and Systems ICMCS’2012, IEEE conference, Tangier, Morocco, 10- 12 Mai 2012.

[Dennai et Benslimane, 2013] : DENNAI A. et BENSLIMANE S. M., “Toward an Update of a Similarity Measurement for a Better Calculation of the Semantic Distance between Ontology Concepts”, 2nd International Conference on Informatics Engineering & Information Science ICIEIS’2013, Kuala Lumpur, Malaysia, 12- 14 November 2013, publié dans The Society of Digital Information and Wireless Communications (SDIWC), (ISBN N° : 978-0-9891305-2-3), http://paper.researchbib.com/?action=viewList&isbn=978-0-9891305-2-3, Nov-2013.

[Dennai et Benslimane, 2014] : DENNAI A. et BENSLIMANE S. M., “Building a Semantic Index from HTML Pages or XML Documents”, International Conference on Computing Technology and Information Management, ICCTIM 2014, Dubai, E.A.U, 09- 11 April 2014.

[Dennai et Benslimane, 2015] : DENNAI A. et BENSLIMANE S. M., “Semantic Indexing of Web Documents Based on Domain Ontology”, International Journal of Information Technology and Computer Science (IJITCS), ISSN : 2074-9007 (Print), ISSN : 2074-9015 (Online), DOI : 10.5815/ijitcs, Published By: MECS Publisher, IJITCS Vol. 7 Issue 2, Jan. 2015.

[Deerwester et al., 1990] : Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K. et Harshman R. “Indexing by latent semantic analysis”, Journal of the American Society for Information Science, Vol. 41 Issue 6, p. 391-407, 1990.

[Desmontils et Jacquin, 2001] : Desmontils E. et Jacquin C., “Des ontologies pour indexer un site web”, actes des journées francophones d’Ingénierie des Connaissances, Nantes France, 2001.

[Dice, 1945] : Dice L. R., “Measures of the amount of ecologic association between species. Ecology”, Vol. 26 Issue 3, p. 297-302, 1945.

[Di Lucca et al., 2002] : Di Lucca G. A., Fasolino A. R., Pace F., Tramontana P. et De Carlini U., “Ware: A tool for the reverse engineering of web applications”, Proceedings 6th European Conference on Software Maintenance and Reengineering (CSMR2002), Budapest, Hongrie, p. 241-250, 2002.

[Domingue, 1998] : Domingue J. “Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web”. 11th Workshop on KAW'98, Banff, Canada, 1998.

[Ehrig et al., 2004] : Ehrig M., Haase P., Hefke M. et Stojanovic N., “Similarity for ontology-a comprehensive framework”, Workshop on Ontology and Enterprise Modelling: Ingredients for Interoperability In Conjunction with 5ème International Conference on Practical Aspects of Knowledge Management, Vienna Austria, 2004.

[Ebert et al., 1999] : Ebert J., Kullbach B. et Winter A., “Querying as an enabling technology in software reengineering”, Proceedings 3rd European Conference on Software Maintenance and Reengineering CSMR ’99, p. 42, IEEE Computer Society, Washington, DC USA, 1999.

[Elsa, 2013] : Elsa N., “Comparaison de textes, quelques approches …”, Cahier du LAMSADE n° 338, Laboratoire d’Analyses et Modélisation de Systèmes pour l’Aide à la Décision, DAUPHINE Université de PARIS France, 2013.

[El Mabrouka, 2005] : El Mabrouka E.H, “Indexation des documents multilingues d'actualité incluant l’arabe : équivalence interlangues et gestion des connaissances chez les indexeurs”, Thèse doctorat, Institut de la communication, Université Lumière Lyon 2, France, 2005.

[El-Ghalayini et al., 2006] : El-Ghalayini H., Odeh M. et McClatchey R., “Deriving conceptual data models from domain ontologies for bioinformatics”, The 2nd International Conference on Information and Communication Technologies from Theory to Application ICTTA, 2006.

[Erich et John, 1991] : Erich B. et John H., "A software reverse engineering experience", CASCON '10 CASCON First Decade High Impact Papers, (DOI 10.1145/1925805.1925808), 1991.

[Estiévenart et al., 2003] : Estiévenart F., François A., Henrard J. et Hainaut J., "A tool-supported method to extract data and schema from web sites", Proceedings 5th international workshop on Web site evolution, Amsterdam, Pays-Bas, p. 3–11, 2003.

[Farqhuar et al., 1996] : Farqhuar A., Fikes R. et Rice J., "The Ontolingua Server: Tool for Collaborative Ontology Construction". 10th Knowledge Acquisition for Knowledge-Based Systems Workshop, Vol. 44, p. 1-19, 1996.

[Fellbaum, 1998] : Fellbaum C., "WordNet : An Electronic Lexical Database (Language, Speech, and Communication) ", The MIT Press, Ed. illustrated edition, 1998.

[Fernandez-Lopez et al., 1997] : Fernandez-Lopez M., Gomez-Pérez A. et Juristo N., "Methontology: From Ontological Art Toward Ontological Engineering", Spring Symposium Series on Ontological Engineering. AAAI'97, Stanford, California, USA, p. 33-40, 1997.

[Fluher, 1992] : Fluher C., "Le traitement du langage naturel dans la recherche d'information", Dans Interface Intelligente dans l'Information Scientifique et Technique, Klingenthal, INRIA, p. 103-130, 1992.

[Fürst, 2004] : Fürst F., "Contribution à l'ingénierie des ontologiques : Une méthode et un outil d'opérationnalisation", Thèse doctorat, Université de Nantes, France, Nov. 2004.

[Gabrilovich et Markovitch, 2007] : Gabrilovich E. et Markovitch S. "Computing semantic relatedness using Wikipedia-based explicit semantic analysis", Proceedings 20th International Joint Conference on Artificial Intelligence, p. 1606-1611, 2007.

[Gaeremynck et al., 2003] : Gaeremynck Y., Bergman L. D. et Lau T., "More for less: Model recovery from visual interfaces for multi-device application design", A. Press (Ed.), Proceedings International Conference on Intelligent user interfaces, Miami Florida, USA, p. 69–76, 2003.

[Gagnon, 2013] : Gagnon O., "Indexation de documents web à l'aide d'ontologies", Maitrise en sciences appliquées, Ecole Polytechnique de Montréal, CANADA, 2013.

[Gandon, 2002] : Gandon F., "Ontology Engeneering : A survey and a return on experience", Rapport de recherche n° 4396, INRIA, 2002.

[Gerardo et Massimiliano, 2007] : Gerardo C. et Massimiliano D. P., "New Frontiers of Reverse Engineering", FOSE '07 2007 Future of Software Engineering, (ISBN 0-7695-2829-5,DOI 10.1109/FOSE.2007.15, 2007.

[Gibson et Conheeney, 1995] : Gibson M. D. et Conheeney K., "Domain knowledge reuse during requirements engineering", Proceedings 7th International Conference, CAiSE '95 Jyväskylä, Finlande, LNCS 932, p. 283-296, 1995.

[Gillies et Cailliau, 2000] : Gillies J. et Cailliau R., “How the Web was Born: The Story of the World Wide Web”, p. 212-213 (ISBN 978-0-19-286207-5), Oxford, Oxford University Press, 2000.

[Gómez-Pérez et Rojas-Amaya, 1999] : Gómez-Pérez A. et Rojas-Amaya Ma D., “Ontological Reengineering for Reuse”, 11th European Workshop, EKAW’99 Dagstuhl Castle, Allemagne, LNCS 1621, p. 139–156, 1999.

[Grefenstette, 2009] : Grefenstette E., “Analyzing document similarity measures”, Thèse Master, université d’Oxford UK, 2009.

[Gruber, 1993] : Gruber T. R., “A translation approach to portable ontology specifications”, Knowledge Acquisition, Vol. 5, Issue 2, p. 199-220, 1993.

[Guarino et al., 1999] : Guarino N., Masolo C. et Vetere G., “OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yellow Pages and Product Catalogs”, publié en ligne : Academia.edu, Jan. 1999.

[Hadj Henni, 2009] : Hadj Henni M., “Approche ontologique pour la modélisation sémantique, l’indexation et l’interrogation des documents Coraniques”, Mémoire Magister, Ecole Supérieur d’Informatique (E.S.I) Oued-Smar, Alger, 2009.

[Halkidi et al., 2003] : Halkidi M., Nguyen B., Varlamis I. et Vazirgiannis M., ‘’Organizing web document collections based on link semantics”, Journal on Very Large Databases, Special Edition on the Semantic Web, Nov. 2003.

[Hassan, 2002] : Hassan A. E., “Architecture recovery of web applications”, Thèse master mathématique en informatique, Université Waterloo Ontario CANADA, 2002.

[Hazen, 2010] : Hazen T. J., “Direct and latent modeling techniques for computing spoken document similarity”, Spoken Language Technology Workshop SLT, IEEE, p. 366-371, 2010.

[Heflin, 2001] : Heflin J., “Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment”, Thèse Ph.D, Université de Maryland, College Park, 2001.

[Helmut, 1994] : Helmut S., “Probabilistic Part-of-Speech Tagging Using Decision Trees”, Proceedings International Conference on New Methods in Language Processing, Manchester, UK, 1994.

[Helmut, 1995] : Helmut S., “Improvements in Part-of-Speech Tagging with an Application to German”, Proceedings ACL SIGDAT-Workshop, Dublin, Ireland, 1995.

[Hirst et St-Onge, 1998] : Hirst G. et St-Onge D., “Lexical chains as representations of context for the detection and correction of malapropisms”, Christiane Fellbaum editor, Cambridge, MA:The MIT Press, 1998.

[Hofmann, 1999] : Hofmann T., "Probabilistic latent semantic indexing", Proceedings 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99, p. 50-57, ACM, New York USA, 1999.

[Huang, 2008] : Huang A., "Similarity Measures for Text Document Clustering", dans Holland J., Nicholas A. et Brignoli D., editors, New Zealand Computer Science Research Student Conference, p. 49-56, 2008.

[Hubert et Valérie, 2003] : Hubert K. et Valérie M., "Les web services. Techniques, démarches et outils XML, WSDL, SOAP, UDDI, RosettaNet, UML", Dunod 2003.

[Jaccard, 1901] : Jaccard P., "Étude comparative de la distribution florale dans une portion des alpes et des jura", Bulletin de l la Société Vaudoise des Sciences Naturelles, Vol. 37, p. 547-579, 1901.

[Jiang et Conrath, 1997] : Jiang J. J. et Conrath D. W., "Semantic similarity based on corpus statistics and lexical taxonomy", Proceedings International Conference Research on Computational Linguistics (ROCLING X), Taiwan, 1997.

[Karen et Boughanem, 2006] : Karen P. S. et Boughanem M., "Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée", Ed. Cépaués, p. 77-98, 2006.

[Kazai et al., 2002] : Kazai G., Lalmas M. et Roelleke T., "Focussed Structured Document Retrieval", 9th International Symposium on String Processing and Information Retrieval (SPIRE), p. 241-247, Lisbon, Portugal, Sep. 2002.

[Khan et al., 2004] : Khan L., Mcleod D. et Hovy E., "Retrieval effectiveness of an ontology-based model for information selection", International Journal on Very Large Data Bases, Vol. 13, p. 71-85, 2004.

[Krötzsch, 2010] : Krötzsch M., "Description Logic Rules", IOS Press., ISBN 978-1-61499-342-1, p. 10, Oct. 2010.

[Lalmas, 2000] : Lalmas M., "Uniform representation of content and structure for structured document retrieval", 20th SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, p. 215-228, Cambridge, UK, Déc. 2000.

[Leacock et Chodorow, 1998] : Leacock C. et Chodorow M., "Combining Local Context and WordNet Similarity for Word Sense Identification, in WordNet: An Electronic Lexical Database", MIT Press, 1998.

[Lee et al., 1993] : Lee J. H., Kim M.H. et Lee Y. J., "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies", Journal of Documentation, Vol. 49 Issue 2, p. 188-207, 1993.

[Lesk, 1986] : Lesk M, “Automatic sense disambiguation using machine readable dictionaries : how to tell a pine cone from an ice cream cone”, Proceedings 5th Annual International Conference on Systems Documentation SIGDOC '86, p. 24-26, New York USA, ACM Press, 1986.

[Levenshtein, 1966] : Levenshtein V., “Binary codes capable of correcting deletions, insertions and reversals”, Soviet Physics Doklady, Vol. 10 Issue 8, p. 707-710, 1966.

[Lin, 1998] : Lin D., “An Information-Theoretic Definition of similarity”, Proceedings 15ème International Conference on Machine Learning (ICML'98), p. 296-304, Morgan-Kaufmann, San Francisco USA, 1998.

[LP_LBM_EISTI, 2006] : Coopération : Laboratoire Paragraphe – Université Paris 8, Laboratoire de Biologie Moléculaire – Université de Cergy Pontoise et EISTI : Ecole Internationale des Sciences et Traitement de l’Information, “Cours en informatique”, Ethnoinformatique : http:// www.ethnoinformatique.fr /course/, Mis en ligne 11 Déc. 2006, (Consulté Juin. 2014).

[Luhn, 1957] : Luhn H., “A statistical approach to mechanized encoding and searching of literary information”. IBM Journal of Research and Development, Vol. 4 Issue 1, p. 309-317, 1957.

[MacGregor, 1991] : MacGregor R., “Inside the LOOM Description Classifier”, SIGART Bulletin, Vol. 2 Issue 3, p. 88-92, 1991.

[Maedche, 2002] : Maedche A., “Ontology Learning for the Semantic Web”, Boston: Kluwer Academic Publishers, 2002.

[Maedche et al., 2002] : Maedche A., Motik B., Stojanovic L., Studer R. et Volz R., “Ontologies for Enterprise Knowledge Management”, IEEE Intelligent Systems, Vol. 18 Issue 2, p. 26-33, 2003.

[Maniraj et Sivakumar, 2010] : Maniraj V. et Sivakumar R., “Ontology Languages – A Review”, International Journal of Computer Theory and Engineering, Vol. 2 Issue 6, Déc. 2010.

[Maron et Kuhns, 1960] : Maron M. et Kuhns J., “On relevance, probabilistic indexing and information retrieval”. Journal of the Association for Computing Machinery 7, p. 216–244, 1960.

[Mass et Matan, 2003] : Mass Y. et Matan M., “Retrieving the most relevant XML Component”, the Second Workshop of the Initiative for The Evaluation of XML Retrieval, INEX, p. 53-58, 2003.

[Mass et Matan, 2004] : Mass Y. et Matan M., “Component ranking and automatic query refinement for XML retrieval”, INEX 2004, p. 134–140, 2004.

[Mihalcea et al., 2006] : Mihalcea R., Corley C. et Strapparava C, “Corpus-based and knowledge-based measures of text semantic similarity”, Proceedings AAAI.06, p. 775-780, 2006.

[Miller, 1995] : Miller A. G., “Wordnet: A lexical database”, Actes de *ACM*, Vol. 38 Issue 11, p. 39-41, Nov. 1995.

[Miller et al., 1993] : Miller G. A., Beckwith R., Fellbaum C., Gross D. et Miller K., “Introduction to WordNet: An On-line Lexical Database”, Cognitive Science Laboratory, Princeton University, Princeton USA, Technical Report, 1993.

[Mihalcea et Moldovan, 2000] : Mihalcea R. et Moldovan D. “Semantic indexing using WordNet senses”, Proceedings ACL Workshop on IR & NLP, Hong Kong, Oct. 2000.

[Mohammad et Hirst, 2012] : Mohammad S. M. et Hirst G., “Distributional measures of semantic distance : A survey”, CoRR, abs/1203.1858, dblp Computer Science Bibliography, 2012.

[Mohler et Mihalcea, 2009] : Mohler M. et Mihalcea R., “Text-to-text semantic similarity for automatic short answer grading”, Proceedings EACL 12th Conference of the European Chapter of the Association for Computational Linguistics, p. 567-575, Stroudsburg USA, 2009.

[Motta, 1999] : Motta E., “Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving”, lOS Press, 1999.

[Moussa et al., 2001] : Moussa L., Amrane H. et Patrick R., “Un modèle de conception d’application Web basé sur XML”, ISPS’2001 – Alger, Mai. 2001, RIST Vol. 11 Issue 1, 2001.

[Muller et al., 2000] : Muller H. A., Jahnke J. H., Smith D. B., Storey M. A., Tilley S. R. et Wong K., “Reverse engineering : A roadmap”, Proceedings International Conference on The Future of Software Engineering ICSE ’00, ACM Press, p. 47-60, New York USA, 2000.

[Musen et al., 2000] : Musen M. A., Fergerson R. W., Grosso W. E., Noy N. F., Crubezy M. et Gennari J. H., “Component-Based Support for Building Knowledge-Acquisition Systems”, Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress, Beijing, 2000.

[Myaeng et al., 1998] : Myaeng S. H., Jang D.-H., Kim M.-S. et Zhoo Z.-C., “A Flexible Model for Retrieval of SGML documents”, Proceedings 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 138–140, Melbourne, Australia, ACM Press, New York, 1998.

[Myers et al., 1996] : Myers B., Hollan J., Cruz I., Bryson S., Bulterman D., Catarci T., Citrin W., Glinert E., Grudin J. et Ioannidis Y., “Strategic directions in human computer interaction”, ACM Computing Surveys, Vol. 28 Issue 4, p. 794–809, 1996.

[Navarro, 2001] : Navarro G., “A guided tour to approximate string matching”, ACM Computing Surveys, Vol. 33 Issue 1, p. 31-88, 2001.

[Neches et al., 1991] : Neches R., F. R. E., Finin T., Gruber T. R., Senator T. et Swartout W. R, “Enabling technology for knowledge sharing”, AI Magazine, Vol. 12, p. 35-56, 1991.

[Nie, 2003] : Nie J-Y., “Le domaine de la recherche d’information, survol d’une longue histoire”, Gaussier Ed., Assistance intelligente à la recherche d’information, Collection Traité des sciences et techniques de l’information, Paris, Lavoisier, p.19-28, 2003.

[Papineni et al., 2002] : Papineni K., Roukos S., Ward T. et jing Zhu W, “Bleu : A method for automatic evaluation of machine translation”, Proceedings ACL '02 40th Annual Meeting on Association for Computational Linguistics, p. 311-318, Stroudsburg USA, 2002.

[Patwardhan et al., 2003] : Patwardhan S., Banerjee S. et Pedersen T., “Using measures of semantic relatedness for word sense disambiguation”, Computer Science, Vol. 2588, p. 241-257, 2003.

[Pedersen et al., 2007] : Pedersen T., Pakhomov S. V. S., Patwardhan S. et Chute C. G., “Measures of semantic similarity and relatedness in the biomedical domain”, Journal of Biomedical Informatics, Vol. 40, p. 288–299, 2007.

[Peterson et al., 1998] : Peterson B. J., Andersen W. A. et Engel J., “Knowledge Bus: Generating application-focused databases from large ontologies”, Proceedings 5th KRDB Workshop, Seattle, WA, 1998.

[Pomart et Sutter, 1997] : Pomart P.D. et Sutter E., “Indexation”, Article du Dictionnaire Encyclopédique de l’Information et de la Documentation, Paris, Nathan, p. 284-287, 1997.

[Rada et al., 1989] : Rada R., Mili H., Bicknell E. et Blettner M., “Development and application of a metric on semantic nets”, IEEE Transaction on Systems, Man and Cybernetics, Vol. 5 Issue 1, p. 17-30, 1989.

[Resnik, 1995] : Resnik P., “Using information content to evaluate semantic similarity in a taxonomy”, Proceedings International Joint Conference for Artificial Intelligence (IJCAI), p. 448-453, 1995.

[Resnik, 1999] : Resnik P., “Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language”, Journal of Artificial Intelligence Research, Vol. 11, p. 95-130, 1999.

[Rocchio, 1971] : Rocchio J., “Relevance Feedback in Information Retrieval”, Ed. G. Salton, the SMART retrieval system – Experiments in automatic document processing, Prentice Hall Inc., Englewood Cliffs, p. 313-323, 1971.

[Roussey et al., 1999] : Roussey C., Calabretto S. et Pinon J-M., “Etat de l’art en indexation et recherche d’information”, Document Numérique, N° spécial : Gestion des documents et gestion des connaissances, Vol. 3, Issue 3 et 4, p. 121-150, Déc. 1999.

[Sagot et Fišer, 2008] : Sagot B. et Fišer D., “Building a free French WordNet from multilingual resources”, dans OntoLex, Marrakech Maroc, 2008.

[Salton, 1968] : Salton G., "Automatic Information Organization and Retrieval", McGraw.Hill Book Company, New York USA, 1968.

[Salton et McGill, 1986] : Salton G. et McGill M. J., "Introduction to modern information retrieval", Livre, McGraw-Hill Inc, New York USA, 1986.

[Savoy, 2005] : Savoy J., "Indexation manuelle et automatique : Une évaluation comparative basée sur un corpus en langue française", Prceeding 2ème Conférence Francophone en Recherche d'Information et Applications - CORIA 2005, Grenoble, France, Mars. 2005.

[Schileder et Holger, 2002] : Schileder T. et Holger M., "Querying and ranking XML documents", Journal of the American Society for Information Science and Technology, p. 489-503, 2002.

[Schreiber et al., 1995] : Schreiber G., Wielinga B. et Jansweijer W., "The KACTUS view on the'O'word", Proceedings IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, p. 159-168, 1995.

[Slimani et al., 2008] : Slimani T., Ben Yaghlane B. et Mellouli K., "A New Similarity Measure based on Edge Counting", Proceedings World Academy of Science, Engineering and Technology, Vol. 23, p. 773-777, 2008.

[Strehl et al., 2000] : Strehl A., Ghosh J. et Mooney R., "Impact of Similarity Measures on Web-page Clustering", Proceedings 17th National Conference on Artificial Intelligence : Workshop of Artificial Intelligence for Web Search (AAAI 2000), 30-31 July 2000, Austin Texas USA, p. 58-64, 2000.

[Studer et al., 1998] : Studer R., Benjamins R. et Fensel D., "Knowledge engineering: Principles and methods", Data and Knowledge Engineering, Vol. 25 Issue 1 et 2, p. 161–198, 1998.

[Swartout et al., 1997] : Swartout B., Ramesh P., Knight K. et Russ T., "Towards Distributed Use of Large Scale Ontologies", Spring Symposium Series on Ontological Engineering. AAAI'97, Stanford, California, USA, p. 138-148, 1997.

[Takale et Nandgaonkar, 2010] : Takale S. A. et Nandgaonkar S. S., "Measuring semantic similarity between words using web documents", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 1 Issue 4, p. 78-85, Oct. 2010.

[Tramontana, 2005] : Tramontana P., "Reverse engineering web applications", Ed. IEEE, Proceedings 21st International Conference on Software Maintenance (ICSM05), p. 705–708, 2005.

[Uschold et King, 1995] : Uschold M. et King M., "Towards a Methodology for Building Ontologies". Workshop on Basic Ontological Issues, Knowledge Sharing, 1995.

[Valéry, 2007] : Valéry P., "Rôle des ontologies en ingénierie des EIAH : Cas d'un système d'assistance au design pédagogique", thèse doctorat, université de Québec Montréal CANADA, 2007.

[Valéry et al., 2003] : Valéry P., Olavo M. et Jacqueline B., "Apport de l'ingénierie ontologique aux environnements de formation à distance", Revue STICEF, Vol. 10, 2003, ISSN : 1764-7223, mis en ligne le 5/02/2004, http://sticef.org.

[Valli et Véronis, 1999] : Valli A. et Véronis J. "Étiquetage grammatical des corpus de parole : Problèmes et perspectives", Revue française de linguistique appliquée, Vol. 4 Issue 2, p. 113-133, 1999.

[Vanderdonckt et al., 2001] : Vanderdonckt J., Bouillon L. et Souchon N., "Flexible reverse engineering of web pages with VAQUISTA", Proceedings 8th Working Conference on Reverse Engineering (WCRE'01), p. 241–248, Stuttgart Allemagne, 2001.

[Vasilecas et Bugaite, 2007] : Vasilecas O. et Bugaite D., "An algorithm for the automatic transformation of ontology axioms into a rule model", Proceedings International Conference on Computer Systems and Technologies (CompSysTech '07), Bulgaria, p. 1–6, 2007.

[Volk et al., 2002] : Volk M., Ripplinger B. et Vintar S., "Semantic annotation for concept-based cross-language medical information retrieval", International Journal of Medical Informatics, Vol. 67, p. 1-3, Dec. 2002.

[Volk et al., 2003] : Volk M., Vintar S. et Buitelaar P., "Ontologies in cross-language information retrieval", Proceedings 2nd Conference on Professional Knowledge Management, Lucerne Switzerland, 2003.

[Voorhees, 1993] : Voorhees E. M., "Using WordNet to Disambiguate Word Senses for Text Retrieval", Proceedings 16th annual international ACM, p. 171-180, 1993.

[Wilkinson, 1994] : Wilkinson R., "Effective retrieval of structured documents", Proceedings 17th Annual International Conference on Research and Development in Information Retrieval, p. 311-317, Springer – Verlag, Dublin Ireland, July 1994.

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

[Wong et al., 2006] : Wong W., Liu W. et Bennamoun M., "Featureless similarities for terms clustering using tree-traversing ants", Proceedings International Symposium on Practical Cognitive Agents and Robots, PCAR '06, p. 177-191, ACM, New York USA, 2006.

[Wu et Palmer, 1994] : Wu Z. et Palmer M., "Verb semantics and lexical selection", Proceedings 32nd Annual Meeting of the Associations for Computational Linguistics, p. 133-138, 1994.

[W3C Recommendation, 2002] : W3C Recommendation, “XHTML™ 1.0 The Extensible HyperText Markup Language, 2ème Edition - A Reformulation of HTML 4 in XML 1.0”, http://www.w3.org/TR/2002/REC-xhtml1-20020801, Mis en ligne 26 Jan. 2000, Révisé 1 Août 2002, (Consulté Juin. 2014).

[W3C Recommendation, 1998] : W3C Recommendation, “Langage de balisage extensible”, http://www.w3.org/TR/1998/REC-xml-19980210, Mis en ligne 10 Fév. 1998, (Consulté Juin. 2014).

[W3C Recommendation, 2008] : W3C Recommendation, “eXtensible Markup Language, 5ème Edition”, http://www.w3.org/TR/2008/REC-xml-20081126, édité en ligne 26 Nov. 2008, (Consulté Juin. 2014).

[W3C Ubiquitous Web domain, 2014] : W3C Ubiquitous Web domain, “eXtensible Markup Language (XML)”, http://www.w3.org/XML/, Mis à jour 03.Oct.2014, Consulté Mai. 2013.

[Zargayouna et Saloti, 2004] : Zargayouna H. et Salotti S., “SemIndex: A model of semantic indexing on XML documents”, in 26ème European Conference on Information Retrieval (ECIR'2004), VOL. 2, 2004.

Liste des Figures et des Tableaux

A.1. Liste des figures

A.2. Liste des tableaux

Liste des Sigles et des Abréviations

API	Application Programming Interface.
ASP	Active Server Pages.
BEPs	Best Entry Points.
CGI	Common Gateway Interface.
CMS	Content Management System.
COM	Component Object Model.
CORBA	Common Object Request Broker Architecture.
CSS	Cascading Style Sheets.
DAM	Digital Asset Management.
DC	Dublin Core.
DCOM	Distributed Component Object Model.
DLL	Dynamic Link Library.
DOS	Disk Operating System.
DTD	Document Type Definition.
EAE	Entité Association Entité.
EDI	Environnement de Développement Intégré.
EJB	Enterprise JavaBeans.
ESA	Explicit Semantic Analysis.
FL	Fenêtres Logiques.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
IEF^d	Inverse Element Frequency for document.
IHM	Interface Homme Machine.
JSP	Java Server Pages.
KIF	Knowledge Interchange Format.
LOM	Learning Object Metadata.
LOOM	Language for Object Oriented Methods.
LDA	Latent Dirichlet analysis.
LSA	Latent Semantic Analysis.
MPEG-7	Motion Picture Expert Group 7.
NGD	Normalized Google Distance.

OCML	Operational Conceptual Modeling Language.
ODE	Ontology Design Environment.
OIA	Objets d'Interaction Abstraits.
OIC	Objets d'Interaction Concrets.
OIL	Ontology Inference Layer.
OilEd	Ontology Inference Layer Editor.
OKBC	Open Knowledge Base Connectivity.
OntoEdit	Ontology Editor.
OWL	Web Ontology Language.
PCDATA	Parsed Character DATA.
PDF	Portable Document Format.
Perl	Practical extraction and report language.
PLSA	Probabilistic Latent Semantic Analysis.
PHP	Personal Home Page ou Hypertext Preprocessor.
PL-SQL	Procedural Language-Structured Query Language.
POO	Programmation Orientée Objet.
POS	Part Of Speech.
PSVI	Post-Schema-Validation Infoset.
RAD	Rapid Application Development.
RDF	Resource Description Framework.
RDF(s)	Resource Description Framework Schema.
Relax NG	Regular Language for XML Next Generation.
RTF	Rich Text File.
SGML	Standard Generalized Markup Language.
SHOE	Simple HTML Ontology Extensions.
SRI	Système de Recherche d'Information.
SVD	Singular Value Decomposition.
TAL	Traitement Automatique des Langues
TF-IDF	Term Frequency - Inverse Document Frequency.
TF-ITDF	Term Frequency-Inverse Tag and Document Frequency.
TC	TextCorpora.
UML	Unified Modeling Language.
UP	Unités de Présentation.

UTF-8	Universal Character Set Transformation Format - 8 bits.
VB	Visual Basic.
VBScript	Visual Basic Script.
VBX	Visual Basic eXtension.
VCL	Visual Component Library.
VSM	Vector Space Model.
W3C	World Wide Web Consortium.
WebOde	Web Ontology Design Environment.
WebOnto	Web Ontology.
WWW	World Wide Web.
WYSIWYG	What You See Is What You Get.
XAML	eXtensible Application Markup Language.
XHTML	eXtensible HyperText Markup Language.
XML	eXtended Markup Language.
XML Schema	eXtended Markup Language Schema.
XOL	XML based Ontology exchange Language.
XSD	XML Schema Definition.
XSL	eXtensible Stylsheet Language.

Qu'est ce que Wordnet ?

C.1. Définition

WordNet [Miller, 1995] est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991.

Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Par rapport aux outils fournis, un développeur peut aussi accéder la base de données à partir des interfaces disponibles pour de nombreux langages de programmation [Miller, 1995].

S'il n'est pas exempt de critiques (Granularité très fine, absence de relations paradigmatiques…), WordNet n'en reste pas moins l'une des ressources de TAL (Traitement Automatique des Langues) les plus populaires.

Des versions de WordNet pour d'autres langues existent, mais la version anglaise est cependant la plus complète à ce jour. La dernière version distribuée en avril 2013 est la 3.12. Cette version est par ailleurs consultable en ligne [Chaumartin, 2007].

Fig. 29. Ressources disposant d'une traçabilité vers WordNet [Chaumartin, 2007].

C.2. Notion de synset

Le synset (Ensemble de synonymes) est la composante atomique sur laquelle repose WordNet. Un synset correspond à un groupe de mots interchangeables, dénotant un sens ou un usage particulier. Un synset est défini d'une façon différentielle par les relations qu'il entretient avec les sens voisins [Chaumartin, 2007].

Les noms et les verbes sont organisés en hiérarchie. Des relations d'hyperonymie (Est-un) et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs « spécialisations ». Au niveau racine, ces hiérarchies sont organisées en types de base. Le réseau des noms est bien plus profond que celui des autres parties du discours. A titre indicatif, les deux premiers niveaux de la hiérarchie des noms se constituent des concepts abstraits suivants : [Chaumartin, 2007]

- **ABSTRACTION**: ATTRIBUTE, MEASURE/QUANTITY/AMOUNT, RELATION, SET, SPACE, TIME…
- **HUMAN ACTION**: ACTIVITY, COMMUNICATION, DISTRIBUTION, INACTIVITY, JUDGMENT, LEANING, LEGITIMATION, MOTIVATION, PROCLAMATION, PRODUCTION, SPEECH ACT…
- **ENTITY**: ANTICIPATION, CAUSAL AGENT, ENCLOSURE, EXPANSE, LOCATION, PHYSICAL OBJECT, SKY, SUBSTANCE, THING…
- **EVENT**: GROUP ACTION, NATURAL EVENT, MIGHT-HAVE-BEEN, MIGRATION, MIRACLE, NONEVENT, SOCIAL EVENT…
- **GROUP**, GROUPING: ASSOCIATION, BIOLOGICAL GROUP, PEOPLE, COLLECTION, AGGREGATION, COMMUNITY, ETHNIC GROUP, KINGDOM, MULTITUDE, POPULATION, RACE, RARE-EARTH ELEMENT…
- **PHENOMENON**: EFFECT/RESULT, LEVITATION, FORTUNE/CHANCE, REBIRTH, NATURAL PHENOMENON, PROCESS, PULSATION…
- **POSSESSION**: ASSETS, CIRCUMSTANCES, PROPERTY/MATERIAL POSSESSION, TRANSFERRED PROPERTY, TREASURE…
- **PSYCHOLOGICAL FEATURE**: COGNITION/KNOWLEDGE, FEELING, MOTIVATION/NEED…
- **STATE**: ACTION/ACTIVITY, EXISTENCE, STATE OF MIND, CONDITION, CONFLICT, DAMNATION, DEATH, DEGREE, DEPENDENCY, DISORDER, EMPLOYMENT, END, FREEDOM, ANTAGONISM, IMMATURITY, IMMINENCE, IMPERFECTION, INTEGRITY, MATURITY, OMNIPOTENCE, PERFECTION, PHYSIOLOGICAL STATE, RELATIONSHIP, STATE OF AFFAIRS, STATUS, TEMPORARY STATE, NATURAL STATE…

L'organisation des adjectifs est différente. Un sens « tête » joue un rôle d'attracteur ; des adjectifs « satellites » lui sont reliés par des relations de synonymie. On a donc une partition de l'ensemble des adjectifs en petits groupes. Les adverbes sont le plus souvent définis par les adjectifs dont ils dérivent. Ils héritent donc de la structure des adjectifs [Chaumartin, 2007].

C.3. Ontologies et relations sémantiques

À l'instar d'un dictionnaire traditionnel, WordNet offre ainsi, pour chaque mot, une liste de synsets correspondant à toutes ses acceptions répertoriées. Mais les synsets ont également d'autres usages, ils peuvent représenter des concepts plus abstraits, de plus haut niveau que les mots et leurs sens qu'on peut organiser sous forme d'ontologies.

Une ontologie est un système de catégories permettant de classifier les éléments d'un univers. Le système de catégorisation correspond aux relations sémantiques. Ceci permet de regrouper de manière cohérente toutes les composantes d'un univers linguistique telles que les mots, les sens ou bien les concepts.

La relation sémantique servant de critère pour l'agrégation d'un groupe de concepts définira le type de l'ontologie. WordNet répertorie ainsi une grande variété de relations sémantiques permettant d'organiser le sens des mots (Et donc par extension les mots eux-mêmes) en des systèmes de catégories qu'on peut consulter de manière cohérente et uniforme. On pourra ainsi interroger le système quant aux hyperonymes d'un mot particulier. À partir par exemple du sens le plus commun du mot car (Correspondant au synset car, auto...) la relation d'hyperonymie définit un arbre de concepts de plus en plus généraux :

1. Car, auto, automobile, machine, motorcar.
    - Motor vehicle, automotive vehicle.
        - Vehicle.
            - Conveyance, transport.
                - Instrumentality, instrumentation.
                    - Artifact, artefact.
                        - Object, physical object.
                            - Entity, something.

Dans cet exemple, le dernier concept, « Entity, something », est le plus général, le plus abstrait. Il pourrait ainsi être le super-concept d'une multitude de concepts plus spécialisés.

On peut également interroger le système quant à la relation inverse de l'hyperonymie, l'hyponymie. WordNet offre en fait une multitude d'autres ontologies, faisant usage de relations sémantiques plus spécialisées et restrictives. On peut ainsi interroger le système quant aux méronymes d'un mot ou d'un concept, les parties constitutives d'un objet (HAS-PART). Les méronymes associés au sens car, auto... du mot car sont :

1. Car, auto, automobile, machine, motorcar.
    - HAS PART: Accelerator, accelerator pedal, gas pedal, gas, throttle, gun.
    - HAS PART: Air bag.
    - HAS PART: Auto accessory.
    - HAS PART: Automobile engine.
    - HAS PART: Automobile horn, car horn, motor horn, horn.
    - (...)

On peut aussi consulter le système quant à la relation inverse, l'holonymie, ou encore pour les relations de synonymie et d'antonymie.

C.4. Limites de WordNet.

- *Informations manquantes.*

WordNet ne précise pas l'étymologie, la prononciation, les formes de verbes irréguliers et ne contient que des informations limitées sur l'usage des mots [Chaumartin, 2007].

- *Profusion de sens pour un mot donné.*

La contrepartie de son importante couverture est que WordNet est très précis dans le sens des définitions. On a une granularité très (Trop ?) fine des sens. Par exemple, le verbe « to give » (« Donner ») n'a pas moins de 44 sens. Une telle profusion ne facilite pas une tâche de désambiguïsation lexicale [Chaumartin, 2007].

- *Absence de relations pragmatiques.*

WordNet ne matérialise pas d'une façon formelle tout le sens contenu dans les définitions des termes. Par exemple, l'information qu'un chat ne rugit pas figure dans la définition, mais ne se retrouve formalisée dans aucune relation. De même, des relations pragmatiques telles que savon / bain (SOAP#1 / BATH#2) sont absentes de WordNet [Chaumartin, 2007].

Qu'est ce que TreeTagger ?

D.1. Définition

En linguistique, l'étiquetage morpho-syntaxique (Aussi appelé étiquetage grammatical, POS tagging (Part-Of-Speech tagging) en anglais) est le processus qui consiste à associer aux mots d'un texte les informations grammaticales correspondantes comme la partie du discours, le genre, le nombre, etc. à l'aide d'un outil informatique [Valli et Véronis, 1999 ; Adda et al., 1999].

TreeTagger est un outil pour l'annotation du texte avec une partie du discours (POS) et de l'information lemme. Il a été développé par Helmut Schmid dans le projet de TC (TextCorpora) à l'Institute for Computational Linguistics de l'Université de Stuttgart (Allemagne). TreeTagger a été utilisée avec succès pour étiqueter allemand, anglais, français, italien, néerlandais, espagnol, bulgare, russe, portugais, galicien, chinois, swahili, slovaque, latin, estonien, polonais et les anciens textes français. Il est adapté à d'autres langues si le lexique et l'étiquetage manuel d'un corpus d'apprentissage sont disponibles [Helmut, 1994 ; Helmut, 1995].

Le mot	POS	Lemme
The	DT	The
TreeTagger	NP	TreeTagger
Is	VBZ	Be
Easy	JJ	Easy
To	TO	To
Use	VB	Use
.	SENT	.

Tab. 5. Exemple d'étiquetage d'une phrase à l'aide de TreeTagger.

D.2. Installation et paramétrage

La version Windows de TreeteTagger est disponible dans l'adresse suivante : « http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger-windows-3.2.zip », En décompressant le fichier zip et en suivant les instructions dans le fichier INSTALL.txt.

Les fichiers de paramètres doivent être téléchargés séparément. Les fichiers de paramètres Français et Italien sont fournis par Achim Stein (Institut für Linguistik/Romanistik Universität Stuttgart) sur le lien : « http ://www.uni-stuttgart.de/lingrom/stein/forschung/resource.html ».

D.3. Exemples d'étiquettes de TreeTagger

- **Pour la langue anglaise : The penn Treebank**

Tagset

CC Coordinating conjunction (and,but,or...)
CD Cardinal Number
DT Determiner
EX Existential *there*
FW Foreign Word
IN Preposition or subordinating conjunction
JJ Adjective
JJR Adjective, comparative
JJS Adjective, superlative
LS List Item Marker
MD Modal (can, could, might, may...)
NN Noun, singular or mass
NNP Proper Noun, singular
NNPS Proper Noun, plural
NNS Noun, plural
PDT Predeterminer (all, both ... when they precede an article)
POS Possessive Ending (Nouns ending in 's)
PRP Personal Pronoun (I, me, you, he...)
PRP$ Possessive Pronoun (my, your, mine, yours...)
RB Adverb (Most words that end in -ly as well as degree words like quite, too and very)
RBR Adverb, comparative (Adverbs with the comparative ending -er, with a strictly comparative meaning)
RBS Adverb, superlative
RP Particle
SYM Symbol (Should be used for mathematical, scientific or technical symbols)
TO *to*
UH Interjection (uh, well, yes, my...)
VB Verb, base form (subsumes imperatives, infinitives and subjunctives)
VBD Verb, past tense (includes the conditional form of the verb to be)
VBG Verb, gerund or persent participle
VBN Verb, past participle
VBP Verb, non-3rd person singular present
VBZ Verb, 3rd person singular present
WDT Wh-determiner (which, and *that* when it is used as a relative pronoun)
WP Wh-pronoun (what, who, whom...)
WP$ Possessive wh-pronoun **(**w, where why)
WRB Wh-adverb ho

Punctuation Tags

```
#
$
''
(
)
,
.
:
``
```

- **Pour la langue française**

```
ABR       Abreviation
ADJ       Adjectif
ADV       Adverbe
DET:ART   Article
DET:POS   Pronom Possessif (ma, ta, ...)
INT       Interjection
KON       Conjunction
NAM       Nom Propre
NOM       Nom
NUM       Numéral
PRO       Pronom
PRO:DEM   Pronom Démonstratif
PRO:IND   Pronom Indefini
PRO:PER   Pronom Personnel
PRO:POS   Pronom Possessif (mien, tien, ...)
PRO:REL   Pronom Relatif
PRP       Préposition
PRP:det   Préposition + Article (au,du,aux,des)
PUN       Ponctuation
PUN:cit   Ponctuation de citation
SENT      Balise de phrase
SYM       Symbole
VER:cond  Verbe au conditionnel
VER:futu  Verbe au futur
VER:impe  Verbe à l'impératif
VER:impf  Verbe à l'imparfait
VER:infi  Verbe à infinitif
VER:pper  Verbe au participe passé
VER:ppre  Verbe au participe présent
VER:pres  Verbe au présent
VER:simp  Verbe au passé simple
VER:subi  Verbe à l'imparfait du subjunctif
VER:subp  Verbe au présent du subjunctif
```

D.4. Utilisation

TreeTagger est un programme qui ne possède pas d'interface graphique. Il faut donc lancer le programme à l'aide de commandes dans l'invite de commandes DOS (Disk Operating System).

Une fois que l'invité de commandes est ouvert, on doit spécifier le chemin des programmes du TreeTagger pour que ceux-ci soient exécutables quelque soit le dossier dans lequel vous vous trouvez. Pour cela, on tape la commande suivante :

Set PATH=C:\TreeTagger\bin;%PATH%

Pour utiliser TreeTagger sur un fichier texte (Ecrit en langue française par exemple), on tape :

Perl tokenise-fr.pl *fichier-a-etiqueter.txt* **| bin\tree-tagger.exe lib\french.par -lemma -token -sgml >** *resultat-etiquetage.txt*

Dans cet exemple :

- Le fichier à étiqueter s'appelle : *fichier-a-etiqueter.txt*.
- Le programme tokenise-fr.pl « segmente » le texte à étiqueter et produit un flux de sortie (Avec un mot par ligne) qui est étiqueté par **TreeTagger**. Le résultat est stocké dans un fichier en sortie : *resultat-etiquetage.txt*.

TreeTagger utilise ici les 3 options :

-lemma : *Prints the lemma as well.*

-token : *Prints the token as well.*

-sgml : *Don't tag SGML annotations, i.e. lines starting with '<' and ending with '>'*

***Remarque** :*

On peut aussi ajouter l'option -no-unknown (Après -sgml) pour ne pas avoir en sortie l'affichage <unknown> si le lemme n'est pas connu.

- Le résultat est contenu dans : resultat-etiquetage.txt.

En sortie les zones textuelles hors balises seront constituées sur chaque ligne par : Un mot, une tabulation, sa catégorie, une tabulation et son lemme.

D.5. Options de TreeTagger

Les options du TreeTagger sont listées dans le fichier README-treetagger qui se trouve dans le dossier d'installation.

```
-token:          Prints the token as well.
-lemma:          Prints the lemma as well.
-sgml:           Don't tag SGML annotations, i.e. lines starting
           with '<' and ending with '>'.
-threshold <p>:Print all tags with a probability higher than
                 <p> times the probability of the best tag.
-prob:           Print tag probabilities (requires option -
           threshold).
-no-unknown:     Print the token rather than <unknown> for
           unknown lemmas.
-quiet:          Don't print status messages.
-pt-with-lemma:If this option is specified, then each
           pretagging tag (see above) has to be followed
      by a whitespace and a lemma.
-pt-with-prob:  If this option is specified, then each
           pretagging tag (see above) has to be followed
      by whitespace and a tag probability value. If -
pt-  with-rob  and -pt-with-lemma have been
specified, and then each     pretagging tag is
followed by a probability and a lemma in                    that
order.
-files f:        Read the names of input and output file
           pairwise from the file f. The format of f is
      the lexicon file format described below.
-lex f:          Read auxiliary lexicon entries from the file f.
-eos-tag <tag>:The SGML tag <tag> signals the end of a
           sentence. This option implies the option –sgml
```