

# Une Méthode Incrémentale d'Extraction de Connaissances Didactiques sur le Web (Version Longue)

Pierre Pompidor, Michel Sala, Danièle Hérin

► **To cite this version:**

Pierre Pompidor, Michel Sala, Danièle Hérin. Une Méthode Incrémentale d'Extraction de Connaissances Didactiques sur le Web (Version Longue). SETIT: Sciences Electroniques Technologies de l'Information et des Télécommunications, Mar 2003, Sousse, Tunisie. lirmm-00269621

**HAL Id: lirmm-00269621**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00269621>**

Submitted on 4 Sep 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Une méthode incrémentale d'extraction de connaissances didactiques sur le Web

Pierre Pompidor, Michel Sala, Danièle Hérin

LIRMM, 161 rue Ada  
34090 Montpellier  
[pompidor, sala, dh]@lirmm.fr

## 1. Introduction

Bien que les méthodes d'indexation de pages Web se soient notablement améliorées ces dernières années [GOOGLE 02], la pertinence des réponses fournies est loin d'être au niveau des attentes des internautes, et notamment des enseignants essayant d'y puiser matière pour leurs cours. Travaillant sur un projet de synthèse semi-automatique de connaissances extraites du Web, (et non préalablement annotées pour créer de nouveaux documents électroniques [LEVY 93]), nous nous sommes rapidement confrontés au double problème posé par l'imprécision, d'une part des requêtes que nous essayons de formuler, et d'autre part, par celle des réponses fournies par les moteurs de recherche interrogés.

Dans ce but, nous avons développé un prototype qui interroge automatiquement, (et non manuellement comme dans [BLONDEL et al 02]), un ou plusieurs moteurs de recherche en utilisant des listes de mots clefs de plus en plus élaborées. Ces mots clefs sont incrémentalement intégrés dans une ontologie [GRUBER 93] qui représente également l'ossature du cours en cours de réalisation. Ces mots clefs sont extraits incrémentalement des pages analysées, hormis les tous premiers qui doivent être manuellement insérés dans l'ontologie initiale. L'analyse effectuée pour extraire ces mots clefs est réalisée à partir d'une base de patrons syntaxiques extraits de l'analyse de milliers de définitions de différents dictionnaires en ligne, et ne concerne que des motifs de définitions ou de spécialisations. Un brouillon de cours final est généré lorsque plus aucun nouveau mot clef n'est intégré à l'ontologie.

## 2. Fonctionnement général de notre système

Le système débute par une phase préalable qui consiste :

- en l'**analyse des définitions d'un certain nombre de dictionnaires en ligne**
- et en la constitution d'une **ontologie minimale du domaine**

Se poursuit par un certain nombre de cycles qui enchaînent trois phases :

- une **phase de requête** où un requêteur interroge un ou plusieurs moteurs de recherche à partir des concepts de l'ontologie

- une **phase d'analyse textuelle** automatique des pages renvoyées, à la recherche de nouvelles connaissances pédagogiques (nouveaux concepts ou explications).
- et l'**enrichissement de l'ontologie** par ces nouveaux concepts,  
Et se termine quand plus aucun concept n'est intégré dans l'ontologie par :
- la production d'un brouillon de cours
- l'application de procédures de synthétisation pour éviter les redondances
- l'application d'une procédure normative pour rendre cette ressource réutilisable

Nous prendrons comme exemple la problématique suivante : l'enseignant veut créer un cours sur les «architectures multi-tiers» dont il ne connaît que quelques notions. Il crée une ontologie affiliant les concepts «**serveur d'application**» et «**serveur d'objets**» au concept «**architectures multi-tiers**».

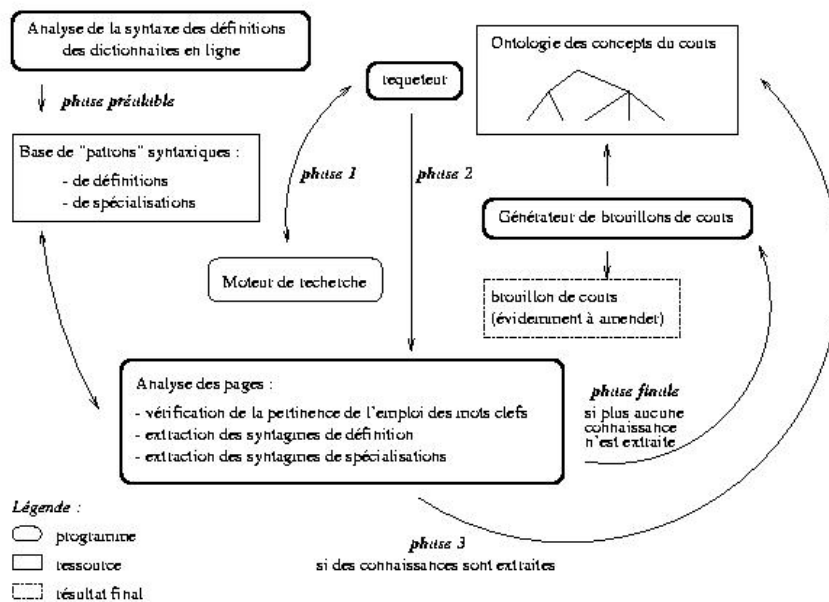


Figure 1. Fonctionnement général du système

Le requêteur invoque Google sur une liste de mots clefs pris sur chaque branche de l'ontologie, soit dans notre exemple, une des combinaisons suivante :

- «architecture multi-tiers» «serveur d'application»
- «architecture multi-tiers» «serveur d'objets»

(les mots clefs composites devant être d'un seul tenant), puis il importe les pages référencées et les transmet à l'analyseur qui mène une analyse syntaxique développée pour un système d'identification de services sur le Web (« Chimère » [SEGRET et al 00]. Pour chaque concept de l'ontologie, l'analyse extrait :

- les définitions qui donnent une explication de ce concept
- les spécialisations qui listent des instances de ce concept

Cette analyse s'appuie sur des patrons appris de l'analyse de milliers de définitions de plusieurs dictionnaires en ligne (calculs de fréquences sur les sous-arbres des arbres syntaxiques générés par l'analyse syntaxique de ces dictionnaires).

Extraction de définitions et de spécialisations à partir de deux ressources :

Un **serveur d'application** est basé sur une **architecture multi-tiers**. C'est un modèle d'architecture d'applications dans lequel *on sépare la présentation, les traitements et les données*. L'objectif poursuivi est de permettre ... .

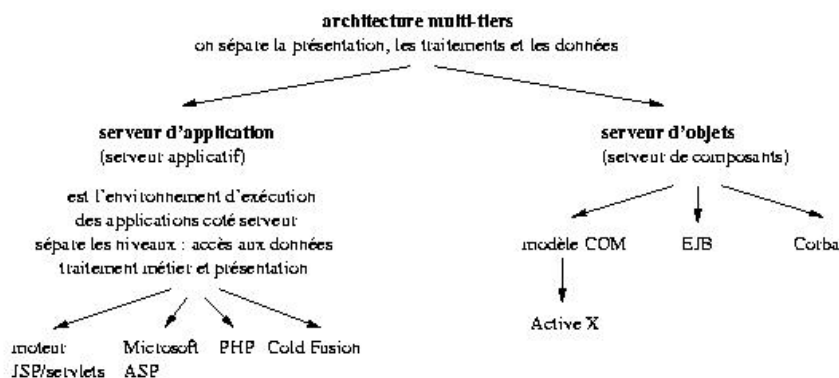
De : <http://users.skynet.be/johant/servapp.htm>

- les mots clefs **serveur d'application** et **architecture multi-tiers** s'apparient
- le patron *c'[être] un(e) généralisation-du-Concept-X dans lequel* s'apparie  
-> extraction de « *on sépare la présentation, les traitements et les données* »

Le **serveur d'application** est *l'environnement d'exécution des applications côté serveur*. Il prend en charge ... . Les moteurs JSP/Servlets, Microsoft ASP, Cold Fusion, PHP ... sont à ce titre des **serveurs d'application** ... .

De : <http://medias.obs-mip.fr/cours/sqbd/cours013.html>

- le mot clef **serveur d'application** s'apparie (trois fois) -> la page est analysée
  - Le référent **II** est déterminé (il est valué par **serveur d'application**)
  - la dernière phrase sera analysée par un patron de spécialisation
- A partir de ces extractions et d'autres induites par les patrons de spécialisation,



l'ontologie est enrichie :

**Figure 2.** *L'ontologie après l'intégration des définitions et des spécialisations*

A partir de cette ontologie enrichie, le requêteur transmet de nouvelles requêtes au moteur de recherche en intégrant des mots clef qui devront être satisfaits, ce qui permet d'arrêter l'enrichissement. Notre méthode convergeant, finalement l'ontologie n'évolue plus et le générateur de brouillon de cours produit un texte synthétique dont les concepts de l'ontologie sont l'ossature :

Dans une **architecture multi-tiers**, on sépare la présentation, les traitements et les données. Il se compose de serveur d'application et de serveur d'objets. Un **serveur d'application** est l'environnement d'exécution des applications côté serveur. Il se compose en moteur JSP/Servlets, Microsoft ASP, PHP et Cold Fusion ... .

Il est évident que ce brouillon de cours est fortement à amender, la construction des

phrases étant très monotone ou approximative.

#### 4. Conclusion

Notre méthode, et le prototype associé, commencent à produire des résultats concrets. En effet, l'enseignant à partir de quelques concepts peut rapidement intégrer des connaissances didactiques dans un brouillon de cours, et ceci sur un sujet dont il ne connaît que les fondements et la logique globale. L'hypothèse que nous retenons est de construire un cours par rapport à des ressources non certifiées, sinon par leur apparition en tête de listes de moteurs de recherche reconnus (comme Google ou Teoma). En effet, ces documents ne sont ni référencés, ni préalablement annotés sémantiquement. Idéalement, nous devrions pouvoir comparer le brouillon de cours produit, à un cours électronique de référence sur le domaine.

Par ailleurs, nous travaillons à amoindrir les contraintes suivantes :

- les techniques de synthétisations de connaissances que nous appliquons sont pour l'instant trop sommaires pour éviter toute duplication de définitions ou d'annotations. De plus, nous n'y avons pas encore intégré de mécanismes de révisions permettant de détecter et de lever des définitions contradictoires.
- Enfin, le cycle d'apprentissage doit être continué pour faire de la ressource d'apprentissage (l'ontologie), une ressource réutilisable et normée.

#### 5. Bibliographie

[BLONDEL et al 02] Blondel Fr.M., Le Touzé J.C., Tarizzo M., « *ARI : un assistant logiciel pour accompagner la formation à la recherche d'informations* », *TICE 2002*, Lyon 2002,

[GOOGLE 02] <http://www.google.com/technology/>

[GRUBER 93] Gruber T., « *A Translation Approach to Portable Ontology Specifications* », *Knowledge Acquisition* 5, 1993, p. 19-220

[LEVY 93] Levy D. M., « *Document reuse and document systems* », *Electronic publishing*, vol. 6(4), 1993, p. 339-348

[SEGRET et al 00] Segret M.-S., Pompidor P., Hérin D., « *Extraction et intégration d'informations semi-structurées dans les pages Web – Projet Chimère* », *Journées francophones d'Ingénierie des Connaissances*, 2000