

# Construction d'une ontologie pour la recommandation de films à un utilisateur

Latifa Baba-Hamed, Réda Soltani et Kamel Sabri

Université d'Oran, Département d'informatique  
lbabahamed@yahoo.fr

**Résumé :** Avec l'avènement d'internet, nous assistons aujourd'hui à une surcharge d'informations. Les systèmes de recommandation sont des outils efficaces conçus pour pallier au problème de surcharge informationnelle, en fournissant aux utilisateurs les informations les plus pertinentes. Nous avons développé une application qui recommande des films à des utilisateurs en se basant sur la similarité sémantique qui existe entre la description d'un film et le profil d'un utilisateur. A cet effet, nous avons construit une ontologie dans laquelle figure tous les concepts présents dans le profil de l'utilisateur et dans le descripteur du film. Le challenge étant de mesurer la similarité entre deux concepts en se basant sur cette ontologie. Cet article présente, essentiellement, la démarche suivie et les outils utilisés pour le développement de notre ontologie.

**Mots-clés :** Système de recommandation, Similarité sémantique, Ontologie de films, Concept, Classe, Propriété, Instances, Editeur d'ontologie.

## 1 Introduction

Une ontologie (Gruber 1993) définit le vocabulaire partagé pour aboutir à une compréhension commune d'un domaine donné. Dans ce papier, nous construisons une ontologie pour filtrer des requêtes utilisateur dans le domaine du cinéma. Un système de filtrage (ou système de recommandation) (Middelton et al., 2004) collecte, sélectionne, classe et suggère à l'utilisateur les informations qui répondent vraisemblablement à ses intérêts à long terme. Nous pouvons distinguer deux grandes approches de filtrage: basé sur le contenu et collaboratif. Le filtrage basé sur le contenu (Shoval et al., 2007) compare le descriptif des nouveaux produits au profil de l'utilisateur, et recommande ceux qui sont les plus proches. Cette classe de systèmes utilise des mesures de similarité (D'Amato et al., 2008), (Schickel-Zuber & Faltings, 2007), pour définir le degré de ressemblance entre concepts. Cet article présente les différentes étapes suivies pour la construction de notre ontologie de films. Il est organisé comme suit. La section 2 décrit brièvement, l'architecture générale du système afin de comprendre le rôle et l'utilité d'une ontologie pour notre application. La section 3 expose la démarche suivie pour le développement de l'ontologie des films. Enfin, la section 4 présente une conclusion et quelques perspectives.

## 2 Architecture générale du système

Notre système se compose essentiellement de six modules (voir figure 1).

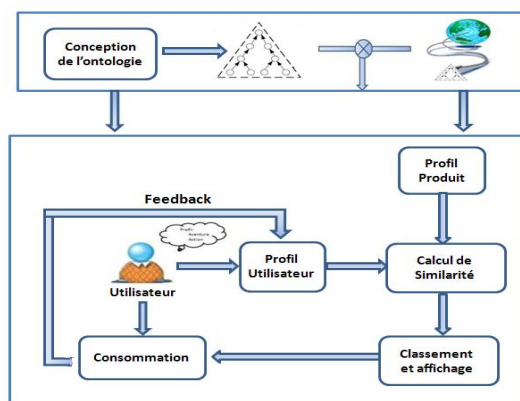


Fig. 1 –L'architecture générale du système

- Le module *Conception de l'Ontologie* a comme tâche, la construction d'une ontologie de produits (films dans notre cas).
- Le module *Création de Profil Utilisateur* a pour tâche de créer un profil utilisateur. Pour simplifier, nous avons supposé, dans notre système, que le profil de l'utilisateur se résume à un ensemble de prédicats pondérés. Un prédicat est de la forme <attribut opérateur de comparaison valeur>, où attribut est un élément du profil, et valeur est une valeur appartenant au domaine de l'attribut du profil. P1 : <Film.genre, '=', action> est un exemple de prédicat. Un utilisateur U1 qui a une grande préférence pour les films d'action peut attribuer à ce prédicat une pondération (poids)  $w_1=0,9$  par exemple. Le profil de l'utilisateur U1 sera donc constitué des paires d'éléments : (P1,  $w_1$ ), (P2,  $w_2$ ), etc.
- Le module *Création de Profil Produit* a pour tâche de créer le descriptif du produit (film dans notre cas). Cela correspond, dans notre application, aux genres associés aux films. Un film peut avoir plusieurs genres.
- Le module *Calcul de Similarité* a pour objectif le calcul de la similarité entre les genres associés à un film et les genres de films préférés par un utilisateur (Sabri & Soltani, 2009). Par exemple, supposons que nous disposons de deux films : film1(aventure) et film2(drame), et qu'il n'y a qu'un seul prédicat <Film.genre, '=', action> dans le profil de l'utilisateur U1. En utilisant une ontologie de films, et une mesure de similarité sémantique, il faut calculer la similarité entre le profil de U1 et les deux films, pour décider lequel des deux films lui recommander. Cela revient (pour cet exemple) à mesurer les similarités :  $sim(Action, Aventure)$  et  $sim(Action, Drame)$ . Dans ce cas, film1 sera recommandé à U1 car le

concept Aventure est plus similaire au concept Action que le concept Drame.

- Le module *Classement et affichage des produits* est chargé de classer et d'afficher les produits (films) par ordre décroissant selon la similarité

### **3 Construction de l'ontologie**

Pour le développement de notre ontologie, nous avons essayé de suivre les étapes proposées dans (Noy & McGuinness, 2004). Notre application nécessite une ontologie de films, qui doit nous permettre de calculer le degré de ressemblance entre des genres de films (les genres préférés par un utilisateur et les genres associés à un film) dans le but de recommander une liste de films à un utilisateur. Par conséquent nous pouvons limiter le domaine de notre ontologie aux genres de films, aux propriétés des films et leurs valeurs.

OWL (w3c : <http://www.w3.org>) est le langage connu pour la construction d'une ontologie. En ce qui concerne notre ontologie, nous avons utilisé l'éditeur « Protégé », distribué en open source par l'université en informatique médicale de Stanford et disponible sur le site : <http://protege.stanford.edu/>. Il permet, à travers son interface graphique la génération automatique du code OWL correspondant à l'ontologie.

Nous présentons dans ce qui suit quelques étapes pour la construction de notre ontologie.

#### **3.1 Définition des classes et de la hiérarchie des classes**

Nous avons, dans un premier temps, recensé tous les genres de films en s'inspirant des données des sources IMDb se trouvant sur le site : <http://www.imdb.com/> , et MovieLens se trouvant sur le site : <http://movielens.umn.edu/>. Nous avons ensuite, classé les genres de films en classes et sous-classes formant ainsi une hiérarchie de classes ayant pour racine la classe Film. Ces classes constituent les concepts de notre ontologie. Nous avons retenu 18 genres de films : Aventure, Action, Dessin-Animés, Enfant, Manga, Comédie, Documentaire, Drame, Fantastique, Film-Noir, Horreur, Musical, Mystère, Romance, Science-Fiction, Thriller, Guerre, Western. Pour établir la hiérarchie des classes, nous avons procédé de haut en bas en commençant par les concepts les plus généraux et en terminant par la spécialisation des concepts. Nous avons, donc, commencé par les classes les plus générales, à savoir, Aventure, Thriller, Drame, Enfant, Documentaire. Ensuite, nous avons affiné chacune de ces classes. Par exemple, la classe Thriller a été affinée en les concepts : Film-Noir, Horreur, et Mystère, ainsi que la classe Aventure qui a été spécialisée en les sous-concepts : Action, Fantastique, Guerre, Science-fiction, Western, et ainsi de suite pour les autres concepts. owl:Thing est une classe prédéfinie. Toute classe OWL est une sous-classe d'owl:Thing. La figure 2 est

une représentation graphique de la hiérarchie des classes de notre ontologie, à l'aide de l'outil Jambalaya sous Protégé.

### 3.2 Définition des propriétés des classes

Les classes seules ne fourniront pas assez d'information. Nous devons donc décrire la structure interne ou propriétés des concepts. Il existe deux types de propriétés : les propriétés d'objet, qui permettent de relier des instances à d'autres instances, et les propriétés de type de donnée qui permettent de relier des individus à des valeurs de données. Une propriété d'objet est une instance de la classe owl:ObjectProperty, une propriété de type de données est une instance de la classe owl:DatatypeProperty. Ces deux classes sont elles-mêmes sous-classes de la classe Property. Nous n'avons utilisé, dans notre cas, que des propriétés de type (pas de propriétés d'objet). En effet, les seules propriétés retenues pour les films, sont: Acteurs, Année-de-Production, Durée, Pays, Réalisateur, Scénaristes. Une propriété typée peut faire correspondre un type aux instances de la classe. Par exemple, la propriété Année-de-Production (resp. Acteurs) fait correspondre aux instances de la classe Film le type entier positif (resp. le type chaîne de caractères) représentant l'année de production (resp. la liste des acteurs) du film. Nous pouvons également exprimer des restrictions sur les propriétés. La restriction peut s'exprimer sur la cardinalité d'une propriété comme, par exemple, le fait qu'un film doit avoir une seule année de production (exprimée avec Protégé comme : Année-de-Production exactly 1), ou bien, le fait qu'un film doit avoir deux réalisateurs au maximum (exprimée par : Réalisateur max 2).

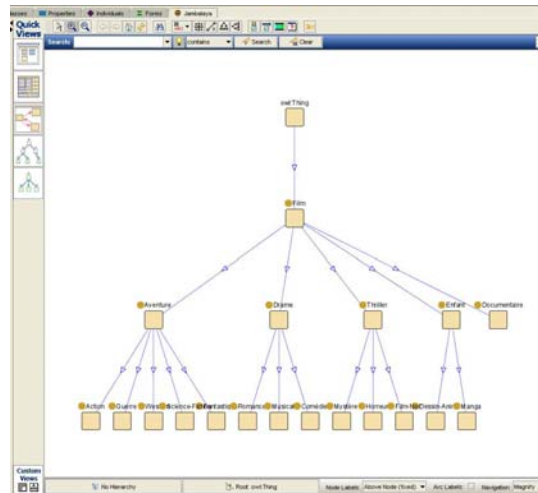


Fig. 2 –Représentation hiérarchique de l'ontologie des films

## 4 Conclusion

Dans l'objectif de développer un petit système de recommandation, nous avons été amenés à construire une ontologie de films dont certaines étapes sont présentées dans ce papier. La méthode de construction de l'ontologie est manuelle. La difficulté de l'accès à l'ontologie de films existante nous a empêchés de la réutiliser, ce qui nous a amenés à construire notre propre ontologie. Nous avons la possibilité d'utiliser cette dernière telle quelle, ou bien de l'héberger dans un site et d'en faire appel à partir de l'URI correspondant. Nous avons développé notre application sur une machine Intel Pentium D, sous Windows XP. Le langage utilisé est JBuilder2005 ainsi que l'éditeur d'ontologie Protégé 3.4 beta. Jena qui est un parseur, disponible à l'adresse : <http://jena.sourceforge.net/>, nous a été d'une grande utilité pour l'exploitation de l'ontologie ainsi créée. Notre système est facilement adaptable à d'autres ontologies de produits différents de films, comme par exemple les livres. C'est ce que nous essaierons de montrer dans un travail futur.

## Références

- Middelton, S. E.; Shadbolt, N. R.; and De Roure, D. C. (2004). Ontological User Profiling in Recommender Systems. *ACM Transactions of Information Systems* 22(1), P.54 – 88.
- Shoval, P. Maidel V., and Shapira B. (2007). An Ontology- Content-based Filtering Method. *I.Tech-2007, Information Research and Applications*.
- D'AMATO C., STAAB S. & FANIZZI N. (2008). On the influence of description logics ontologies on conceptual similarity. In *EKAW 2008, International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns*, p. 48–63.
- Schickel-Zuber V. & Faltings B. (2007). OSS: A Semantic Similarity Function based on Hierarchical Ontologies. In *IJCAI'07*, P. 551-556.
- Gruber T.R. (1993). Towards Principles for the Design of Ontologies used for Knowledge Sharing. In N. Guarino (Ed.). *Int. Workshop on Formal Ontology*, Padova, Italy.
- Sabri K. & Soltani R. (2009). Matching entre un profil utilisateur et un produit. Mémoire d'ingénieur, université, d'Oran.
- Noy N. F. & McGuinness D. L. (2005). Développement d'une ontologie 101 : Guide pour la création de votre première ontologie, Université de Stanford. Rapport technique.