

WikiTaaable, un wiki sémantique utilisé comme un tableau noir dans un système de raisonnement à partir de cas textuel

Amélie Cordier, Jean Lieber, Pascal Molli, Emmanuel Nauer, Hala Skaf-Molli, Yannick Toussaint

► To cite this version:

Amélie Cordier, Jean Lieber, Pascal Molli, Emmanuel Nauer, Hala Skaf-Molli, et al.. WikiTaaable, un wiki sémantique utilisé comme un tableau noir dans un système de raisonnement à partir de cas textuel. Béatrice Fuchs and Amedeo Napoli. 17ème atelier de Raisonnement à Partir de Cas - RàPC 2009, Jun 2009, Paris, France. 2009. <inria-00437399>

HAL Id: inria-00437399

<https://hal.inria.fr/inria-00437399>

Submitted on 30 Nov 2009

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.

WIKITAAABLE, un wiki sémantique utilisé comme un tableau noir dans un système de raisonnement à partir de cas textuel

Amélie Cordier¹, Jean Lieber², Pascal Molli²,
Emmanuel Nauer², Hala Skaf-Molli², Yannick Toussaint²

¹Université de Lyon, CNRS,
Université Lyon 1, LIRIS, UMR5205, F-69622, France.
{firstname.lastname}@liris.cnrs.fr

²INRIA Nancy-Grand Est
Nancy Université
LORIA – UMR 7503. B.P. 239, F-54506 Vandœuvre-lès-Nancy cedex, France.
{firstname.lastname}@loria.fr

Résumé

Les wikis sémantiques permettent à des communautés d'utilisateurs de produire des connaissances formalisées compréhensibles et utilisables par les machines. Afin d'aller plus loin, on pourrait imaginer d'utiliser les wikis sémantiques comme des tableaux noirs permettant aux humains et aux machines d'interagir afin de construire des connaissances qui soient à la fois utiles aux humains et aux machines. Dans ce papier, nous présentons une étude de cas portant sur l'utilisation d'un wiki sémantique (connu sous le nom de Semantic Media Wiki) comme un tableau noir pour gérer des données et des connaissances culinaires. Cette étude de cas est réalisée dans le contexte de TAAABLE, le fameux système de raisonnement à partir de cas en ligne capable de résoudre des problèmes culinaires en utilisant une base de recettes existantes. Avec WIKITAAABLE, l'évolution de TAAABLE intégrant un wiki sémantique, nous voulons montrer comment un wiki sémantique peut apporter support et assistance aux utilisateurs chargés des tâches fastidieuses de gestion des connaissances. Nous montrons en particulier comment le wiki facilite la prise en compte du *feedback* des utilisateurs. Les questions relatives à la présence d'utilisateurs multiples ainsi qu'à l'intégration de plusieurs mécanismes de gestion des connaissances dans une application unique sont discutées à la fin de ce papier.

Mot-clés : Acquisition collaborative de connaissances, wikis sémantiques, annotation et indexation textuelle, ontologies, raisonnement à partir de cas (RÀPC), coopération homme-machine, enrichissement d'ontologies.

1 Introduction

Par le passé, les wikis ont démontré leurs capacités à transformer une communauté d'étrangers en communauté de collaborateurs. En intégrant des technologies issues du Web Sémantique, les wikis sémantiques [8, 7, 2] permettent désormais à ces communautés de contributeurs de produire des connaissances formalisées exploitables par les machines. Il est fort probable que la prochaine étape soit d'utiliser les wikis sémantiques comme des tableaux noirs permettant aux humains et aux machines d'interagir plus facilement pour produire des connaissances partagées, compréhensibles et utilisables par tous, humains et machines.

Forts de cette observation, dans ce papier nous étudions la question en la développant dans le cadre du projet TAAABLE [1]. TAAABLE¹ permet à ses utilisateurs d'envoyer des requêtes culinaires

¹TAAABLE est accessible à l'adresse <http://taaable.fr/>

auxquelles le système répondra en s'appuyant sur une base de recettes de cuisine. Par exemple, on peut demander à TAAABLE une recette de dessert à la rhubarbe mais qui ne contienne pas de chocolat. Si aucune recette de la base de recette ne permet de répondre à cette requête, TAAABLE en choisit une proche (par exemple, la recette de la tarte rhubarbe-chocolat) et l'adapte afin de répondre à la requête posée (par exemple, en retirant le chocolat, ou encore en le remplaçant par un autre ingrédient, comme les myrtilles ou les framboises qui se marient à merveille avec la rhubarbe et apportent l'équilibre en sucre nécessaire²). Comme tout lecteur attentif s'en doute, le moteur de raisonnement de TAAABLE s'appuie sur les principes du raisonnement à partir de cas (RÀPC [6]) et pratique donc couramment l'adaptation. Le moteur de TAAABLE utilise différentes sources de données et de connaissances : un ensemble de recettes annotées (la base de recettes) ; des ontologies pour les ingrédients, types de plats, origines géographiques des plats, types de régimes alimentaires (végétarien, sans oléagineux, sans alcool) ; et enfin, un ensemble de règles d'adaptation. Les recettes annotées sont élaborées à partir des recettes initiales, (souvent mal) écrites en langue naturelle : un outil d'annotation utilise les différentes ontologies pour produire ces recettes annotées.

Lorsque le système fonctionne bien (autant dire jamais), la maintenance de la base de recettes et des connaissances périphériques est triviale. En revanche, s'il est nécessaire de faire des corrections ou des mises à jour un peu compliquées, les problèmes commencent à apparaître. Par exemple, si l'on souhaite ajouter une recette à la base, il faut que cette recette puisse être indexée comme les autres sans quoi elle ne sera jamais utilisée. Par ailleurs, si l'on détecte une erreur dans les résultats produits par le système, il serait souhaitable de pouvoir la corriger immédiatement plutôt que de la noter sur un petit cahier en attendant de la corriger dans la version de l'année suivante. Ces différents points sont des limites évidentes de la première version de TAAABLE : le processus d'indexation des recettes en langage naturel est loin d'être infallible, le système n'offre aucune possibilité d'interactions en cas d'erreur et, pour couronner le tout, même si toutes les requêtes sont tracées, aucune prise en compte du *feedback* des utilisateurs n'est effectuée. En résumé, à l'exception des corrections apportées par les concepteurs, qui elles-mêmes ne sont pas aisées, rien ne permet au système TAAABLE d'améliorer ses capacités d'adaptation internes.

Dans ce papier, nous décrivons l'utilisation d'un wiki sémantique dans TAAABLE et nous étudions sa capacité à nous aider à gérer les données (les recettes et les informations terminologiques du domaine culinaire) et les connaissances (les ontologies et les règles d'adaptation) utilisées par le moteur de RÀPC. La vocation finale du wiki sémantique est d'aider les utilisateurs à produire de nouvelles connaissances : ajout de nouvelles recettes, correction des problèmes d'indexation existants, prise en compte du *feedback* sur les résultats de l'adaptation. Le wiki rassemble également les processus permettant d'effectuer des adaptations, d'indexer de nouvelles recettes et de prendre en compte le *feedback* des utilisateurs pour améliorer les capacités internes d'adaptation et d'indexation. Même si l'objectif final est de permettre à tous les membres d'une communauté de collaborer et de contribuer à la construction des connaissances de TAAABLE, le wiki est pour l'instant utilisé uniquement par les concepteurs de l'application et apporte déjà une aide significative.

Ce papier est découpé en deux sections principales, l'une portant sur la « vieille » application TAAABLE (section 2) et l'autre sur la nouvelle application WIKITAAABLE (section 3), l'idée étant de montrer en quoi la nouvelle version apporte des solutions aux problèmes soulevés par l'ancienne. L'article se termine par une conclusion ainsi qu'une rapide discussion sur les travaux en cours et à venir.

2 Le projet TAAABLE

Le projet TAAABLE a participé en 2008 au Computer Cooking Contest³ (CCC), une compétition scientifique internationale visant à faire s'affronter des systèmes informatiques capables de résoudre des problèmes de cuisine. Selon les règles du concours, les systèmes candidats devaient être à même de

²Recette disponible sur demande. Nous vous la recommandons, c'est fameux

³<http://www.wi2.uni-trier.de/eccbr08/index.php?task=ccc>

répondre à des requêtes (du type de celle présentée en introduction relative à la tarte à la rhubarbe) en retrouvant et/ou en adaptant des recettes existantes afin qu'elles satisfassent les contraintes imposées par les utilisateurs. Le livre de recettes, identique pour tous les participants, était constitué d'un ensemble de recettes textuelles décrites simplement par un titre, une liste faiblement structurée d'ingrédients et un ensemble d'instructions de préparation. Les requêtes des utilisateurs, quant à elles, peuvent porter non seulement sur les ingrédients à inclure ou à éviter dans la recette mais peuvent aussi poser des contraintes sur les types de plats, leurs origines ou encore les régimes alimentaires à prendre en compte. Par exemple : « je veux une salade méditerranéenne végétarienne, sans produits à base de noix, contenant des tomates mais pas d'ail et encore moins de concombres » est une requête qui aurait pu figurer parmi les exercices imposés aux participants du CCC en 2008.

Pour résoudre de tels problèmes, l'assemblage de différentes compétences est souhaitable. C'est pourquoi le projet TAAABLE implique des chercheurs qui s'intéressent à des champs de recherches diversifiés tels que le raisonnement à partir de cas (RÀPC), l'ingénierie des ontologies, la fouille de données et de textes, l'indexation de ressources textuelles ou encore la classification hiérarchique. Grâce à la fusion de ces différentes compétences, l'équipe est parvenue à réaliser un système fonctionnel qui a obtenu le titre de vice-champion du CCC en 2008. Pour 2009, les règles du nouveau CCC n'ont que très sensiblement été modifiées, nous avons donc décidé de réutiliser notre expérience pour cette nouvelle édition. Par conséquent, nos objectifs et nos recherches en cours visent désormais à améliorer le système existant en développant de nouvelles possibilités d'évolution de l'application.

Passons aux choses sérieuses. Le reste de cette section présente une vision générale de l'architecture de TAAABLE (ancienne version) et décrit ses principales composantes en montrant les limites.

2.1 L'application TAAABLE : une vision générale

L'architecture du système TAAABLE est composée de deux parties distinctes. La partie dite *offline* couvre les différents aspects de la gestion des connaissances et l'annotation des recettes du CCC. Des outils spécifiques ont été développés pour les besoins de cette partie. Cette partie se présente donc comme une chaîne de traitement à l'issue de laquelle est obtenu un ensemble de données et de connaissances utilisées par le moteur de raisonnement à partir de cas de l'application.

Ce moteur, associé à l'interface web de requête et de présentation des résultats constitue la partie dite *online* de l'application. La figure 1 reprend ces différents éléments.

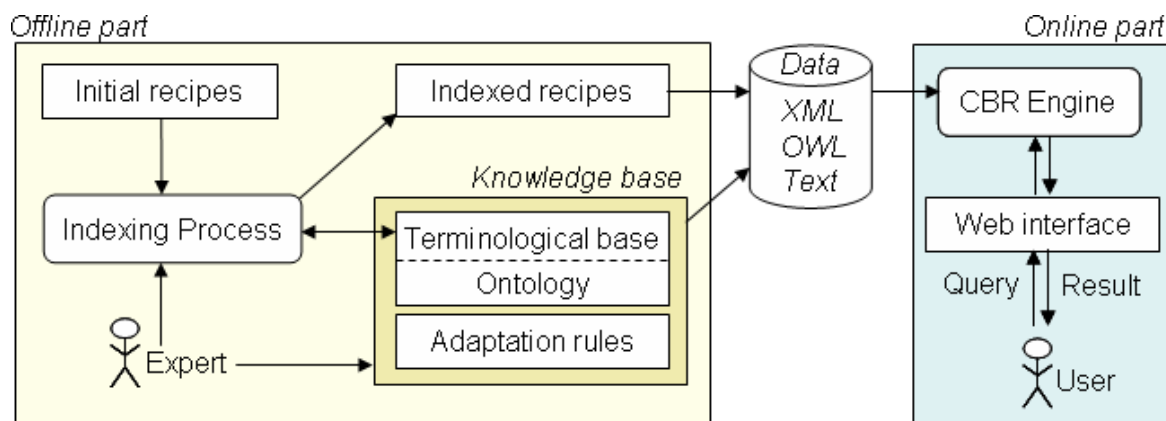


FIG. 1 – Architecture du système TAAABLE.

L'interface web de l'application permet aux utilisateurs de saisir leurs requêtes. Le système répond à une requête en retournant une liste de recettes jugées satisfaisantes. Si des adaptations de ces recettes sont nécessaires, elles sont signifiées à l'utilisateur qui peut alors en consulter le détail. La figure 2 montre un exemple de réponse à la requête « je vous saurai gré de bien vouloir me proposer une recette de tarte à l'orange ».



Ingredients

I want: ? I don't want: ?

Type of dish

I want: ? I don't want: ?

More options

Vegetarian Nut-free No alcohol [Advanced Configuration ?](#)

Your request: orange D:pie

Common path: 1<citrus_fruit--> orange>

Common cost: 1.3778748590755354

#	Original recipe name	Adaptation overview	Cost
1	Apple Crumble Pie	Replace: lemon_juice by citrus_fruit	5
2	Delicious Key Lime Pie	Replace: key_lime_juice by citrus_fruit, key_lime_peel by citrus_fruit	5
3	Key Lime Pie	Replace: key_lime by citrus_fruit, key_lime_juice by citrus_fruit	5
4	Strawberry Lime Pie	Replace: lime by citrus_fruit	5
5	UPSIDE DOWN APPLE PIE	Replace: lemon_juice by citrus_fruit	5

Results 1 - 5 on 5 | Processing time: 0.0246 secondes



[Help me!](#) | [More about Taaable](#) | [The team](#) | [Administration](#)

FIG. 2 – L'interface web de l'application de RÀPC TAAABLE

Pour TAAABLE, adapter une recette consiste à remplacer certains ingrédients par d'autres. En cliquant sur le titre d'une recette, l'utilisateur peut en voir le contenu détaillé, la liste des ingrédients initiaux et surtout les ingrédients à substituer par d'autres. Sur la figure 3, la recette intitulée *Key Lime Pie* est adaptée en remplaçant le *key lime juice* et/ou le *key lime* par de l'orange (vision simpliste mais efficace de l'adaptation).

2.2 Organisation des connaissances

Hiérarchie des ingrédients. Le moteur de raisonnement à partir de cas adapte les recettes en substituant des ingrédients par d'autres. Pour l'instant, pour des raisons de simplicités, chaque ingrédient est remplacé par au plus un autre ingrédient. L'idée sous-jacente à cette démarche est qu'en général, une recette peut être adaptée en remplaçant un ingrédient par un autre ingrédient en étant « proche ». Toute la question est donc de savoir ce qu'est un ingrédient « proche ». Pour cela, TAAABLE utilise une ontologie qui organise les ingrédients en fonction des besoins de l'application (et non en fonction des propriétés botaniques des légumes par exemple...). Le moteur de raisonnement à partir de cas est capable de calculer des « coûts de substitution » en fonction de cette ontologie : plus les ingrédients sont proches dans l'ontologie et plus le coût de substitution de l'un par l'autre est faible. Par exemple, le coût de remplacement d'un orange par un citron est plus faible que le coût de remplacement d'une orange par un steak haché car, selon l'ontologie utilisée, l'orange est plus proche

Key Lime Pie

Type(s): D:baked good,D:dessert,D:cheesecake,D:pie,D:sweet

Substitution List

- ◆ Replace **key_lime_juice**, **key_lime**, by **orange**

Original Recipe

Ingredients :

Original ingredient	Preferred term
3/4 cup key lime juice	key lime juice
2 teaspoons key lime juice	key lime juice
2 1/4 cups sweetened condensed milk	sweetened condensed milk
1 teaspoon grated key lime rind	key lime
3 egg yolks	egg yolk
9 inch graham cracker pie crust	graham cracker
sweetened whipped cream	whipped cream

Recipe :

yield: 8 servings squeeze juice from 4 large or 6 small key lime and grate rind set aside using a whisk beat egg yolk until buttercup yellow add about half the condensed milk blend well with a whisk and add remaining milk add half the lime juice and blend slowly add remaining juice and blend add grated rind mix and pour into chilled pie crust refrigerate for 4 hours may be frozen slice while still frozen and let stand for about 10 minutes top with whipped cream

[Go back](#)

FIG. 3 – Un exemple d’adaptation d’une recette.

du citron que du steak haché.

Hiérarchies des types de recettes. Le typage des recettes est nécessaire afin de permettre de répondre aux requêtes du CCC. En effet, les origines des recettes, par exemple, ne sont pas précisées dans les recettes initiales et doivent être ajoutées à la main ou bien inférées d’une façon ou d’une autre. Afin de typer les recettes, trois hiérarchies complémentaires ont été définies. La hiérarchie des *dish moments* (les moments du plat) tels que les gâteaux apéritifs ou encore les desserts, la hiérarchie des *dish types* (les types de plats) comme les gâteaux ou les pizzas, et la hiérarchie des *dish origins* (les origines des plats) qui contient des concepts tels que « Méditerranéen » ou « Chinois ».

Dans la suite, nous parlons des quatre hiérarchies que nous venons de décrire comme de *l’ontologie* du système. L’ontologie de TAAABLE décrit donc les concepts du domaine utilisés dans l’application. Le lien entre la structure conceptuelle (l’ontologie) et le niveau linguistique (les recettes originales) est établi grâce à la base terminologique : chaque concept de l’ontologie est associé à sa forme linguistique qui se présente comme un ensemble de variantes terminologiques. Par exemple, le concept *litchi* est associé aux termes *litchi*, *lichi*, *lychee*, *leechee*, et *laichee*.

L’ontologie et la base terminologique ont été construites conjointement par les experts. En fonction des résultats du processus d’indexation (cf. ci-dessous pour plus de détails) qui peut mettre en évidence des problèmes dans la terminologie, les experts pouvaient décider « au cas par cas » quels termes et quels concepts (éventuellement associés à des termes) devaient être ajoutés dans la hiérarchie.

En complément de l'ontologie, le système de RÀPC peut être amené à utiliser une base de règles d'adaptation pour adapter les recettes. Une règle d'adaptation est définie comme une paire ordonnée d'ensemble d'ingrédients (s_1, s_2) associée à un coût établi par l'expert. Formellement, (s_1, s_2) signifie que l'ensemble d'ingrédients s_1 peut être remplacé par l'ensemble d'ingrédients s_2 .

Les règles d'adaptation et l'ontologie constituent la **base de connaissances** du système.

2.3 Annoter les recettes en fonction de la base terminologique

Le **processus d'annotation** est automatique et s'appuie sur les données disponibles dans la base terminologique. En entrée, il prend les recettes dans leur forme textuelle initiale. Pour chaque recette, la sortie du processus contient : (1) une version annotée de la recette au format XML initial mais avec des balises additionnelles, (2) une liste des concepts permettant d'indexer la recette exprimés en logique propositionnelle et (3) un rapport d'erreur mettant en évidence les lignes d'ingrédients de la recette initiale qui n'ont pas pu être annotées en utilisant la base terminologique (ce rapport est très informatif sur les failles éventuelles de la base utilisée). Ces trois points sont détaillés ci-après.

2.3.1 Recettes annotées.

Comme nous l'avons vu plus haut, les recettes du CCC sont représentées en XML mais sont très faiblement structurées. Des balises sont utilisées pour identifier le titre de la recette (élément TI), les ingrédients (éléments IN) et le corps de texte décrivant la préparation (élément PR). Le processus d'annotation permet donc d'ajouter des balises précisant chaque ingrédient de la recette. Il recherche dans chaque ligne d'ingrédient la présence de termes de la base terminologique et introduit, pour chaque terme trouvé, le concept impliqué. Par exemple, la ligne d'ingrédient `<IN>300 g mashed bananas</IN>` sera annoté comme : `<IN><ING>banana</ING><QT>300</QT><U>g</U><QL>mashed</QL><R/> </IN>` où `<ING>banana</ING>` est le concept associé au terme *bananas*, `<QT>300</QT>` est la quantité, `<U>g</U>` est l'unité, `<QL>mashed</QL>` est un "qualifier" et `<R/>` est le reste de la ligne d'ingrédient, c'est-à-dire l'ensemble des éléments qui n'ont pas été reconnus par le parseur étant données les informations contenues dans la base terminologique. Dans l'exemple présent, comme tout a été reconnu, le reste est vide. La ligne ainsi annotée est utilisée par les experts pour contrôler la correction de l'étape d'analyse du parseur.

Les différents types (types de plats, origines, moments des plats) d'une recette n'apparaissent pas explicitement dans la description initiale de la recette. Ces types doivent être inférés à partir des éléments disponibles. Dans TAAABLE, ceci est réalisé par un processus en trois étapes. Tout d'abord, on recherche si une recette ayant le même titre existe sur le site *recipesource.com*. Si tel est le cas, on récupère le maximum d'informations de typage depuis ce site. Si cette première étape échoue, on recherche si le titre de la recette contient des éléments qui en eux-mêmes représentent un type de plat ou une origine (par exemple, le *Banana Butterfinger cake* est sans aucun doute un *cake*). Enfin, si l'étape 2 échoue également, on utilise un ensemble de règles d'association du type *ingredient(s) → type ou origine* (par exemple, *mascarpone* \wedge *coffee* \rightarrow *tiramisu*) afin de typer la recette. Les règles d'association proviennent en partie d'une extraction réalisée sur l'ensemble des ressources disponibles dans la base de données du site *recipesource.com*, les autres règles sont définies manuellement en fonction des besoins observés.

L'annotation des recettes par les types est un processus relativement bruité. 30% des recettes n'ont aucun type affecté, certains types sont manquants (en particulier les moments des plats) et il y a certaines erreurs (par exemple, *pizza sauce* n'est pas vraiment une *pizza*). Afin d'éviter les comportements anormaux du système, les experts doivent vérifier manuellement les annotations de chaque recette.

2.3.2 Indexation des recettes sous forme propositionnelle.

Le langage de représentation des connaissances utilisé par le moteur de RÀPC est un fragment de logique propositionnelle. L'ontologie est représentée sous forme d'un ensemble d'axiomes du type

$a \Rightarrow b$. Par exemple, l'axiome $\text{apple} \Rightarrow \text{fruit}$ de l'ontologie O signifie que toute recette contenant des pommes est une recette contenant des fruits.

Toutes les recettes du livre de recettes sont indexées par une conjonction de littéraux. Par exemple, la recette de la tarte aux pommes, que nous noterons R est indexée par :

$$\text{Idx}(R) = \text{apple} \wedge \text{pie} \wedge \text{sugar} \wedge \text{pastry}$$

L'ensemble des recettes indexées constitue la base de cas du système.

2.3.3 Rapports d'erreurs.

Nous sommes confrontés à deux principaux types d'erreurs. Le premier provient des recettes « mal écrites » et l'autre provient des concepts manquants dans l'ontologie. Dans les deux cas, les erreurs doivent être corrigées afin que l'application fonctionne au mieux. Voici quelques exemples d'erreurs.

Recettes mal écrites. Certaines parties de certaines recettes requièrent des corrections car elles présentent des erreurs typographiques.

- $\langle \text{IN} \rangle 4 \text{ ts Baking power} \langle / \text{IN} \rangle$ n'a rien à voir avec le super pouvoir de cuire les choses mais devrait être remplacé par $\langle \text{IN} \rangle 4 \text{ ts Baking powder} \langle / \text{IN} \rangle$ qui, en revanche, a le pouvoir de faire lever les gâteaux ;
- les deux lignes d'ingrédients consécutives $\langle \text{IN} \rangle 1 \text{ lb Boneless pork, cut in } 3/4 \langle / \text{IN} \rangle$ et $\langle \text{IN} \rangle \text{Inch cubes} \langle / \text{IN} \rangle$ devraient être fusionnées en une seule $\langle \text{IN} \rangle 1 \text{ lb Boneless pork, cut in } 3/4 \text{ inch cubes} \langle / \text{IN} \rangle$,
- $\langle \text{IN} \rangle \text{Salt ; pepper, Worcestershire and lemon juice} \langle / \text{IN} \rangle$ devrait être découpé en $\langle \text{IN} \rangle \text{Salt} \langle / \text{IN} \rangle$, $\langle \text{IN} \rangle \text{pepper} \langle / \text{IN} \rangle$, $\langle \text{IN} \rangle \text{Worcestershire} \langle / \text{IN} \rangle$ et $\langle \text{IN} \rangle \text{lemon juice} \langle / \text{IN} \rangle$.

La plupart de ces erreurs ont été détectées par les experts alors qu'ils vérifiaient les recettes annotées.

Concepts absents de l'ontologie. Certaines lignes d'ingrédients n'ont été indexées par aucun des termes de la base terminologique. Cela signifie qu'aucun concept de type « ingrédient » n'a été reconnu dans cette ligne. Dans une telle situation, l'expert doit vérifier dans un premier temps que la ligne est écrite correctement. Si c'est le cas, l'erreur provient certainement de concepts manquants dans l'ontologie ou de termes manquants dans la base terminologique. Par exemple, si le concept spam^4 n'existe pas dans la hiérarchie des ingrédients, alors la ligne d'ingrédients $\langle \text{IN} \rangle 1/2 \text{ cn Spam, in } 3/4 \text{'' cubes} \langle / \text{IN} \rangle$ ne peut pas être indexée. L'expert doit alors ajouter le nouveau concept spam dans l'ontologie, définir sa position dans la hiérarchie, et associer ce concept à une liste de termes (spam, \dots) dans la base terminologique. Une fois toutes ces étapes effectuées, les différents éléments sont mis à jour et l'erreur ne devrait plus se reproduire lors de l'exécution du processus d'annotation suivant. Des problèmes identiques existent pour les hiérarchies de types, de moments et d'origine des plats.

2.4 Raisonnement à partir de cas

Envoyer des requêtes au système. Les requêtes exprimées en langue naturelle sont traitées grâce à une interface web et formalisées dans le système par une conjonction de littéraux. Pour cela, l'interface aide l'utilisateur à saisir la requête et à l'organiser de façon à ce qu'elle soit intelligible par le système. Pour reprendre l'exemple initial, la requête « je veux un dessert avec de la rhubarbe mais sans chocolat » est transformée en une requête notée Q telle que :

$$Q = \text{dessert} \wedge \text{rhubarb} \wedge \neg \text{chocolate}$$

⁴*Spam* signifie *spiced ham*, une sorte de jambon épicé en conserve.

Mécanismes de remémoration et d'adaptation. La première étape du processus de RÀPC consiste à retrouver parmi les recettes disponibles une recette similaire à la requête. La remémoration est effectuée par le biais d'une classification s'appuyant à la fois sur la requête et sur les recettes indexées. Dans un premier temps, une classification dure est appliquée : le système recherche une recette dont l'index correspond exactement à l'index de la requête. Si la classification dure échoue (c'est-à-dire si aucune recette de la base ne permet de répondre immédiatement et sans adaptation à la requête), une classification élastique est alors déclenchée : la requête est généralisée et exécutée jusqu'à ce qu'une solution satisfaisante soit obtenue [5]. La classification élastique conduit à une mise en correspondance approximative des résultats et implique donc l'adaptation des recettes retrouvées afin de répondre à la requête.

L'adaptation d'une recette s'appuie sur des connaissances d'adaptation. Dans la première version de TAAABLE, les connaissances d'adaptation sont uniquement présentes sous forme de substitutions d'ingrédients. Par exemple, une recette de tarte aux pommes peut être adaptée en une recette de tarte à la rhubarbe en substituant les pommes par la rhubarbe dans la recette originale. Cependant, nous pourrions souhaiter effectuer des adaptations plus complexes de cette recette (comme le fait d'ajouter du sucre à la recette adaptée pour adoucir le goût de la rhubarbe). Un de nos objectifs est de rendre cela possible dans la future version de l'application.

2.5 Synthèse : leçons tirées de TAAABLE

Il est évident que la base de connaissances utilisée par le moteur de RÀPC n'est ni correcte ni complète et doit en permanence être améliorée et mise à jour. Cependant, dans TAAABLE, l'indépendance des deux parties du système (*online* et *offline*) rend cette mise à jour très compliquée : rien ne peut être fait dynamiquement ni interactivement et aucune prise en compte « directe » du *feedback* n'est possible. C'est sans aucun doute une limitation importante de l'architecture de TAAABLE. D'un point de vue purement pratique, c'était également une limitation forte pendant le développement de l'application : la moindre modification d'un des éléments de la base de connaissances demandait un redéploiement total de l'application, ce qui est vite devenu ingérable et a généré beaucoup de problèmes qui auraient pu être évités.

Pour tenter de résoudre ces problèmes, nous avons décidé de lier les deux parties dans une nouvelle version de TAAABLE (que nous avons baptisée WIKITAAABLE). Dans WIKITAAABLE, le moteur de RÀPC et les outils de gestion des connaissances sont couplés et travaillent de concert. Les premiers effets bénéfiques sur l'évolution de la base de connaissances et la correction des erreurs sont déjà visibles au sein du groupe de travail.

Pourquoi avons nous choisi Semantic Media Wiki ? Des cas d'utilisation pratiques des wikis sémantiques ont déjà été présentés dans la littérature [4]. Cependant, dans notre projet, nous nous devons d'aller plus loin que ce qui est fait dans ces projets. Les simples outils de navigation et d'édition des bases de connaissances, ne sont pas suffisants car nous devons également intégrer un moteur de raisonnement à l'application et faire en sorte de ré-injecter directement le *feedback* des utilisateurs dans le système sans passer par des intermédiaires coûteux. En résumé, ce que nous devons faire c'est un outil permettant une intégration souple des interactions humain-humain et humain-machine. Les wiki semblent être des outils appropriés pour cela. Le wiki est un outil social destiné à apporter un support aux collaborations entre humains. Le wiki sémantique apporte l'avantage supplémentaire de permettre la construction collaborative de connaissances en impliquant à la fois les humains et les machines. Nous pensons que les wiki sémantiques sont donc des outils particulièrement prometteurs dans une perspective de partenariat humain-machine.

Avant de porter notre choix sur Semantic Media Wiki, nous avons étudié certaines alternatives. Par exemple, l'approche utilisée dans IkeWiki [7] est bien adaptée à nos besoins puisqu'elle s'appuie sur une ontologie de fond qui est pré-chargée dans la base de connaissances et qu'elle limite l'indexation des éléments aux prédicats définis dans la base de connaissances. L'approche utilisée dans Semantic

Media Wiki est également intéressante puisque les données sémantiques rendent le contenu du wiki traitable par des processus automatiques et que ces données sont noyées dans le texte du wiki et donc visibles par les utilisateurs sans problème.

Nous avons donc décidé d'utiliser Semantic Media Wiki pour plusieurs raisons. Tout d'abord, Semantic Media Wiki est basé sur Media Wiki, et utilise donc la même interface. Or, l'interface de Media Wiki est très populaire puisque ce wiki est utilisé par des sites mondialement connus comme : le site de la communauté RÀPC francophone⁵ le site de la communauté CBR internationale⁶ le site, moins connu, appelé wikipédia⁷. Par conséquent, nous pensons que la plupart des utilisateurs familiers de Media Wiki seront en mesure de s'approprier facilement l'interface. Par ailleurs, la forte communauté de développeurs autour des projets Media Wiki et Semantic Media Wiki nous apporte l'assurance de disposer d'outils fiables et évoluant rapidement.

Dans la section suivante, nous présentons WIKITAAABLE et nous montrons comment nous y avons intégré les différents éléments de TAAABLE : les bases de connaissances, le moteur de raisonnement à partir de cas et l'interface.

3 WIKITAAABLE : un wiki sémantique pour TAAABLE

Avec WIKITAAABLE, la nouvelle génération du système TAAABLE, nous nous attaquons à plusieurs problèmes que présentait l'application en utilisant Semantic Media Wiki (SMW) [8] comme un tableau noir pour la construction des connaissances. Le wiki sémantique nous apporte des solutions techniques simples à mettre en œuvre pour parcourir, éditer, valider et interroger la base de connaissances représentée dans le wiki. Des travaux dans ce sens ont été illustrés dans [4]. De plus, grâce à ce wiki, la base de connaissances peut être automatiquement mise à jour par les résultats produits par le moteur de RÀPC et par l'outil d'annotation automatique. La vue de l'architecture du système est présentée figure 4 et les différents composants de cette architecture sont détaillés ci-dessous.

Semantic Media Wiki. Semantic Media Wiki est utilisé comme un tableau noir partagé entre les utilisateurs, le moteur de raisonnement à partir de cas et les bots d'annotation et d'indexation des recettes (un bot est un petit programme chargé d'exécuter une tâche automatique). Le moteur et les bots s'appuient sur un ensemble de requêtes sémantiques prédéfinies pour collecter leurs entrées. La base de connaissances est représentée par un réseau sémantique de pages wiki. Les pages représentent les recettes indexées, l'ontologie des ingrédients et l'ontologie des types de plats (voir figure 5).

Interface web utilisateur Media Wiki. C'est l'interface par défaut de l'application. Grâce à cette interface, les utilisateurs peuvent ajouter de nouvelles recettes, modifier les ingrédients, les types de plats et les origines des plats.

Bot d'annotation des recettes. Le bot d'annotation des recettes parcourt les pages contenant des recettes, en extrait les informations sur les ingrédients et met à jour les pages de recettes avec les annotations sémantiques permettant l'indexation et la catégorisation des recettes. La navigation entre les recettes et leur mise à jour est effectuée grâce à l'API de Media Wiki et l'accès à la base de connaissances se fait par l'intermédiaire de requêtes sémantiques prédéfinies. La figure 6 montre une recette originale et la visualisation du résultat de son indexation par le bot dans Semantic Media Wiki. Les relations natives de Semantic Media Wiki sont utilisées pour représenter les différents éléments d'une ligne d'ingrédients.

RÀPC. Le moteur de RÀPC récupère la base de connaissances grâce à un ensemble de requêtes sémantiques prédéfinies. Ensuite, il répond aux requêtes émises depuis l'interface web en appliquant les principes du RÀPC. Les recettes proposées au travers de l'interface sont des recettes originales ou bien des adaptations de recettes existantes. Les utilisateurs sont invités à donner un

⁵<http://liris.cnrs.fr/rapc>

⁶<http://cbrwiki.fdi.ucm.es>

⁷<http://www.wikipedia.org>

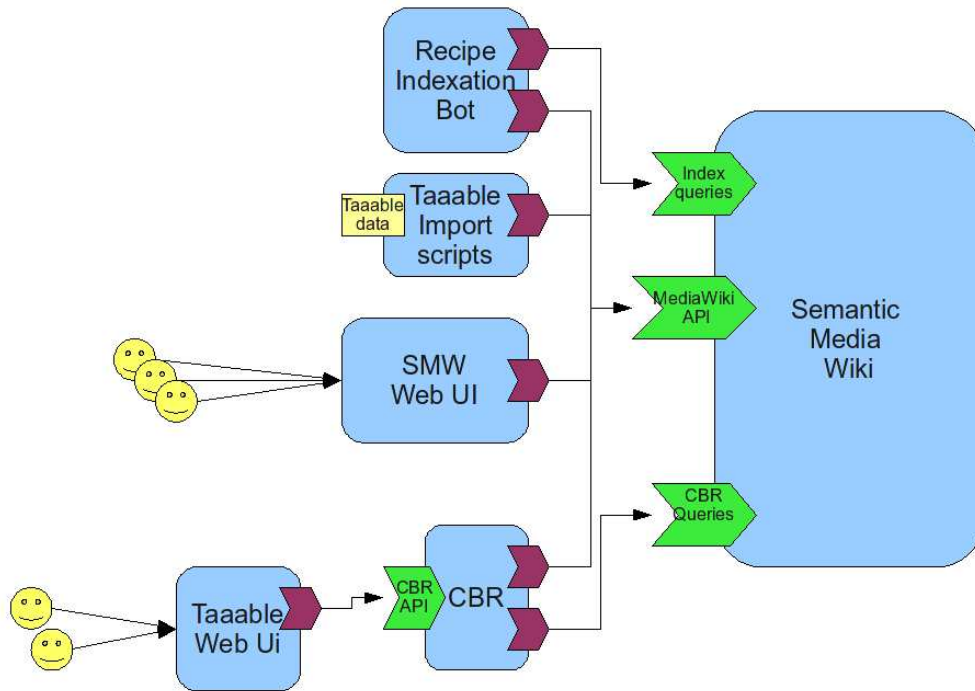


FIG. 4 – Les composants de WIKITAAABLE.

retour sur les différentes recettes. La figure 7 illustre une façon possible de capturer facilement les retours des utilisateurs. La capture d'écran n'est qu'une maquette et l'interface est en cours de développement. Cette figure illustre le scénario suivant. Supposons qu'un utilisateur souhaite une « *pie* » avec des « *oranges* » avec dans l'idée d'obtenir une tarte à l'orange. Le système propose alors les résultats présentés dans la figure 7. L'utilisateur choisit alors « *I don't like it* » pour la première proposition car il estime que remplacer le jus de citron par des oranges dans la recette « *apple crumble pie* » ne permet pas de transformer une « *apple crumble pie* » en une « *orange pie* ». En revanche, les proposition 2 et 3 sont acceptables. Ainsi, l'utilisateur choisit « *I like it* » pour ces deux propositions.

Si le *feedback* est positif, la nouvelle recette est ajoutée à la base de connaissances et une nouvelle page de wiki sémantique est créée pour la représenter. La nouvelle recette est marquée comme étant « générée » afin de ne pas être confondue avec les recettes du livre de recettes initial. Si le feedback de l'utilisateur est négatif, la recette générée est également ajoutée à la base de connaissances et une nouvelle page est créée. Toutefois, cette nouvelle page est rangée dans la catégorie « refusée ». Les recettes refusées sont gardées pour des usages futurs. Par exemple, les experts peuvent de temps à autres analyser les recettes refusées et les utiliser comme une base de travail pour un processus d'acquisition de connaissances dirigé par les échecs [3].

Scripts d'import de TAAABLE. Nous avons écrit des scripts permettant d'importer la base de connaissances actuelle de TAAABLE dans Semantic Media Wiki. La base de recettes contient environ 1400 recettes, l'ontologie des ingrédients contient 8506 classes différentes d'ingrédients. Enfin, nous utilisons *RAP - RDF API for PHP* pour parser l'ontologie.

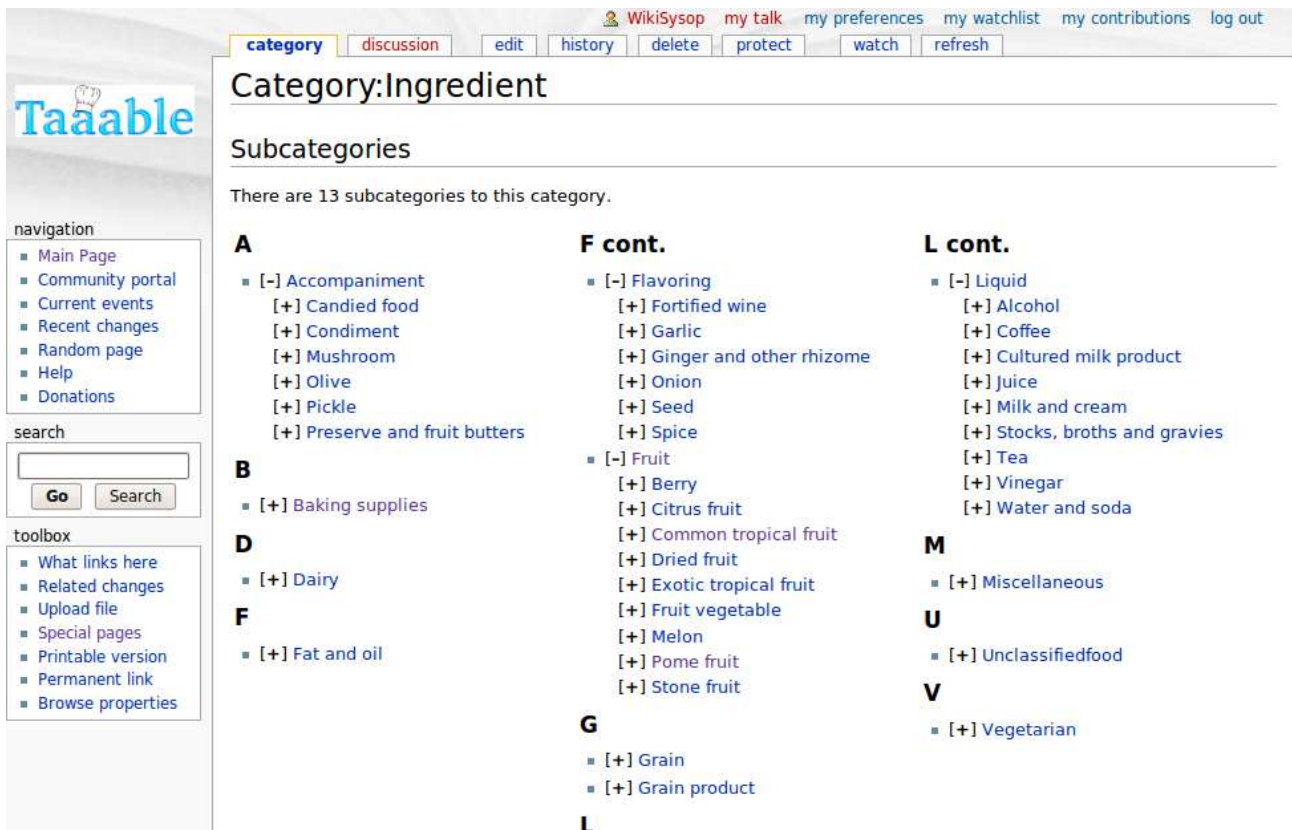


FIG. 5 – L'ontologie des ingrédients représentée sous WIKITAAABLE.

4 Discussion et travaux présents et futurs

Comme nous l'avons vu, les wiki sémantiques permettent à une communauté de mettre à jour et d'enrichir facilement un ensemble de recettes annotées ainsi qu'une ontologie. Le système WIKITAAABLE bénéficie de ces atouts et a de nombreux avantages par rapport à la version originale de TAAABLE.

- Les utilisateurs peuvent ajouter de nouvelles recettes facilement.
- Les utilisateurs peuvent corriger des recettes mal annotées.
- Les utilisateurs peuvent naviguer et parcourir l'ontologie. simplement.
- Les personnes responsables de la maintenance de l'ontologie peuvent rapidement modifier l'ontologie des ingrédients, des types de plats et tester les effets des modifications sur le moteur de raisonnement.
- Le *feedback* sur l'adaptation des recettes peut désormais être capturé et représenté dans les wiki sémantiques. La base de connaissances du système augmente du simple fait d'être utilisée.

Une des raisons du succès des systèmes de RÀPC est qu'ils sont théoriquement capables d'apprendre à partir de l'expérience en acquérant des connaissances additionnelles après chaque épisode de résolution de problème. Dans TAAABLE cependant, apprendre à partir de l'expérience est difficile en raison du manque de mécanismes embarqués permettant la prise en compte du *feedback* pour améliorer les bases de connaissances existantes. Dans WIKITAAABLE, l'utilisation du wiki sémantique facilite ce processus en permettant l'enrichissement et la mise à jour des données et des connaissances utilisées par le système par une communauté d'utilisateurs. Par exemple, les mainteneurs des ontologies peuvent facilement les modifier et tester l'impact de ces modifications sur les résultats produits par le moteur de RÀPC.

Le développement de WIKITAAABLE a soulevé de nombreuses questions qui sont principalement liées au maintien de la cohérence du système. Comment peut-on garantir la cohérence d'une application alors que plusieurs utilisateurs ayant souvent des points de vues différents, sont autorisés à modifier

== Ingredients ==

- * 1 c rice
- * 2 c water
- * 1/2 c sugar
- * 1 ts salt
- * 2 c evaporated milk
- * 1 c raisins
- * 3 eggs separated
- * 3/4 ts vanilla
- * 1/4 ts cinnamon
- * 1/4 ts nutmeg

== Preparation ==

* combine the rice water sugar and salt in a large saucepan bring the water to a boil and cover the saucepan reduce the heat to low and continue to cook for 12 - 15 minutes or until the water is absorbed combine the milk and egg yolk add them to the rice then mix in the raisin vanilla and cinnamon simmer for five minutes remove from the heat beat the egg white until stiff fold them into the rice chill and garnish with nutmeg before serving also taste good warm

The screenshot shows a web page for 'ARROZ DULCE' on the Taaable website. The page has a navigation bar at the top with links like 'page', 'discussion', 'edit', 'history', 'delete', 'move', 'protect', 'watch', 'refresh', 'WikiSysop', 'my talk', 'my preferences', 'my watchlist', 'my contributions', and 'log out'. The main content area is divided into sections: 'Ingredients' with a list of items and their categories (e.g., '1 c rice category:Rice (? , c, 1)'), 'Preparation' with instructions, and 'Facts about ARROZ DULCE' which shows a structured list of ingredients and their categories. The page footer includes a timestamp 'This page was last modified 19:42, 2 March 2009.', 'This page has been accessed 3 times.', 'Privacy policy', 'About Wikitaable', and a 'Powered By MediaWiki' logo.

FIG. 6 – Un exemple de recette annotée : « ARROZ DULCE ».

les connaissances codées dans le système ? Comment peut-on combiner efficacement une procédure semi-automatique et un processus d'enrichissement manuel pour construire une ontologie ? Ces questions ont une importance capitale puisque l'ontologie joue un rôle central dans le système TAAABLE et est utilisée à deux niveaux principaux (elle guide le processus d'indexation en identifiant les concepts impliqués dans chaque recette et elle est utilisée par le système de RÀPC pour adapter les recettes).

Si n'importe quel utilisateur peut librement modifier l'ontologie, le risque d'introduire des incohérences dans les connaissances est d'autant plus élevé. En conséquence, selon une telle hypothèse,

Ingredients

I want: ? I don't want: ?

Type of dish

I want: ? I don't want: ?

More options

Vegetarian
 Nut-free
 No alcohol
 [Advanced Configuration ?](#)

Your request: orange D:pie

Common path: 1-<citrus_fruit --> orange>

Common cost: 1.3778748590755354

#	Original recipe name	Adaptation overview	I like/I don't like	Cost
1	Apple Crumble Pie	Replace: lemon_juice by citrus_fruit	I don't like it !	5
2	Delicious Key Lime Pie	Replace: key_lime_juice by citrus_fruit, key_lime_peel by citrus_fruit	I like it !	5
3	Key Lime Pie	Replace: key_lime by citrus_fruit, key_lime_juice by citrus_fruit	I like it !	5
4	Strawberry Lime Pie	Replace: lime by citrus_fruit	I don't know	5
5	UPSIDE DOWN APPLE PIE	Replace: lemon_juice by citrus_fruit	I don't know	5

Results 1 - 5 on 5 | Processing time: 0.0141 secondes

t a @ a ble

[Help me!](#) |
 [More about Taaable](#) |
 [The team](#) |
 [Administration](#)

FIG. 7 – Capturer le *feedback* utilisateur grâce à l’interface de TAAABLE.

le moteur de RÀPC et les bots d’indexation et d’annotation produiront certainement des résultats imprévisibles. Plusieurs stratégies peuvent être imaginées pour éviter ce problème :

- Restreindre les droits de mise à jour de l’ontologie aux « administrateurs ». C’est une solution possible au problème, mais c’est également une limitation considérable au travail collaboratif. De plus, l’amélioration de l’ontologie dépend fortement de la disponibilité des dits administrateurs et risque donc de ne pas être faite souvent.
- Validation des changements avant qu’ils ne soient intégrés à la version en production du système. Cette stratégie a également quelques limites : le support de tels processus n’est pas trivial dans les wikis sémantiques (car ils ne sont pas conçus pour cela), les problèmes de synchronisation entre plusieurs versions d’un système unique sont importants, et pour finir, tout cela est très coûteux en temps et pas très motivant pour les administrateurs. De plus, cela ne résout pas le problème des conflits entre plusieurs changements en parallèle de l’ontologie.
- Adaptation d’approches d’intégration continue utilisées en génie logiciel dans le contexte du système WIKITAAABLE. Par exemple, si l’adaptation d’une recette a été validée par plusieurs utilisateurs, alors l’ontologie devrait être modifiée afin de préserver cette adaptation (dans ce cas, le *feedback* des utilisateurs collecté par WIKITAAABLE devrait générer des tests de non-régression).
- Vers un WIKITAAABLE pair à pair ? Dans une telle approche, chaque utilisateur pourrait avoir sa propre version de l’application s’appuyant à la fois sur une base de connaissances commune et sur une base de connaissances d’adaptation personnelle. Les connaissances d’adaptation pourraient alors être partagées entre plusieurs utilisateurs sur la base de la confiance qu’ils accordent à leurs pairs.

La prochaine étape du développement de WIKITAAABLE est d’intégrer totalement le moteur de RÀPC au wiki sémantique et de mettre en place des possibilités d’interactions à plusieurs niveaux. Une attention toute particulière sera portée à la capacité du wiki sémantique à représenter et gérer des règles d’adaptation complexes incluant, par exemple, des contraintes booléennes.

Références

- [1] F. Badra, R. Bendaoud, R. Bentebitel, P.A. Champin, J. Cojan, A. Cordier, S. Desprès, S. Jean-Daubias, J. Lieber, T. Meilender, A. Mille, E. Nauer, A. Napoli, et Y. Toussaint. TAAABLE : Text Mining, Ontology Engineering, and Hierarchical Classification for Textual Case-Based Cooking. In M. Schaaf, editor, *ECCBR 2008, The 9th European Conference on Case-Based Reasoning, Trier, Germany, September 1-4, 2008, Workshop Proceedings*, pages 219–228, 2008.
- [2] Michel Buffa, Fabien L. Gandon, Guillaume Ereteo, Peter Sander, et Catherine Faron. Sweetwiki : A semantic wiki. *Journal of Web Semantics*, 6(1) :84–97, 2008.
- [3] Amélie Cordier. *Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning*. PhD Thesis, Université Claude Bernard Lyon 1, November 2008.
- [4] Markus Krötzsch, Sebastian Schaffert, et Denny Vrandecic. Reasoning in semantic wikis. In Grigoris Antoniou, Uwe Aßmann, Cristina Baroglio, Stefan Decker, Nicola Henze, Paula-Lavinia Patranjan, et Robert Tolksdorf, editors, *Reasoning Web*, volume 4636 of *Lecture Notes in Computer Science*, pages 310–329. Springer, 2007.
- [5] J. Lieber. Strong, Fuzzy and Smooth Hierarchical Classification for Case-Based Problem Solving. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02), Lyon, France*, pages 81–85. IOS Press, Amsterdam, 2002.
- [6] C. K. Riesbeck et R. C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1989.
- [7] S. Schaffert. IkeWiki : A Semantic Wiki for Collaborative Knowledge Management. *1st International Workshop on Semantic Technologies in Collaborative Applications (STICA06), Manchester, UK, 2006*.
- [8] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, et Rudi Studer. Semantic wikipedia. *Journal of Web Semantics*, 5(4), 2007.