

---

# TAAABLE : un système de raisonnement à partir de cas qui adapte des recettes de cuisine

Emmanuelle Gaillard, Jean Lieber, Emmanuel Nauer

Université de Lorraine, LORIA — 54506 Vandœuvre-lès-Nancy, France

CNRS — 54506 Vandœuvre-lès-Nancy, France

INRIA — 54602 Villers-lès-Nancy, France

{prénom.nom@loria.fr}

---

**RÉSUMÉ.** TAAABLE est un système de raisonnement à partir de cas (RÀPC) qui utilise un livre de recettes de cuisine comme base de cas pour répondre à des requêtes culinaires. TAAABLE participe au Computer Cooking Contest depuis 2008 et a remporté de nombreux prix grâce à l'utilisation combinée de nombreuses approches issues des systèmes à base de connaissances telles que le RÀPC, la représentation, l'acquisition et la gestion des connaissances, l'extraction de connaissances à partir de données et le traitement automatique des langues. Dans cet article, nous décrivons TAAABLE et ses principaux modules. Nous présentons tout d'abord le moteur du système de RÀPC et ses caractéristiques, en particulier le processus de recherche fondé sur une généralisation minimale de la requête, ainsi que différents processus d'adaptation. Nous décrivons par la suite les différents conteneurs de connaissances utilisés par le système et la façon de les acquérir. Pour conclure, nous discutons des développements issus de ces travaux.

**ABSTRACT.** TAAABLE is a Case-Based Reasoning (CBR) system that uses a recipe book as a case base to answer cooking queries. TAAABLE participates in the Computer Cooking Contest since 2008, and succeeds thanks to a smart combination of various knowledge-based system methods and techniques, such as: CBR, knowledge representation, acquisition, discovery and management, and natural language processing. In this paper, we first present the CBR engine and its features such as the retrieval process based on minimal generalization of a query and different adaptation processes. Then the knowledge containers used by the system, and how they were acquired is detailed. To conclude, we discuss some research developments resulting from this application.

**MOTS-CLÉS :** raisonnement à partir de cas, ontologie, annotation sémantique, wiki sémantique, RDF(S), extraction de connaissances, traitement automatique des langues.

**KEYWORDS:** case-based reasoning, ontology, semantic annotation, semantic wiki, RDF(S), knowledge discovery, natural language processing.

---

DOI:10.3166/RIA.31.207-235 © 2017 Lavoisier

## 1. Introduction

TAAABLE est un système de raisonnement à partir de cas (RÀPC) qui retourne des recettes de cuisine en réponse à des requêtes posées par des utilisateurs, à partir d'un ensemble limité de recettes. Si aucune recette ne satisfait exactement la requête, TAAABLE recherche des recettes *similaires* et les adapte pour satisfaire la requête. TAAABLE a été initialement développé pour participer au *Computer Cooking Contest* (<http://computercookingcontest.net/>), une compétition annuelle organisée dans le cadre des conférences ICCBR (*International Conference on Case-Based Reasoning*) dont le but est de comparer les performances de différents systèmes informatiques sur des tâches de résolution de problèmes, dans un domaine d'application commun (la cuisine), avec un jeu de données commun (un livre de recette). Le jeu de données, fourni par les organisateurs du concours, consiste en un ensemble de recettes au format XML, tel que présenté en figure 1. Le schéma des données se limite à quelques balises pour identifier le titre (élément `TI`), les ingrédients (éléments `IN`) et des instructions textuelles de préparation (élément `PR`). Ce jeu de données, où ingrédients et préparation sont en langage naturel, requiert un prétraitement important afin de pouvoir être exploité par un processus de raisonnement.

TAAABLE participe à cette compétition depuis sa création en 2008 et a déjà remporté plusieurs prix, dans des défis où différents types d'adaptation étaient sollicités. Par exemple, le défi principal consiste à retourner une recette qui satisfait une requête composée d'aliments souhaités ou non, du type et de l'origine du plat, et/ou d'une compatibilité avec certains régimes (e.g. «Je veux une entrée végétarienne avec de l'endive, du jus de citron mais sans oignon»). Dans ce défi, l'adaptation consiste uniquement à substituer des ingrédients par d'autres, contrairement à d'autres défis, dans lesquels les quantités ou encore la préparation doivent être adaptées.

```
<RECIPE>
<TI>Glutinous Rice with Mangoes</TI>
<IN>3 c Glutinous rice</IN>
<IN>1 1/2 c Coconut cream</IN>
<IN>1/2 c Sugar</IN>
<IN>1 ts Salt</IN>
<IN>1 1/4 c Coconut cream</IN>
<IN>2 tb Sugar</IN>
<IN>1/4 ts Salt</IN>
<IN>6 Ripe mangoes, well chilled</IN>
<IN>2 tb Sesame seeds, toasted</IN>
<PR>SEASONINGS SAUCE GARNISH Soak the rice in cold water for 2 hours. Drain.
Line a steamer with cheesecloth, heat steamer and lay rice on the cheesecloth.
Steam for 30 minutes or until cooked through. The rice will become glossy.
Mix the SEASONINGS ingredients in a large bowl and gently mix in the hot
steamed rice. Cover tightly and let soak for 30 minutes to absorb the coconut
flavour. Blend the SAUCE ingredients in a pot and heat until it just reaches
the boiling point. Let cool. Peel the mangoes, slice lengthwise and remove the
pits. Divide the rice among 6 plates. Place mango slices on top and cover with
the sauce. Sprinkle with the sesame seeds and serve.</PR>
</RECIPE>
```

Figure 1. Un exemple de recette issue du jeu de données fourni par le CCC

**Taaable**

Dietary practices:  Vegetarian  Nut-free  No alcohol  Low cholesterol  Gout Diet [Customize your dietary practices...](#)

[Adapt a specific recipe...](#)  
**Example.** If you want an apple pie without cinnamon, enter "apple pie\_dish -cinnamon". [Learn more about advanced queries...](#)

---

Your request is: **dessert\_dish rice fig**  
 The request used for adaptation is: **dessert\_dish rice fig**

#	Original recipe name (click to open recipe)	Adaptation overview (click to see the details)
1	<a href="#">Glutinous rice with mangoes</a>	Replace: Mango by Fig

Results 1 - 1 on 1 | Processing time: 0.6189 seconds

Figure 2. Une illustration de l'interface utilisateur de TAAABLE

**Glutinous rice with mangoes**

---

**The ingredient substitutions**    Mango → Fig

---

**Ingredient Quantities**

Ingredient	Initial Quantity	New Quantity
<a href="#">Salt</a>	=	1 tsp (7.0 grams)
<a href="#">Mango</a>	6 whole (1242.0 grams)	0 whole (0.0 grams)
<a href="#">Coconut cream</a>	=	2 cup (648.0 grams)
<a href="#">Glutinous rice</a>	=	3 cup (522.0 grams)
<a href="#">Sesame seed</a>	2 tbsp (16.0 grams)	1 tbsp (10.0 grams)
<a href="#">Fig</a>	0 grams (0.0 grams)	1242 grams
<a href="#">Sugar</a>	=	2 cup (625.0 grams)

---

**Preparation Adaptation**

Cut figs into wedges. SEASONINGS SAUCE GARNISH Soak the rice in cold water for 2 hours Drain Line a steamer with cheesecloth, heat steamer and lay rice on the cheesecloth Steam for 30 minutes or until cooked through The rice will become glossy Mix the SEASONINGS\_ingredients in a large bowl and gently mix in the hot steamed rice Cover tightly and let soak for 30 minutes to absorb the coconut flavour Blend the SAUCE\_ingredients in a pot and heat until it just reaches the boiling point Let cool ~~Peel the mangoes, slice lengthwise and remove the pits~~ Divide the rice among 6 plates Place ~~mango~~-fig slices on top and cover with the sauce Sprinkle with the sesame seeds and serve

Figure 3. Un exemple de résultat d'adaptation

L'organisation de ce document est la suivante. La section 2 donne une vue générale du fonctionnement du système TAAABLE. Le moteur d'inférences de TAAABLE qui implémente les opérations fondamentales du RÀPC est ensuite détaillé : la section 3 présente le processus de remémoration, la section 4 présente différents processus d'adaptation. La section 5 présente comment les connaissances utilisées par les processus d'adaptation sont gérées et comment elles ont été acquises. La section 6 fournit des éléments d'évaluation sur TAAABLE, comparativement à d'autres systèmes ayant participé au concours. La section 7 conclut par une discussion sur les liens entre

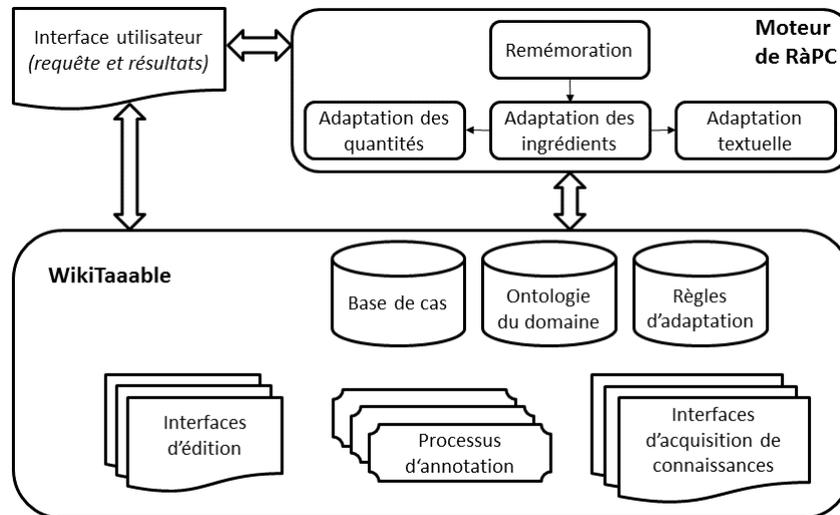


Figure 4. Architecture générale du système TAAABLE

TAAABLE et le domaine de l'intelligence artificielle (IA) et notamment comment les travaux menés dans cette application culinaire ont donné naissance à des outils génériques utilisables dans d'autres domaines.

## 2. Description du système TAAABLE

La figure 2 présente l'interface utilisateur de TAAABLE, sur un exemple d'interrogation d'un dessert avec du riz et des figues, mais sans régime particulier. La recette *Glutinous rice with mangoes* est retournée avec l'adaptation qui consiste à remplacer la mangue par de la figue. Le langage de requêtes est inspiré de celui des interfaces de moteur de recherche sur le web. Si l'utilisateur emploie un terme hors du vocabulaire du système, une alerte est affichée et ce terme n'est pas pris en compte dans la requête.

L'utilisateur peut consulter la recette originale ainsi que le détail de l'adaptation en cliquant sur les liens adéquats. La figure 3 présente le résultat de l'adaptation de la recette. La première partie de l'adaptation concerne l'adaptation des ingrédients, la deuxième, l'adaptation des quantités, la troisième, l'adaptation du texte de la préparation. Ces trois types d'adaptation ainsi que les processus associés sont détaillés en section 4.

L'architecture du système TAAABLE, présentée en figure 4, fait intervenir trois composants majeurs. La partie centrale du système est le moteur de RàPC qui intègre le processus de remémoration, présenté en section 3 ainsi que les différents processus d'adaptation présentés en section 4. Le moteur de raisonnement exploite une base de connaissances constituée de différents conteneurs : la base de cas (les recettes), les connaissances du domaine sous la forme d'une ontologie, les connaissances d'adapta-

tion sous la forme de règles d'adaptation. L'intégralité des connaissances est gérée via un wiki sémantique, nommé WIKITAAABLE (cf. section 5.3), facilitant l'édition et la gestion des connaissances. WIKITAAABLE intègre des interfaces spécifiques pour l'édition des connaissances, pour l'acquisition de connaissances d'adaptation ainsi que des processus d'annotation qui permettent de passer des représentations textuelles des recettes à une représentation formelle. Enfin, l'interface de requête/résultats qui vient d'être présentée permet à l'utilisateur d'interagir avec le moteur de RÀPC pour l'interroger. Des illustrations de certaines interfaces sont données en section 5.3 et section 5.4

### 3. Le processus de remémoration

Dans TAAABLE, la remémoration de recettes consiste à sélectionner des recettes  $R$  de la base de recettes  $Recettes$  (utilisée comme base de cas), qui satisfont au mieux la requête  $Q$ . Cet ensemble de recettes est obtenu en recherchant une généralisation  $\Gamma(Q)$  de  $Q$  qui est minimale au regard d'une fonction de coût et telle qu'il existe au moins une recette qui satisfait exactement  $\Gamma(Q)$ . Ce processus de recherche utilise les connaissances du domaine  $CD$ .

**Les connaissances du domaine**  $CD$  peuvent être considérées comme un ensemble d'axiomes  $a \Rightarrow b$  en logique des propositions, où  $a$  et  $b$  sont des variables propositionnelles représentant des classes de recettes. Par exemple, `citron` (resp., `agrume`) représente la classe des recettes contenant du citron (resp., des agrumes) et l'axiome `citron` $\Rightarrow$ `agrume` établit que toute recette avec des citrons est une recette avec des agrumes. Concrètement, chaque nom d'aliment  $x$  est interprété comme «la classe des recettes contenant l'aliment  $x$ ». Il y a également d'autres variables propositionnelles, telles que `mediterrannée` qui représente la classe des recettes méditerranéennes. Par conséquent, `huileOlive` $\Rightarrow$ `mediterrannée` établit que chaque recette contenant de l'huile d'olive est une recette méditerranéenne. D'un point de vue de l'implémentation,  $CD$  est une hiérarchie, i.e. un graphe direct sans circuit dont les nœuds sont des variables propositionnelles et les arcs  $a \longrightarrow b$  représentent les axiomes  $a \Rightarrow b$ . La racine de la hiérarchie, notée  $\top$ , désigne l'univers des recettes.

La section 5.1 présente l'acquisition de l'ontologie du domaine  $\mathcal{O}$ , constituée de hiérarchies de classes organisées par une relation de spécialisation notée  $\sqsubseteq$ . Par exemple, il existe une hiérarchie d'aliments dans  $\mathcal{O}$  déclarant, en particulier, que `Citron`  $\sqsubseteq$  `Agrume`. Les connaissances du domaine  $CD$  exploitées par les processus de remémoration et d'adaptation de cas proviennent de cette ontologie avec un changement de formalisme (du langage de l'ontologie à celui de la logique propositionnelle). En particulier, `citron` représente les recettes avec du citron et, comme `Citron` est une sous-classe de `Agrume`, chaque recette avec du citron est une recette avec de l'agrume, d'où l'axiome `citron` $\Rightarrow$ `agrume` dans  $CD$ . La figure 5 présente un extrait des connaissances du domaine (les valeurs sur les arcs représentent des coûts qui seront expliqués par la suite).

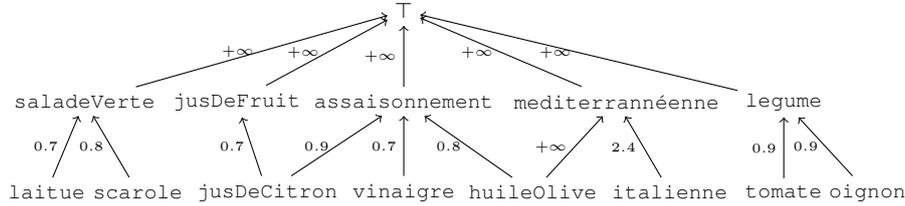


Figure 5. Un extrait des connaissances du domaine avec les coûts de généralisation. Un arc  $a \xrightarrow{c} b$  signifie que  $a \Rightarrow b \in CD$  et  $\text{coût}(a \rightsquigarrow b) = c$

**La base de recettes** *Recettes* est la base de cas du système. Ces recettes sont fournies par les organisateurs du CCC (il y avait environ 900 recettes pour le premier CCC et environ 1500 pour les CCC suivants). Une recette  $R \in \text{Recettes}$  est encodée à travers un document XML qui doit être transformé dans le formalisme de représentation de TAAABLE. Pour la remémoration des cas,  $R$  est transformée, via un processus d’annotation présenté en section 5.1, en  $\text{idx}(R)$ , l’index de  $R$  qui est une conjonction de littéraux. Par exemple :

$$\text{idx}(R) = \text{laitue} \wedge \text{vinaigre} \wedge \text{huileOlive} \wedge \text{tomate} \wedge \text{rien d'autre} \quad (1)$$

est une représentation formelle et abstraite de la recette  $R$  contenant de la laitue, du vinaigre, de l’huile d’olive, de la tomate et rien d’autre. L’hypothèse du monde clos est associée à  $\text{idx}(R)$  : si une propriété ne peut pas être inférée de  $\text{idx}(R)$  et des connaissances du domaine  $CD$ , alors elle est considérée comme fausse. Autrement dit, si  $\text{idx}(R) \not\models_{CD} a$  alors le terme *rien d’autre* est une conjonction de littéraux contenant le littéral  $\neg a$ . Dans l’exemple précédent,  $\text{idx}(R) \models_{CD} \neg \text{meat} \wedge \neg \text{fish}$ . On peut également souligner que  $\text{idx}(R)$  a exactement un modèle : pour chaque variable propositionnelle  $a$ , soit  $\text{idx}(R) \models_{CD} a$ , soit  $\text{idx}(R) \models_{CD} \neg a$ .

En l’absence de l’hypothèse du monde clos représentée par le terme *rien d’autre*, la recette de l’équation (1) ne répond pas, par exemple, à la recherche d’une recette végétarienne (exprimée par la conjonction  $\neg \text{meat} \wedge \neg \text{fish}$ ) bien que ce soit une recette végétarienne. Formellement, si  $\varphi$  est la conjonction  $\text{idx}(R)$  sans le terme *rien d’autre*, alors  $\varphi \not\models_{CD} \neg \text{meat} \wedge \neg \text{fish}$ .

**Une requête  $Q$**  est représentée comme une conjonction de littéraux. Par exemple,

$$Q = \text{scarole} \wedge \text{jusDeCitron} \wedge \neg \text{oignon} \quad (2)$$

représente la requête « Je veux une recette avec de la scarole et du jus de citron (parmi les ingrédients de la recette) mais sans oignon ».

**Remémoration**  $(Q, \text{Recettes}, \text{CD}) : (\{R_i\}_i, \Gamma)$

**Entrée** requête  $Q$ , base de recettes  $\text{Recettes}$ , connaissances du domaine  $\text{CD}$

**Sortie** un ensemble de recettes  $\{R_i\}_i \subseteq \text{Recettes}$  et une fonction de généralisation  $\Gamma$  telle que chaque  $R_i$  satisfait  $\Gamma(Q)$

- 1:  $\Gamma \leftarrow$  fonction identité
- 2:  $\{R_i\}_i \leftarrow \emptyset$
- 3: **Tant que**  $\{R_i\}_i = \emptyset$  **faire**
- 4:  $\Gamma \leftarrow$  généralisation suivante
- 5:  $\{R_i\}_i \leftarrow \{R_i \in \text{Recettes} \mid \text{idx}(R_i) \models_{\text{CD}} \Gamma(Q)\}$
- 6: **Fin tant que**

Algorithme 1: Algorithme de remémoration de TAAABLE.

**La remémoration de recettes** est décrite dans l'algorithme 1 qui est une application de classification élastique telle que décrite dans Lieber (2002). Elle consiste à trouver une généralisation  $\Gamma(Q)$  de  $Q$  qui satisfait au moins une recette de la base de recettes (i.e. il existe  $R \in \text{Recettes}$  telle que  $\text{idx}(R) \models_{\text{CD}} \Gamma(Q)$ ).

La généralisation  $\Gamma(Q)$  de  $Q$  est recherchée dans l'espace des états, où chaque état est une conjonction de littéraux, l'état initial est  $Q$  et les successeurs de l'état  $e$  forment l'ensemble des états  $\gamma(e)$  où  $\gamma$  est une transition d'état qui est :

- soit une étape de généralisation d'un littéral positif en  $e$  selon l'ontologie, e.g. si  $e$  contient le littéral `citron`, alors  $\gamma = \text{citron} \rightsquigarrow \text{agrume}$  ( $x \rightsquigarrow y$  est une substitution de  $x$  par  $y$ ) est une des  $\gamma$  si  $\text{citron} \Rightarrow \text{agrume}$  est un axiome des connaissances du domaine ;
- soit une suppression d'un littéral négatif de  $e$ , e.g. si  $e$  contient  $\neg \text{garlic}$ , alors  $\gamma = \neg \text{garlic} \rightsquigarrow \top$  est une des  $\gamma$ .

Ainsi, chaque généralisation de  $Q$  est un  $\Gamma(Q)$  où  $\Gamma$  est une composition de  $\gamma$ . Cet espace d'état est parcouru selon une valeur croissante de  $\text{coût}(\Gamma)$  où  $\text{coût}$  est une fonction de coût additive ( $\text{coût}(\Gamma_2 \circ \Gamma_1) = \text{coût}(\Gamma_1) + \text{coût}(\Gamma_2)$ ) telle que le coût de la fonction identité est nulle et  $\text{coût}(\gamma) > 0$  pour chaque transition d'état  $\gamma$ . Ainsi, la «généralisation suivante» mentionnée en ligne 4 est une généralisation du coût immédiatement plus grand que le coût de  $\Gamma$ . Techniquement, la généralisation suivante est le premier élément d'une liste de généralisations triée par coût croissant, comme dans une recherche A\* classique (Pearl, 1984).

La ligne 5 de l'algorithme est implémentée dans TAAABLE en utilisant une classification hiérarchique de  $\Gamma(Q)$  dans une hiérarchie contenant les index  $\text{idx}(R)$  des recettes. Ce mécanisme est une inférence déductive classique décrite par exemple dans Baader *et al.* (1992).

**La fonction de coût** doit être définie pour chaque  $a \rightsquigarrow b$  telle que  $a \Rightarrow b \in \text{CD}$ . En effet, chaque généralisation  $\Gamma$  est une composition de  $n$  substitutions  $a_i \rightsquigarrow b_i$ ,

et  $\text{coût}(\Gamma) = \sum_{i=1}^n \text{coût}(a_i \rightsquigarrow b_i)$  (car  $\text{coût}$  est additive). Avec l’hypothèse selon laquelle la base de recettes `Recettes` constitue un échantillonnage homogène de recettes dans l’espace des recettes,  $\mu(x) = \frac{\mathcal{N}(x)}{\mathcal{N}(\top)}$  est une bonne approximation de la proportion des recettes de la classe  $x$ , où  $\mathcal{N}(x)$  est le nombre de recettes de la classe  $x$  :

$$\mathcal{N}(x) = |\{R \in \text{Recettes} \mid \text{id}_x(R) \models_{\text{CD}} x\}|$$

Ainsi, si  $a \Rightarrow b \in \text{CD}$ ,  $\mu(b) - \mu(a)$  est la proportion des recettes de la classe  $b$  qui ne sont pas des recettes de la classe  $a$ , et donc une caractérisation du risque associé à la généralisation  $a \rightsquigarrow b$ . C’est pourquoi, une première idée consiste à définir  $\text{coût}(a \rightsquigarrow b) = \mu(b) - \mu(a)$ . Un coût plus précis peut être défini à partir du principe de remémoration guidée par l’adaptation (Smyth, Keane, 1996) dont l’idée est de prendre en compte des connaissances d’adaptation dans la phase de remémoration. En particulier, il est intéressant de faire remarquer que si  $a$  et  $b$  sont des classes relatives à des ingrédients (e.g.  $a = \text{citron}$  et  $b = \text{agrume}$ ) alors l’adaptation consistera en des substitutions tandis que si  $a$  et  $b$  sont, par exemple, des classes relatives à l’origine des recettes (e.g.  $a = \text{italienne}$  et  $b = \text{mediterrannéenne}$ ) alors l’adaptation sera une simple copie de la recette source, induisant plus de risques. C’est pourquoi, le coût d’adaptation dépend du *type*  $\tau$  des variables propositionnelles  $a$  et  $b$ . Ce type est associé à chaque variable propositionnelle, e.g. le type de `citron` et de `agrume` est  $\tau = \text{ingrédient}$  et le type de `italienne` et de `mediterrannéenne` est  $\tau = \text{origine}$ . Ainsi, un coefficient  $K_\tau$  dépendant du type  $\tau$  des classes  $a$  et  $b$  est utilisé et le coût est défini par :

$$\text{coût}(a \rightsquigarrow b) = K_\tau \frac{\mathcal{N}(b) - \mathcal{N}(a)}{\mathcal{N}(\top)} \quad \text{avec } \tau \text{ le type de } a \text{ et } b$$

Avec l’hypothèse selon laquelle l’adaptation relative aux ingrédients est moins risquée que celle relative aux origines,  $K_{\text{ingrédient}}$  est beaucoup plus petit que  $K_{\text{origine}}$ . En pratique, les valeurs suivantes donnent de bons résultats :  $K_{\text{ingrédient}} = 100$  et  $K_{\text{origine}} = 1000$ .

Deux cas particuliers, liés à des généralisations interdites, sont également pris en compte : quand  $a$  et  $b$  sont de types différents (e.g.  $\text{coût}(\text{huileOlive} \rightsquigarrow \text{mediterrannéenne}) = +\infty$ ) et quand  $b = \text{Aliment}$  car la généralisation à `Aliment`, la classe la plus générale de la hiérarchie des aliments, est trop risquée (e.g.  $\text{coût}(\text{viande} \rightsquigarrow \text{Aliment}) = +\infty$ ) en raison de la substitution qui serait alors produite par le processus d’adaptation détaillé en 4.1, consistant à remplacer un ingrédient par un ingrédient de nature différente (e.g. un liquide par une viande).

Tous les coûts sont stockés dans le wiki. Les coûts ont été initialement stockés par l’utilisation directe de la formule ci-dessus, le wiki sémantique permettant d’exprimer des valeurs de propriétés calculées. Ainsi, les valeurs des coûts évoluent dynamiquement en fonction de la modification de la base de recettes. Ces coûts peuvent ensuite être modifiés manuellement dans le wiki en raison de mauvais résultats, pour ajuster

le comportement du système : la modification d'un coût est une modification du risque de généralisation. Cette façon d'appréhender la gestion des coûts est une approche pratique qui a donné de bons résultats. Cependant, une étude théorique sur l'estimation des coûts reste à réaliser et constitue une perspective de ce travail.

**Par exemple**, considérons les connaissances du domaine  $\text{CD}$  avec les coûts donnés en figure 5, la requête de l'équation (2) et une base de cas contenant uniquement un cas: la recette  $R$  indexée par  $\text{idx}(R)$  de l'équation (1). Le processus de remémoration produit les fonctions de généralisation  $\Gamma_t$  suivantes (l'opérateur de composition est noté  $\circ$ ) :

$\Gamma_0 =$ fonction identité	(coût 0)
$\Gamma_1 =$ jusDeCitron $\rightsquigarrow$ jusDeFruit	(coût 0,7)
$\Gamma_2 =$ scarole $\rightsquigarrow$ saladeVerte	(coût 0,8)
$\Gamma_3 =$ jusDeCitron $\rightsquigarrow$ assaisonnement	(coût 0,9)
$\Gamma_4 =$ jusDeCitron $\rightsquigarrow$ jusDeFruit $\circ$ scarole $\rightsquigarrow$ saladeVerte	(coût 1,5)
$\Gamma_5 =$ jusDeCitron $\rightsquigarrow$ assaisonnement $\circ$ scarole $\rightsquigarrow$ saladeVerte	(coût 1,7)

Ainsi, le résultat du processus de remémoration est  $(\{R\}, \Gamma)$  avec  $\Gamma = \Gamma_5$  car  $\text{idx}(R) \models_{\text{CD}} \Gamma_5(Q)$  et  $\text{idx}(R) \not\models_{\text{CD}} \Gamma_t(Q)$  pour  $t < 5$ .

#### 4. Processus d'adaptation

Soit  $\{R_i\}_i$  l'ensemble des recettes remémorées. Étant donné une recette remémorée  $R \in \{R_i\}_i$  et une requête  $Q$ , l'adaptation vise à identifier les modifications de  $R$  afin qu'elle réponde à la requête  $Q$ . Plusieurs processus d'adaptation ont été implémentés dans TAAABLE. Le premier processus d'adaptation (section 4.1) utilise uniquement les résultats de la phase de remémoration :  $R$  et  $\Gamma$ . Le deuxième processus d'adaptation (section 4.2) utilise des connaissances d'adaptation qui sont des règles de substitutions d'ingrédients. Ces deux processus d'adaptation produisent des substitutions de types d'ingrédients, sans s'attacher aux quantités. À l'inverse, un troisième processus d'adaptation (section 4.3) traite de l'adaptation des quantités. Enfin, un processus qui adapte la préparation de  $R$  (et le texte de la préparation associé) est présenté à la section 4.4.

##### 4.1. Adaptation issue de la remémoration

L'adaptation issue de la remémoration utilise uniquement  $R$ ,  $\Gamma$  et  $Q$  et aucune autre unité de connaissances. Elle est fondée sur la séquence de relations suivantes, qui sont satisfaites puisque  $R \in \{R_i\}_i$  et  $(\{R_i\}_i, \Gamma)$  est le résultat du processus de remémoration :

$$\text{idx}(R) \models_{\text{CD}} \Gamma(Q) \xleftarrow{\Gamma} Q \quad (3)$$

La séquence de relations fait intervenir une correspondance entre les ingrédients de la recette et les littéraux positifs de la requête  $Q$ . Pour l'exemple courant, cette correspondance fait intervenir la composition suivante :

$$\begin{aligned} & \text{laitue} \rightsquigarrow \text{saladeVerte} \circ \text{vinaigre} \rightsquigarrow \text{assaisonnement} \\ & \circ \text{saladeVerte} \rightsquigarrow \text{scarole} \circ \text{assaisonnement} \rightsquigarrow \text{jusDeCitron} \end{aligned} \quad (4)$$

La première ligne est relative à  $\models_{\text{CD}}$ . La seconde ligne est relative à  $\stackrel{\Gamma}{\leftarrow}$ , i.e., à  $\stackrel{\Gamma^{-1}}{\mapsto}$ . Cette composition peut être simplifiée (en utilisant l'équation  $a \rightsquigarrow b \circ b \rightsquigarrow c = a \rightsquigarrow c$  et des permutations licites dans la composition) en :

$$\text{laitue} \rightsquigarrow \text{scarole} \circ \text{vinaigre} \rightsquigarrow \text{jusDeCitron} \quad (5)$$

qui décrit quelles substitutions peuvent être effectuées sur la recette  $R$  pour répondre à la requête  $Q$ .

Plus généralement, l'adaptation issue de la remémoration consiste à construire une composition de substitutions telles qu'en (4) puis de la simplifier (si cela est possible) pour obtenir les substitutions à appliquer à la recette remémorée  $R$ . Techniquement, les éléments des substitutions sont identifiés en deux étapes :

1. Suivant  $\text{idx}(R) \models_{\text{CD}} \Gamma(Q)$ , des substitutions de généralisation sont construites. En effet, il peut être démontré, pour  $\varphi$  et  $\chi$ , deux conjonctions de littéraux, que si  $\varphi \models_{\text{CD}} \chi$  alors il existe une composition  $G$  de substitutions de la forme  $a \rightsquigarrow b$  où  $a \Rightarrow b \in \text{CD}$  tel que  $G(\varphi)$  est équivalent à  $\chi$  (selon les connaissances du domaine CD).
2. Suivant  $\Gamma(Q) \stackrel{\Gamma}{\leftarrow} Q$ , des substitutions de spécialisation  $\gamma^{-1}$  sont construites : si  $\Gamma = \gamma_q \circ \dots \circ \gamma_1$  alors  $\Gamma^{-1} = \gamma_1^{-1} \circ \dots \circ \gamma_q^{-1}$ . Une généralisation  $\gamma = a \rightsquigarrow b$  conduira à la spécialisation  $\gamma^{-1} = b \rightsquigarrow a$ . Une généralisation  $\gamma = \neg a \rightsquigarrow \top$  conduira à la spécialisation  $\gamma^{-1} = \top \rightsquigarrow \neg a$ , i.e., à la suppression de  $a$  dans la recette.

#### 4.2. Adaptation par règles

Le moteur d'adaptation de TAAABLE exploite également des connaissances d'adaptation sous la forme de règles de substitution. Une telle règle établit que dans un contexte donné  $\mathcal{C}$ , certains ingrédients  $\mathcal{D}$  peuvent être remplacés par d'autres ingrédients  $\mathcal{P}$ . Par exemple, considérons l'unité de connaissance suivante : dans une recette de salade verte, le vinaigre peut être remplacé par du jus de citron et du sel. Cette unité de connaissance peut être représentée par une règle d'adaptation avec  $\mathcal{C} = \text{saladeVerte}$ ,  $\mathcal{D} = \text{vinaigre}$  et  $\mathcal{P} = \text{jusDeCitron} \wedge \text{sel}$ . Une adaptation de ce type peut être encodé comme une substitution  $\sigma = \mathcal{C} \wedge \mathcal{D} \rightsquigarrow \mathcal{C} \wedge \mathcal{P}$ . Pour l'exemple :

$$\text{saladeVerte} \wedge \text{vinaigre} \rightsquigarrow \text{saladeVerte} \wedge \text{jusDeCitron} \wedge \text{sel} \quad (6)$$

Soit  $CA$  les connaissances d'adaptation, i.e., un ensemble fini de substitutions  $\sigma$  représentant des règles d'adaptation. Un coût associé à chaque  $\sigma \in CA$ ,  $\text{coût}(\sigma) > 0$ ,

est aussi supposé connu. Étant donné  $CA$ , les connaissances du domaine  $CD$ , une requête  $Q$ , et une recette mémorisée  $R$ , l'adaptation à partir de règles combine l'utilisation des règles d'adaptation et de la généralisation de la requête suivant  $CD$ . Elle a pour but de construire une séquence de relations de la forme :

$$idx(R) \xrightarrow{\Sigma} \Sigma(idx(R)) \models_{CD} \Lambda(Q) \xleftarrow{\Lambda} Q \quad (7)$$

où  $\Sigma$  est une composition (éventuellement vide) de règles d'adaptation  $\sigma \in CA$ ,  $\Lambda$  est une fonction de généralisation (qui peut être différente de la fonction de généralisation  $\Gamma$  retournée par le processus de mémorisation) et  $\text{coût}(\Sigma) + \text{coût}(\Lambda)$  est minimal.

La séquence (7) est appelée un *chemin d'adaptation*. L'adaptation consiste à suivre ce chemin : premièrement, la composition  $\Sigma$  des règles d'adaptation est appliquée à la recette  $R$ , puis les généralisations correspondant à  $\Sigma(idx(R)) \models_{CD} \Lambda(Q)$  sont appliquées, et enfin les spécialisations correspondant à  $\Lambda(Q) \xleftarrow{\Lambda} Q$  sont appliquées. On peut souligner que les deuxième et troisième étapes correspondent au chemin d'adaptation (7) quand  $\Sigma =$  fonction identité et  $\Lambda = \Gamma$ .

Par conséquent, la principale difficulté algorithmique de l'adaptation à base de règles est de construire un chemin d'adaptation. Une fois encore, la technique utilisée est fondée sur une recherche au meilleur d'abord dans l'espace d'états. Dans ce contexte, un état est un couple  $(\Sigma, \Lambda)$ , l'état initial correspond à  $\Sigma = \Lambda =$  fonction identité,  $(\Sigma, \Lambda)$  est l'état final si (7) est satisfaite, le coût associé à un état  $(\Sigma, \Lambda)$  est  $\text{coût}(\Sigma) + \text{coût}(\Lambda)$  et les successeurs de l'état  $(\Sigma, \Lambda)$  sont les états  $(\Sigma', \Lambda)$  et les états  $(\Sigma, \Lambda')$  tels que :

- les substitutions  $\Sigma'$  sont telles que  $\Sigma' = \sigma \circ \Sigma$  avec  $\sigma \in CA$  et  $\sigma$  est applicable à  $\Sigma(idx(R))$  ;
- les substitutions  $\Lambda'$  sont telles que  $\Lambda' = \gamma \circ \Lambda$  avec  $\gamma$ , une généralisation fondée sur  $CD$  qui est applicable à  $\Lambda(Q)$  ( $\gamma$  est soit de la forme  $a \rightsquigarrow b$ , soit de la forme  $\neg a \rightsquigarrow \top$ ).

On peut souligner que l'espace de recherche contient au moins un état final : en particulier,  $(\Sigma, \Lambda)$  avec  $\Sigma =$  fonction identité et  $\Lambda = \Gamma$  satisfait (7), puisque (3) est satisfaite ( $R$  étant un cas mémorisé avec une requête généralisée  $\Gamma(Q)$ ), ainsi l'algorithme termine. De plus, ceci montre que l'adaptation à base de règles équivaut à l'adaptation issue de la mémorisation quand il n'y a aucune règle d'adaptation ( $CA = \emptyset$ ).

Par exemple, avec  $idx(R)$  et  $Q$  définies par (1) et (2), avec les connaissances du domaine et les coûts de la figure 5, et avec  $CA = \{\sigma\}$ ,  $\sigma$  étant l'adaptation de (6) et  $\text{coût}(\sigma) = 0,2$ , l'adaptation produit  $\Sigma = \sigma$ ,  $\Lambda = \text{scarole} \rightsquigarrow \text{saladeVerte}$  avec un coût de  $0,2 + 0,8 = 1$  qui implique une adaptation de la recette utilisant la substitution

`laitue  $\rightsquigarrow$  scarole  $\circ$  vinaigre  $\rightsquigarrow$  jusDeCitron  $\wedge$  sel`

**Prise en compte de connaissances d’adaptation spécifiques.** Pour certaines recettes  $R$ , il existe des règles d’adaptation spécifiques, applicables dans le contexte de ces recettes ( $C = R$ ). L’ensemble des règles d’adaptation spécifiques associées à  $R$  est noté  $CA_R$ . Certaines règles d’adaptation proviennent des variantes qui existent dans le texte original des recettes ; si beurre est un ingrédient de  $R$  et qu’il est indiqué qu’il peut être remplacé par de la margarine, alors  $\sigma = \text{beurre} \rightsquigarrow \text{margarine} \in CA_R$  et, comme cette variante est indiquée dans la recette, le coût de cette règle est de 0 (pas de risque lors de l’adaptation).

La prise en compte de règles d’adaptation spécifiques consiste simplement à considérer  $CA_R$ , où  $R$  est la recette remémorée, en supplément des règles d’adaptation de  $CA$  : l’adaptation de  $R$  est fondée sur  $CA \cup CA_R$ .

### 4.3. Adaptation des quantités

La substitution d’ingrédients par d’autres ingrédients nécessite de gérer l’adaptation des quantités, pour conserver un certain équilibre dans la recette adaptée. Ainsi, lors d’une substitution de la bière par du champagne (liquide plus alcoolisé et plus sucré que la bière) dans un cocktail, il est intéressant de conserver le taux d’alcool, le taux de sucre, ainsi que le volume initial du cocktail. Un résultat concret est visible en figure 3, dans lequel l’équivalent en poids d’une mangue est requise en figues, la réduction de la quantité de graines de sésame s’expliquant par la préservation des composants nutritionnels (fibres, calcium, magnésium, ...). TAAABLE utilise pour cela un formalisme de conjonctions de contraintes linéaires. L’adaptation des quantités prend en entrée :

- la recette à adapter avec la liste de ses ingrédients avec leurs quantités et unités ;
- un ensemble de substitutions ;
- des données numériques sur les ingrédients : des conversions d’unités (e.g. le poids d’une mangue) et des données nutritionnelles (e.g. nombre de calories dans un gramme de mangue).

Elle suit l’approche d’adaptation par révision (Cojan, Lieber, 2009) appliquée à un formalisme attribut-valeur. Les cas sont formalisés comme un problème de satisfaction de contraintes sur des attributs.

**Espace de représentation.** Seuls des attributs numériques sont considérés. Les quantités d’ingrédients en gramme et dans une autre unité si cette unité existe dans la recette. Les valeurs des ingrédients sont stockées dans des variables  $[\text{ingrédient}]_{[\text{unité}]}$ . Par exemple, dans l’exemple introductif qui consistait à remplacer des mangues par des figues dans la recette *Glutinous rice with mangoes*, comme la quantité de mangues est donnée en nombre, deux variables sont utilisées pour les mangues :  $\text{mangue}_{\text{unité}}$  et  $\text{mangue}_g$ . En supplément des ingrédients expressément mentionnés dans la recette et ceux provenant des substitutions, leurs généralisations sont également considérées. Par exemple, comme *figue* est une sous-classe de *fruitTropical* et que *fruitTropical* est une sous-classe de *fruit* il y aura

Weight conversions		Nutritional values	
1 c, sliced	165 g	<b>Nutritional value per 100 g (3.5 oz)</b>	
1 unit	207 g	<b>Energy</b>	65 kcal (270 kJ) 65 (kcal)
		<b>Carbohydrates</b>	17 g
		<b>Sugars</b>	14.8 g
		<b>Dietary fiber</b>	1.8 g
		<b>Fat</b>	0.27 g
		<b>Protein</b>	0.51 g
		<b>Water</b>	81.71 g
		<b>Vitamin A (equiv.)</b>	38 µg (4%)
		<b>Thiamine (Vit. B1)</b>	0.058 mg (4%)
		<b>Riboflavin (Vit. B2)</b>	0.057 mg (4%)
		<b>Niacin (Vit. B3)</b>	0.584 mg (4%)
		<b>Pantothenic acid (Vit. B5)</b>	0.16 mg (3%)
		<b>Vitamin B6</b>	0.134 mg (10%)
		<b>Folate (Vit. B9)</b>	14 µg (4%)
		<b>Vitamin C</b>	27.7 mg (46%)
		<b>Calcium</b>	10 mg (1%)
		<b>Iron</b>	0.13 mg (1%)
		<b>Magnesium</b>	9 mg (2%)
		<b>Phosphorus</b>	11 mg (2%)
		<b>Potassium</b>	156 mg (3%)
		<b>Sodium</b>	2 mg (0%)
		<b>Zinc</b>	0.04 mg (0%)
		Percentages are relative to US <a href="#">recommendations</a> for adults.	
		Source: <a href="#">USDA Nutrient database</a>	
		Corresponding ingredient: MANGOS.RAW	

Figure 6. Données de masse et données nutritionnelles de la mangue

une dimension pour chacune de ces classes dans l'espace de représentation. Toutes les données nutritionnelles qui apparaissent dans la page wiki de l'aliment (cf. figure 6) sont représentées par les variables  $lipid_g$ ,  $energy_{Kcal}$ , etc. Les aliments qui ne sont pas associés à des données nutritionnelles ne sont pas pris en compte dans l'adaptation de quantités.

**Connaissances du domaine.** Les connaissances du domaine sont formalisées comme un ensemble de contraintes linéaires :

- Conversion d'unité, par exemple :  $mangue_g = 165 mangue_{tasses}$  (une tasse de mangue contient 165 grammes de mangue).
- Hiérarchie des aliments, par exemple :  $fruitTropical_g = figue_g + mangue_g$  (car dans cet exemple d'adaptation, aucun autre fruit tropical n'intervient).
- Équations sur les données nutritionnelles, par exemple :  $energy_{Kcal} = 0,65 mangue_g + 0,74 figue_g + 0,97 glutinous\_rice_g + \dots$

**Cas source.** La solution d'un cas (i.e. les quantités d'ingrédients) est aussi exprimée comme un ensemble de contraintes linéaires. Par exemple, dans la recette source "Glutinous rice with mangoes" :  $riz_{tasses} = 3$ .

**Cas cible.** Pour chaque substitution  $de\_ing \rightsquigarrow par\_ing$ , une contrainte  $de\_ing = 0$  est ajoutée au cas cible. Cette transcription des substitutions provient du fait que la substitution a été obtenue par un chemin de généralisation-spécialisation de la forme  $de\_ing \overset{\text{généralisation}}{\rightsquigarrow} parent \overset{\text{spécialisation}}{\rightsquigarrow} par\_ing$ . Les connaissances du domaine contiennent par conséquent la contrainte :  $parent_g = de\_ing_g + par\_ing_g$ . Sachant que  $de\_ing_g = 0$  dans le cas cible, tout en préservant  $parent_g$  entraînera la substitution de  $de\_ing$  par  $par\_ing$ .

**Distance.** Le coût d'adaptation est évalué par une distance  $dist$  entre le cas source et la solution proposée pour la cible. Cette distance est une somme pondérée de chaque changement de valeurs (i.e.  $dist(x, y) = \sum_i w_i |y_i - x_i|$  où  $i$  représente une des dimensions). Toutes les dimensions de l'espace de représentation sont prises en compte (quantités, données nutritionnelles, ...). Cela entraîne l'ajustement de certaines quantités qui ne sont pas explicitement impliquées dans la substitution, cf. par exemple comment la quantité de graines de sésame est modifiée en figure 3.

**Calcul.** L'adaptation effectuée s'appuie sur un opérateur de révision des croyances (Alchourrón *et al.*, 1985) dans le formalisme des conjonctions de contraintes linéaires. Pour plus de détails, nous renvoyons à (Cojan, Lieber, 2014).

#### 4.4. Adaptation de la préparation d'une recette et du texte associé

Le remplacement d'un ingrédient  $a$  par un ingrédient  $b$  nécessite d'adapter la partie préparation d'une recette, les ingrédients  $a$  et  $b$  requérant parfois des modes de préparation différents. Le texte de la préparation est adapté en remplaçant la séquence d'actions appliquées à  $a$  par une séquence similaire d'actions appliquées à  $b$  dans une autre recette. L'utilisation d'une préparation de  $b$  qui existe dans une autre recette limite le risque que  $b$  ne soit pas préparé correctement. Chaque séquence d'actions différente appliquée à  $b$  qu'on peut retrouver dans l'ensemble des recettes est appelée « prototype » de  $b$  et définit une des manières de préparer  $b$ .

Considérons la requête présentée en figure 2. Aucun dessert avec des figues et du riz n'existe dans la base de cas utilisée par TAAABLE. Cependant, il existe une recette intitulée « Glutinous rice with mangoes », qui est retournée par TAAABLE avec la substitution de mangue par figue. Dans cette recette, le prototype de la mangue contient les actions `refroidir`, `peler`, `trancher`, `enlever-le-noyau`, `mettre`. Si la substitution de mangue par figue semble être une adaptation d'ingrédient satisfai-

sante, peler ou dénoyauter les figues n'a pas de sens. Dans la base de cas, 2 recettes utilisent des figues. La recette numéro 1163 dont le prototype associé à figue contient les actions *couper-en-deux*, *saupoudrer*, *déposer-une-noisette-de*, *cuire*, *dorer*, *mettre* et la recette numéro 53 dont le prototype contient uniquement *couper*, *mélanger*. La hiérarchie des actions est utilisée pour calculer les similarités entre actions, permettant de considérer des classes d'actions similaires, e.g. *couper* et *couper-en-deux*, *cuire* et *dorer*, etc.

La sélection du prototype de figue le plus approprié utilise l'analyse formelle de concepts (Ganter, Wille, 1999). Un treillis de concepts est construit sur les prototypes des figues et sur celui de la mangue. Le concept relatif à une préparation de figue qui est le plus similaire au concept relatif à la préparation de la mangue constitue le mode de préparation le plus approprié. Dans l'exemple, *figue\_#53* est sélectionné car l'adaptation de séquences ne requiert que la suppression de trois actions, tandis qu'avec *figue\_#1163*, quatre actions devraient être ajoutées en supplément. Nous renvoyons à (Dufour-Lussier *et al.*, 2010) pour des explications détaillées sur la recherche de la préparation la plus similaire.

Une fois que le prototype de figue le plus approprié a été trouvé, l'adaptation textuelle consiste à remplacer les portions de texte relatifs à mangue dans la recette mémorisée par les portions de textes relatifs aux figues dans la recette #53 :

[...] Blend the sauce ingredients in a pot and heat until it just reaches the boiling point. Let cool. ~~Peel the mangoes, slice lengthwise and remove the pits.~~ Cut figs into wedges. Divide the rice mixture among 6 plates.  
[...]

#### 4.5. Note sur la complexité

Les inférences de TAAABLE demandant le plus du temps de calcul sont des recherches dans les espaces d'états (typiquement  $A^*$ ) qui sont de complexité exponentielles dans le pire cas. Cependant, ces opérations portent sur des objets de « petite taille » en général. À titre d'exemple, la mémorisation porte essentiellement sur une requête dont la taille n'excède pas cinq termes (du moins si on s'appuie sur les requêtes qui ont été posées lors des différents concours). Pour une telle taille de requête, le temps de mémorisation est généralement inférieur à la seconde sur un ordinateur actuel. En revanche, une requête avec plus de dix termes prendra parfois un temps de calcul rédhibitoire, surtout quand il n'existe aucune recette proche de la requête.

## 5. Gestion des conteneurs de connaissances de TAAABLE

### 5.1. Une ontologie du domaine culinaire

L'ontologie culinaire de TAAABLE  $\mathcal{O}$  définit les principales classes et relations du domaine culinaire.  $\mathcal{O}$  est composée de 6 hiérarchies :

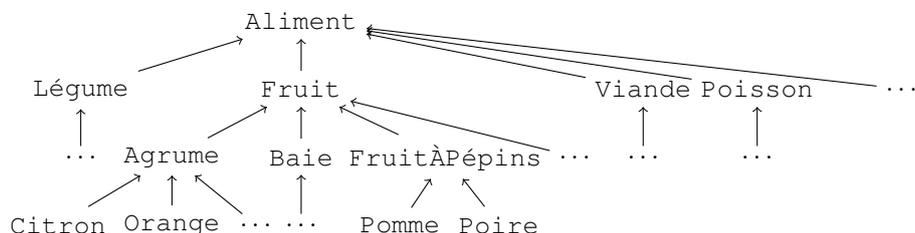


Figure 7. Un extrait de la hiérarchie des aliments

– une hiérarchie *aliment*, contenant les ingrédients des recettes, e.g. Légume, Fruit, Baie, Viande, etc. (un extrait de cette hiérarchie est donné en figure 7).

– une hiérarchie *type de plat*, contenant les différents types de plats associés aux recettes, e.g. Tarte, Salade, Soupe, Cocktail, etc.

– une hiérarchie *rôle de plat*, contenant les moments auxquels les plats associés aux recettes peuvent être mangés, e.g. Snack, Entrée, Plat Principal, Dessert, etc.

– une hiérarchie *origine*, contenant les origines des recettes, e.g. France, Asie, Méditerranée, etc.

– une hiérarchie *régime*, relative aux aliments autorisés ou interdits pour un régime spécifique, e.g. Végétarien, Sans Noix, etc.

– une hiérarchie *action*, relative aux actions culinaires requises pour la préparation des ingrédients, e.g. Couper, Peler, etc.

Étant donné deux concepts A et B de cette ontologie, A est plus spécifique que B, noté « $A \sqsubseteq B$ », si et seulement si l'ensemble des instances de A est inclus dans l'ensemble des instances de B. Par exemple, «Agrume (qui désigne les agrumes) est plus spécifique que Fruit (qui désigne les fruits)» signifie que tous les agrumes sont des fruits.

L'ontologie  $\mathcal{O}$  a été créée pour servir de support aux processus de remémoration et d'adaptation du système TAAABLE. Ainsi, les choix conceptuels pour le développement de cette ontologie ont fortement été guidés par la finalité de TAAABLE. La réutilisation d'ontologies existantes a attentivement été étudiée mais abandonnée car ces ontologies ne répondaient pas aux objectifs fixés par le projet. Ainsi, après identification des classes majeures issues de l'élaboration du modèle conceptuel du domaine de la cuisine, une structuration détaillée des classes a été réalisée, à partir de ressources terminologiques existantes et d'une expertise manuelle.

**La hiérarchie *aliment*.** Une première version de la hiérarchie des aliments a été construite manuellement, à partir de plusieurs ressources accessibles sur le web, telles que le «*Cook's Thesaurus*» (<http://www.foodsubs.com>), une encyclopédie de la cui-

sine qui contient plusieurs milliers d'ingrédients, et Wikipedia. La principale tâche était de sélectionner les classes pertinentes pour l'application et de les organiser selon la relation  $\sqsubseteq$ . De façon concomitante, une première version d'une base terminologique a été construite dans le but d'associer à chaque classe (e.g., BokChoy) une forme linguistique préférée (e.g., *bok choy*) ainsi qu'un ensemble de variantes lexicales qui peuvent être des variantes morphologiques ou des synonymes (e.g., *pak choy*, *pak choï*, *chinese cabbage*, etc.).

Cette version initiale de la hiérarchie des aliments (et de la base terminologique associée) a été ensuite enrichie itérativement en ajoutant de nouvelles classes (et/ou de nouveaux termes) qui apparaissaient dans le livre de recettes mais étaient absents de la hiérarchie. Pour cela, un processus semi-automatique a été développé. Ce processus retourne les lignes d'ingrédients du livre de recettes pour lesquels aucun rattachement à une classe de la hiérarchie des aliments n'a été trouvé par le processus d'annotation. Une expertise manuelle est ici requise pour déterminer la raison de cet échec et faire les rectifications nécessaires. Trois cas ont été identifiés :

- la classe représentant un aliment est manquante dans la hiérarchie : l'expert doit alors la créer et la placer dans la hiérarchie en la rattachant aux classes les plus spécifiques qui la subsument ;
- la classe relative à l'aliment existe dans la hiérarchie mais la forme lexicale qui apparaît dans la recette n'est pas dans la base terminologique : l'expert doit alors ajouter cette nouvelle forme lexicale dans l'ensemble des variantes lexicales associées à la classe ;
- la ligne d'ingrédient contient une erreur (e.g. aucun aliment n'y est mentionné, un ingrédient est mal orthographié, etc.) : l'expert doit alors corriger la ligne d'ingrédients ou supprimer cette ligne dans la recette.

Cette approche a été intégrée au wiki : quand une nouvelle recette est ajoutée, le processus d'annotation est déclenché et donne à l'utilisateur un retour immédiat en cas de problème de telle sorte que ce dernier puisse corriger et compléter les connaissances stockées.

Enfin, certaines informations et connaissances sur les aliments sont automatiquement collectées en utilisant *Freebase* (<http://www.freebase.com>), une base de données RDF. Freebase est interrogée pour collecter une courte description de l'aliment, un lien vers sa page dans Wikipedia, ses variantes lexicales dans différentes langues, sa compatibilité/incompatibilité avec certains régimes, et d'éventuelles illustrations en images.

La hiérarchie des aliments est la plus grande des hiérarchies utilisée par TAAABLE. La version actuelle contient environ 3000 classes, organisées sur 9 niveaux. La base de données terminologiques associées aux aliments contient environ 4200 formes lexicales anglaises, sans comptabiliser les variantes « singulier/pluriel ».

**Hiérarchies des types de plat et des origines.** En partant de l'organisation des types et origines des plats sur le site *Recipe Source* (<http://www.recipesource.com>), une liste de types de plats et une liste d'origines ont été établies et ont été organisées hiérarchiquement selon la relation de subsomption pour la première et la relation de composition pour la seconde.

La hiérarchie des origines contient 41 classes organisées en 2 niveaux. Le premier niveau organise les origines par rapport à des grandes régions, telles que Afrique, Asie, Europe, etc. Chaque classe du premier niveau se spécialise, au second niveau, en pays de provenance des plats. Par exemple, France, Allemagne, Espagne, etc. sont des sous-parties d'Europe.

La hiérarchie des types de plat, contenant 69 classes, est organisée en 3 niveaux. Au premier niveau, il y a des classes comme PlatAuFour, Burger, Dessert, PlatPrincipal, etc. Les deuxième et troisième niveaux introduisent, si nécessaire, des classes plus spécialisées. Par exemple, PlatAuFour se spécialise en Biscuit, Pain, Muffin, Cookie, etc. Cependant, ces classes ne sont pas davantage détaillées, même si des catégories plus spécifiques existent dans *Recipe Source*.

**Hiérarchie des rôles de plat et des régimes.** Ces deux hiérarchies possèdent seulement un niveau, chaque classe étant directement attachée à la racine (c'est-à-dire la classe la plus générale) de la hiérarchie. Il y a actuellement 8 rôles de plats (e. g., Snack, Entrée, etc.) et 7 types de régime (e.g. Végétarien, Régime sans arachide, etc.) dans  $\mathcal{O}$ .

**La hiérarchie des actions.** La hiérarchie des actions a été construite en organisant une liste de verbes automatiquement extraits des textes décrivant les préparations des recettes. La hiérarchie des actions contient 449 classes organisées en 5 niveaux. Chaque classe est décrite par des propriétés syntaxiques et sémantiques, qui permettent au processus d'acquisition de cas automatique décrit ci-après de fonctionner. Par exemple, Couper est une sous-classe de Trancher et est décrit par la propriété linguistique `estUnVerbeTransitifDirect`. Cette propriété implique qu'un complément d'objet direct doit être recherché dans le texte par le processus d'annotation de la préparation. Séparer est également décrit par la propriété fonctionnelle `estUnVerbeUnion`, ce qui signifie que cette action produit plusieurs sorties.

**Données nutritionnelles et de masse.** En utilisant l'*USDA Nutrient database* (<http://www.nal.usda.gov>), les classes d'aliments ont pu être liées à leurs données nutritionnelles (sucre, lipide, protéine, vitamines, etc.) et à leur masse moyenne, deux types de connaissances indispensables à l'adaptation des quantités (cf. section 4.3). La figure 6 donne un exemple de données nutritionnelles relatives à la mangue. Pour lier les données nutritionnelles et de masse aux classes de l'ontologie, un processus semi-automatique a été mis en œuvre pour aligner les aliments dans la base de données de

l’USDA avec ceux de l’ontologie. Comme la mise en correspondance est incomplète, certains aliments de l’ontologie ne possèdent pas ces types d’information.

### 5.2. *Connaissances d’adaptation*

TAAABLE utilise une forme particulière de connaissance d’adaptation (CA), à savoir des règles de substitution de certains ingrédients par d’autres (e.g. dans la recette “*My Strawberry Pie*”, *Fraise* peut être remplacé par *Framboise*). Formellement, en accord avec ce qui est décrit en section 4.2, une CA est un triplet (*contexte, substituants, substitués*), où :

- *contexte* représente la recette ou une classe de recettes sur laquelle la substitution peut s’appliquer. Une CA est spécifique si *contexte* est une recette particulière et générique si *contexte* est une classe de recettes (e.g. un type de plat particulier).
- *substituants* et *substitués* sont respectivement l’ensemble des ingrédients qui doivent être remplacés et l’ensemble des ingrédients qui les remplacent.

Suivant la définition précédente, (“*My Strawberry Pie*”, *Fraise*, *Framboise*), est une CA dont l’interprétation est que les fraises peuvent être remplacées par des framboises dans la recette “*My Strawberry Pie*”. Dans WIKITAAABLE, chaque substitution est encodée comme une page de wiki. Une requête sémantique est utilisée pour alimenter automatiquement la section *Substitutions* d’une page de recette, comme cela peut être vu en figure 8.

### 5.3. WIKITAAABLE : un wiki sémantique pour TAAABLE

WIKITAAABLE est un wiki sémantique utilisé pour représenter, éditer et maintenir les connaissances utilisées par TAAABLE (Cordier *et al.*, 2009). Les figures 8 et 9 illustrent respectivement une page de recette et une page d’une classe de l’ontologie accessible sur <http://wikitaaable.loria.fr/>. Le développement de WIKITAAABLE a été réalisé à partir de Semantic MediaWiki (Krötzsch *et al.*, 2007) qui est une extension de MediaWiki (un moteur de wiki utilisé notamment par Wikipedia) permettant d’ajouter de la sémantique aux pages wiki. L’utilisation de MediaWiki a permis de bénéficier des facilités d’édition collaborative et à distance que fournit le wiki.

Avec WIKITAAABLE, les utilisateurs peuvent consulter, interroger, et éditer la base de connaissances à travers une interface dédiée. Le moteur de RÀPC est connecté à WIKITAAABLE via le RDF store (base de données contenant les triplets RDF) associé à Semantic MediaWiki, dans lequel sont stockées les unités de connaissances ; il peut donc utiliser ces connaissances pour raisonner. WIKITAAABLE contient également des *bots* (de petits programmes) qui réalisent des tâches automatisées telle que l’annotation sémantique des pages de recettes. Enfin, des interfaces additionnelles spécifiques ont été ajoutées à WIKITAAABLE afin que les utilisateurs n’aient qu’un seul outil pour gérer les connaissances. En résumé, WIKITAAABLE fournit un environnement dans lequel des humains et des agents peuvent collaborer.

[page](#)
[discussion](#)
[edit](#)
[history](#)
[delete](#)
[move](#)
[protect](#)
[watch](#)
[refresh](#)

## My strawberry pie

---

### Ingredients [edit]

- 1/2 c Water
- 1 1/2 c Sugar
- 1 Pie crust (9 inch), baked
- 1 1/2 qt Strawberries
- 1/3 c Cornstarch
- 3 c Cool whip

### Preparation [edit]

- Chop 2 cups of berries. In saucepan combine sugar and cornstarch. Slowly add water to combine smoothly. Add chopped strawberries. Cook, stirring constantly until mixture thickens and boils. Cool in refrigerator for about 1/2 hour. Pour about 3/4 of mixture into prepared pie crust. Stand up whole strawberries in syrup (to fill crust). Pour remaining syrup over strawberries. Chill until firm (about 3 hours). Spread cool whip over top of pie and serve.

### Substitutions [edit]

	Context	Replace	By	Origin
Substitution 1 in My strawberry pie	My strawberry pie	Strawberry	Raspberry	Taaable

Figure 8. Exemple d'une recette de WIKITAAABLE : « My strawberry pie »

**Annotation des ingrédients.** Le moteur de raisonnement de TAAABLE requiert une représentation formelle des recettes. Le processus d'annotation vise à encoder formellement, en exploitant les classes et propriétés stockées dans le wiki sémantique, le contenu d'une recette, ainsi que des méta-propriétés comme, par exemple, son origine et le type de plat que la recette va produire. Ce processus se situe entre de l'*indexation contrôlée* (Jo *et al.*, 2008) où les termes proviennent d'une terminologie prédéfinie et de l'*annotation sémantique* (Uren *et al.*, 2006) où les termes (entités nommées, séquences de mots) sont explicitement associées à des classes dans l'ontologie. Le résultat de l'annotation de la partie « ingrédient » d'une recette est un ensemble de classes d'aliments liées chacune à une recette avec la propriété sémantique du wiki `aPourIngrédient`. Pour cela, la liste des ingrédients est analysée. Chaque ligne d'ingrédient dans la recette est analysée et divisée en un quadruplet (quantité, unité, aliment, modificateurs). La base terminologique guide le processus d'analyse : les variantes lexicales sont utilisées conjointement à des expressions régulières pour rechercher dans le texte, une instance d'une classe de  $\mathcal{O}$ . Par exemple, comme "pak choy" est une forme lexicale associée à la classe d'aliment `BokChoy` dans la hiérarchie des aliments, l'entrée « 1 kg sliced pak choy » est analysée comme (1, kg, BokChoi, sliced).

La relation `aPourIngrédient` permet d'encoder l'index d'une recette. Par exemple, la recette intitulée « My strawberry pie », illustrée en figure 8, est indexée par la conjonction des ingrédients `Water`, `Sugar`, `PiePastry`, `Strawberry`, `Cornstarch`, and `CoolWhip`.

page discussion view source history

## Category:Berry

### Description

The botanical definition of a berry is a fleshy fruit produced from a single ovary.  
 Read the whole article on [Wikipedia](#)



### Lexical variants

- English: berry
- Français: baie
- Deutsch: Beere
- Español: Baya

### Recipes using Berry

- Cran-raspberry relish
- Spicy cranberry chutney

### Subcategories

<b>B</b>	<b>C cont.</b>	<b>K</b>
<ul style="list-style-type: none"> <li>[+] Baby kiwifruit (0)</li> <li>[+] Blackberry (0)</li> <li>[+] Blueberry (0)</li> </ul>	<ul style="list-style-type: none"> <li>[+] Currant (0)</li> </ul>	<ul style="list-style-type: none"> <li>[+] Kiwi fruit (0)</li> </ul>
<b>C</b>	<b>F</b>	<b>R</b>
<ul style="list-style-type: none"> <li>[+] Cranberry (0)</li> </ul>	<ul style="list-style-type: none"> <li>[+] Fraise des bois (0)</li> </ul>	<ul style="list-style-type: none"> <li>[+] Raspberry (0)</li> </ul>
	<b>G</b>	<b>S</b>
	<ul style="list-style-type: none"> <li>[+] Grape (3)</li> </ul>	<ul style="list-style-type: none"> <li>[+] Strawberry (0)</li> </ul>

Category: Fruit

Figure 9. Un exemple de page wiki d'une classe de l'ontologie : la classe *Baie*

**Annotation des types et origines des plats.** Un autre processus d'annotation indexe les recettes par leur origine (e.g., *Asie*) et le type de plat produit (e.g., *PlatPrincipal*, *Dessert*). Comme aucune indication sur l'origine ou le type de plat ne figure dans le livre de recette, *Recipe Source* est à nouveau utilisé, cette fois pour construire un corpus dans lequel les recettes sont associées à leurs origine et type(s). Un processus en 3 étapes a été mis en œuvre pour déterminer l'origine et le type d'une recette :

- s'il existe une recette dans *Recipe Source* avec le même titre, alors l'origine et le(s) type(s) de plat de la recette dans *Recipe Source* sont assignées à la recette ;
- si le titre de la recette (e.g., « *Chinese Vegetable Soup* ») contient des mots-clés associés à des classes de la hiérarchie des types de plats (e.g., « *Soup* ») ou de la hiérarchie des origines (e.g., « *Chinese* ») alors ces origine et type(s) de plat sont assignés à la recette ;

– des règles d’association ont également été extraites du corpus de recettes issu de Recipe Source, en utilisant CORON (Szathmary, Napoli, 2005), une plate-forme de fouille de données. À partir des règles d’association extraites de la forme <ensemble d’ingrédients>  $\rightarrow$  <origine ou type de plat> (e.g., vanilla bean, banana, chocolate  $\rightarrow$  dessert) et ayant une confiance de 100%, l’origine et/ou le(s) type(s) de plat de la partie droite de la règle est/sont assignée(s) à la recette si celle-ci est indexée par, au minimum, tous les ingrédients présents dans la partie gauche de la règle.

Par exemple, la recette intitulée « My strawberry pie » est indexée par ses ingrédients, mais aussi par les types de plats *Dessert* et *Tarte*. La représentation sémantique d’une recette étant stockée dans un wiki, les utilisateurs peuvent corriger et/ou compléter l’annotation des recettes.

**Annotation des préparations.** La mise en œuvre du processus d’adaptation textuelle décrit en section 4.4 requiert une représentation formelle de la préparation des ingrédients. Une représentation sous forme d’arbre, comme illustrée en figure 10, a été choisie, pour représenter les combinaisons d’ingrédients jusqu’à obtenir le plat final. Chaque nœud représente une composition d’aliment (un ingrédient ou un mélange d’ingrédients) à un instant donné. Chaque arc  $\langle a, b \rangle$  représente une action à appliquer sur la composition d’aliment  $a$  pour conduire à une nouvelle composition  $b$ . Un ensemble d’arcs arrivant sur un même nœud représente une action qui combine des aliments pour produire une nouvelle composition.

Une telle représentation est construite de façon itérative, en suivant l’ordre des verbes (actions) dans le texte de la préparation. Cependant, certaines phrases sont difficiles à analyser ; la représentation formelle partielle déjà construite peut alors aider à cette analyse. Par exemple, dans la phrase « Peel the mangoes, slice lengthwise and remove the pits », il est aisé pour un humain de comprendre que les mangues sont tranchées et dénoyautées, mais cela nécessite l’utilisation de certaines heuristiques pour

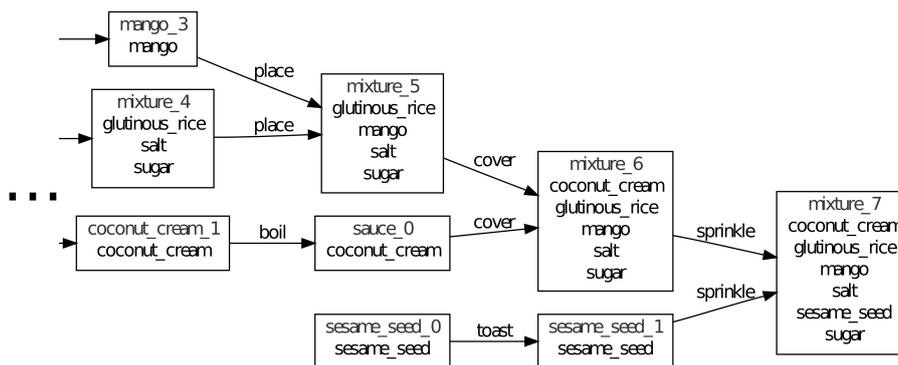


Figure 10. Extrait de la représentation formelle de la recette intitulée « Glutinous rice with mangoes »

être interprété par une machine. Chaque action est associée à un nombre d'arguments, ce qui permet de détecter l'absence éventuelle d'un argument. Lorsqu'un argument est manquant, nous faisons l'hypothèse que l'argument manquant correspond au dernier nœud ajouté à l'arbre. Cette approche permet de gérer les anaphores (i.e. lorsqu'un mot différent, tel qu'un pronom, ou lorsqu'il n'y a pas de mot du tout, la résolution de l'anaphore permet de référer à l'objet).

D'autres types d'anaphore apparaissent également, comme par exemple, l'expression « épices » qui réfère à un certain ensemble d'aliments. La hiérarchie des aliments est utilisée pour retrouver tous les nœuds de l'arbre qui font référence aux aliments qui sont des « épices ». Une phrase telle que « cover with sauce » est plus délicate à analyser car il n'y a pas d'indices flagrants dans le texte ou dans l'ontologie sur ce que « sauce » peut être. Pour ces cas, nous avons construit à partir de l'analyse de milliers de recettes, des ensembles « typiques » d'aliments qui apparaissent dans des compositions d'ingrédients identifiées par le mot « sauce », ou encore « pâte ». Ainsi, comme de nombreuses recettes font référence par exemple à une sauce avec de la crème de noix de coco (et aucune sauce avec du riz, par exemple), nous pouvons faire l'hypothèse que si une composition d'aliments dans une recette donnée contient de la crème de noix de coco, il s'agit de la composition d'aliments auquel le mot « sauce » réfère.

Avec la représentation arborescente, il est facile d'identifier la séquence d'actions appliquées à un ingrédient donné, dans une recette, pour pouvoir le remplacer tel que décrit en section 4.4. Comme le prototype d'un ingrédient est défini comme la séquence d'actions qui lui sont appliquées, jusqu'à la première action de combinaison incluse, cela revient en pratique à considérer que seule la partie la plus à gauche de l'arbre est à remplacer ou à utiliser comme remplaçant.

#### 5.4. *Extraction de connaissances d'adaptation*

Deux systèmes d'extraction de connaissances, nommés AK EXTRACTOR et GENERIC AK EXTRACTOR ont été ajoutés au wiki pour acquérir respectivement des CA spécifiques (i.e. applicables à une recette particulière) et génériques (i.e. applicables sur un ensemble de recettes). Ces deux systèmes sont déclenchés à partir d'une requête de l'utilisateur, qui va avoir en charge de valider les résultats du processus d'extraction de connaissances, en y apportant éventuellement des corrections.

AK EXTRACTOR tient compte de la compatibilité des ingrédients, palliant cette faiblesse du processus d'adaptation de TAAABLE. AK EXTRACTOR exploite les différences d'ingrédients qui existent entre la recette à adapter et un ensemble de recettes similaires. AK EXTRACTOR sélectionne tout d'abord des recettes qui ont un nombre minimal d'ingrédients en commun et un nombre minimal d'ingrédients différents avec la recette à adapter. Des motifs fermés (Ganter, Wille, 1999) sont alors extraits des variations d'ingrédients entre la recette à adapter et chaque recette issue de l'étape précédente. Les motifs fermés sont alors filtrés et classés avec des règles particulières (cf. (Gaillard *et al.*, 2011) pour plus de détails). AK EXTRACTOR affiche les propositions de substitutions d'ingrédients en provenance des motifs les mieux

[special page](#)

---

## Special:AKExtractor

---

### Recipe

My strawberry pie

#### Propositions of substitutions according to -strawberry

<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Raspberry	<input checked="" type="checkbox"/> Food color	<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Peach	<input checked="" type="checkbox"/> Lemon juice	<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Lemon rind	<input checked="" type="checkbox"/> Heavy cream	<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Raspberry		<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Raspberry	<input checked="" type="checkbox"/> Gelatin	<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Peach	<input checked="" type="checkbox"/> Food color	<input checked="" type="checkbox"/> Cream cheese	<input type="button" value="OK"/>	<input type="button" value="Not OK"/>
<input checked="" type="checkbox"/> Strawberry	<input checked="" type="checkbox"/> Cool whip	→	<input checked="" type="checkbox"/> Cinnamon		<input type="button" value="OK"/>	<input type="button" value="Not OK"/>	

*Figure 11. Interface de validation du système AK EXTRACTOR pour la requête : adapter la recette intitulée « My Strawberry Pie » sans fraise*

classés. La figure 11 présente l'interface de validation du système AK EXTRACTOR déclenché pour rechercher l'adaptation de la recette intitulée « My Strawberry Pie » dans laquelle on ne voudrait plus de fraises<sup>1</sup>. La première proposition de substitution signifie que Strawberry et CoolWhip peuvent être remplacés par Raspberry et FoodColor. L'utilisateur peut alors valider (partiellement ou complètement) ou non certaines de ces substitutions.

Un autre système, GENERIC AK EXTRACTOR, a été conçu dans le but d'acquérir des CA génériques, applicables à un ensemble de recettes (e.g. les gâteaux), plutôt qu'à une seule recette. GENERIC AK EXTRACTOR repose également sur une approche utilisant des motifs fermés pour extraire des substitutions génériques à partir d'adaptations spécifiques (Gaillard *et al.*, 2012). Via une interface dédiée, les utilisateurs peuvent déclencher l'extraction de motifs fermés sur un ensemble de recettes. Ces motifs sont filtrés et ordonnés avec des règles spécifiques (fondées sur deux mesures utilisées classiquement dans les processus d'extraction de connaissances par motifs fréquents : le support et la stabilité) avant d'être présentés à l'utilisateur. Le système affiche les propositions de substitutions d'ingrédients. L'utilisateur peut valider ou corriger une proposition de CA pour généraliser ou spécialiser les ingrédients substitués ou substituants, changer le sens de la règle ou encore la classe de recettes pour laquelle la règle est valide. Par exemple, en figure 12, le système est interrogé pour acquérir des CA sur les gâteaux ; 10 règles sont proposées (ordonnées par stabilité décroissante avec un support minimal de 4 et une stabilité minimale de 0,5). Avant

1. Ce type d'adaptation, consistant à adapter une recette avec des contraintes additionnelles, constitue l'« adaptation challenge » proposé par le CCC et remporté par TAAABLE en 2010.

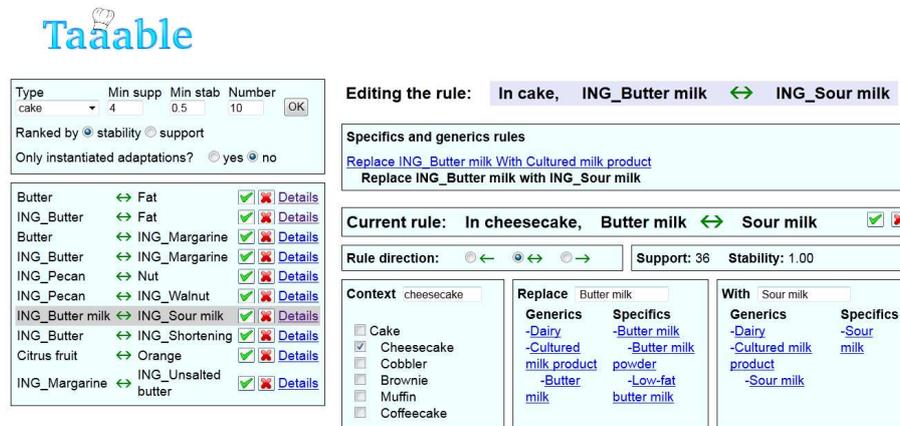


Figure 12. Interface du système GENERIC AK EXTRACTOR sur les gâteaux

de valider une proposition de règle d'adaptation, l'utilisateur peut changer le type de plat sur laquelle la règle va s'appliquer, ainsi que les ingrédients impliqués dans cette règle.

Les règles d'adaptation extraites par ces deux systèmes, une fois validées, sont ajoutées à WIKITAAABLE pour pouvoir être utilisées par TAAABLE pour des adaptations à venir.

## 6. Évaluation et positionnement par rapport aux systèmes concurrents.

De 2008 à 2010, plusieurs systèmes concurrents ont participé au CCC. Même s'ils traitent d'un problème commun, ces systèmes sont très différents dans leur façon d'indexer, de remémorer et d'adapter les cas. Le système "What's in the fridge?" s'est par exemple focalisé sur l'indexation des recettes en utilisant une approche d'*active learning*. La remémoration de cas et l'adaptation utilisent des techniques classiques de recherche d'information et exploitent Wordnet pour calculer la similarité entre la requête et l'ensemble des recettes (Zhang *et al.*, 2008). ColibriCook utilisent également une technique assez simple de recherche d'information : les cas sont remémorés en utilisant une mesure de similarité fondée sur la présence des éléments de la requête (ingrédients, types de plats, etc.) dans les recettes (DeMiguel *et al.*, 2008). D'autres systèmes comme JadaCook (Herrera *et al.*, 2008) ou CookIIS (Hanft *et al.*, 2010) exploitent des hiérarchies pour calculer la similarité. JadaCook utilise une hiérarchie de classes pour calculer la similarité entre la requête et les cas. La remémoration des cas prend en compte la distance entre les ingrédients dans la hiérarchie. CookIIS utilise également une hiérarchie d'ingrédients, cette fois dans le but de calculer la similarité entre ingrédients. Ces similarités entre ingrédients sont alors utilisées pour calculer la similarité entre une requête et les cas pour la remémoration et pour calculer l'adap-

tation. CookIIS traite également l'adaptation du texte de la préparation à un niveau très superficiel, par substitution de chaînes de caractères : une chaîne correspondant à un ingrédient substitué est remplacée par une autre chaîne associée à l'ingrédient substituant). CookingCakeWF (Minor *et al.*, 2010) propose également d'adapter la préparation en considérant les instructions comme un *workflow*. Cependant, ce système n'intègre pas la transformation automatique du texte en un *workflow*, qui reste une tâche manuelle.

L'évaluation intrinsèque d'un système comme TAAABLE ou un de ses concurrents n'est pas aisée. En effet, la qualité d'une recette est subjective et ne s'évalue pas sur une échelle binaire vrai/faux. De plus, les *bonnes* réponses ne sont pas connues *a priori* puisque les systèmes créent de nouvelles réponses par adaptation. Ces deux constats sont justement une des raisons qui ont poussé les membres de la communauté travaillant sur le RÀPC à créer le CCC, pour évaluer les systèmes entre eux. L'évaluation réside en une comparaison des résultats retournés par les systèmes sur quelques requêtes (entre 4 et 6 selon les années) ; elle repose sur différents critères, évalués par un jury majoritairement scientifique, mais impliquant également des cuisiniers professionnels et, en 2015, le public. Un premier critère d'évaluation est la production d'une réponse formelle correcte (i.e. adéquation de la réponse avec la requête). Les fondements logiques sur lesquels TAAABLE a été construit assurent que ce critère est toujours validé. Un deuxième critère d'évaluation utilisé dans le concours est la validité de l'adaptation : la recette est-elle réalisable, et si oui, quelle est sa qualité ? Jusqu'en 2013, l'évaluation de ce critère était effectuée par un cuisinier, puis depuis 2014, des recettes issues des systèmes ont été cuisinées et évaluées par le public qui assistait au concours, après avoir goûté les réalisations. TAAABLE a remporté de nombreux succès : classé 2<sup>ème</sup> en 2008 et 2009, le système remporte en 2010 la 1<sup>ère</sup> place du CCC ainsi que l'« adaptation challenge » (qui consiste à adapter une recette fournie à des contraintes supplémentaires telles que la non disponibilité d'un aliment, par exemple). En 2014, TAAABLE remporte un prix spécial pour l'ensemble des ressources (connaissances et outils génériques) développés et mis à disposition de la communauté. En 2015, TAAABLE remporte le prix du jury (évaluation scientifique de la nouvelle approche proposée pour adapter les quantités) ainsi que le prix du public pour le *cocktail challenge* (consistant à adapter des cocktails en utilisant un ensemble limité d'aliments à disposition et à adapter les quantités) et le prix du public pour le *sandwich challenge* (consistant à adapter des sandwiches avec adaptation des étapes de préparation). Pour ces prix du public, les recettes ont été réalisées et le public, après avoir goûté l'ensemble des cocktails et des sandwiches, devait voter pour son cocktail et son sandwich préférés, sans savoir quel système les avait produits.

## 7. Conclusion et discussion

TAAABLE est un système de RÀPC dont la construction a eu recours à un nombre important de principes, méthodes et techniques du domaine de l'IA tels que la représentation de connaissance, la construction d'ontologies, l'extraction de connaissance à partir de données, l'annotation sémantique, le raisonnement. Une ontologie pour

représenter les connaissances du domaine de la cuisine, ainsi qu’une base terminologique établissant les liens entre textes et concepts de l’ontologie ont été construites. Ces ressources, stockées dans un wiki sémantique, sont utilisées par un processus d’annotation qui permet de construire, à partir de recettes textuelles, une représentation formelle également stockée dans le wiki. L’acquisition de connaissances a également fait appel à des processus d’extraction de connaissances à partir de données, pour enrichir l’ontologie en y ajoutant des concepts afin d’obtenir une structuration plus riche, ou encore pour acquérir des règles d’adaptation. Un processus de RÀPC ainsi que différentes approches pour l’adaptation des ingrédients, des quantités, du texte, qui exploitent ces connaissances, ont été mis en œuvre.

Le travail réalisé depuis 2008 autour de TAAABLE a conduit à la création de deux outils génériques qui sont distribués sous licence libre GPL :

- TUUURBINE (<http://tuuurbine.loria.fr/>), un système de RÀPC générique pour le web sémantique Gaillard *et al.* (2014) qui raisonne sur des connaissances stockées en RDF. TUUURBINE utilise des technologies du web sémantique telles que RDF(S), des RDF store ou SPARQL et implémente un mécanisme d’inférence générique qui consiste à remémorer des cas similaires et à remplacer des caractéristiques de ces cas pour répondre à une requête donnée. La recherche de cas similaires utilise le principe de généralisation/spécialisation à partir de coûts de généralisation et de règles d’adaptation tels que ceux décrits en sections 3, 4.1 et 4.2. La version actuelle de TAAABLE est en fait une instanciation de TUUURBINE qui utilise les connaissances stockées dans le RDF store associé à WIKITAAABLE.

- REVISOR (<http://revisor.loria.fr/>), une bibliothèque d’opérateurs de révision et d’adaptation par révision (Cojan *et al.*, 2013). La version courante de TAAABLE utilise des éléments de REVISOR pour l’adaptation des ingrédients, l’adaptation des quantités et l’adaptation des préparations.

Les ressources (connaissances et logiciels) disponibles à travers le système TAAABLE servent régulièrement de support à des travaux de recherche dont une thèse qui vient d’être soutenue sur le raisonnement exploitant des connaissances provenant d’une communauté en ligne en gérant la fiabilité des connaissances par des interactions sociales (Gaillard *et al.*, 2015).

## Bibliographie

- Alchourrón C. E., Gärdenfors P., Makinson D. (1985). On the logic of theory change: partial meet functions for contraction and revision. *Journal of Symbolic Logic*, vol. 50, p. 510–530.
- Baader F., Hollunder B., Nebel B., Profitlich H.-J. (1992). An Empirical Analysis of Optimization Techniques for Terminological Representation Systems. In *Proceedings of the third International Conference on Principles of Knowledge Representation and Reasoning (KR’92)*, Cambridge, Massachusetts, p. 270–281.
- Cojan J., Dufour-Lussier V., Hermann A., Le Ber F., Lieber J., Nauer E. *et al.* (2013). Révisor : un ensemble de moteurs d’adaptation de cas par révision des croyances. In *JIAF - Septièmes Journées de l’Intelligence Artificielle Fondamentale - 2013*. Aix-en-Provence, France.

- Cojan J., Lieber J. (2009, 10 07). Belief Merging-based Case Combination. In David C. Wilson, Lorraine McGinty (Eds.), *8th International Conference on Case-Based Reasoning - ICCBR 2009 Case-Based Reasoning Research and Development*, vol. 5650, p. 105-119. Seattle United States, Springer Berlin.
- Cojan J., Lieber J. (2014). Applying Belief Revision to Case-Based Reasoning. In *Computational Approaches to Analogical Reasoning: Current Trends*, vol. 548, p. 133 - 161. Springer.
- Cordier A., Lieber J., Molli P., Nauer E., Skaf-Molli H., Toussaint Y. (2009). WIKITAAABLE: A semantic wiki as a blackboard for a textual case-based reasoning system. In *SemWiki 2009 - 4rd Semantic Wiki Workshop at the 6th European Semantic Web Conference - ESWC 2009*. Heraklion, Grèce.
- DeMiguel J., Plaza L., Díaz-Agudo B. (2008). ColibriCook: A CBR System for Ontology-Based Recipe Retrieval and Adaptation. In M. Schaaf (Ed.), *Workshop Proceedings of the 9th European Conference on Case-Based Reasoning*, p. 199–208. Trier.
- Dufour-Lussier V., Lieber J., Nauer E., Toussaint Y. (2010). Text adaptation using formal concept analysis. In I. Bichindaritz, S. Montani (Eds.), *18th International Conference on Case-Based Reasoning - ICCBR 2010*, vol. 6176, p. 96-110. Italy, Springer-Verlag.
- Gaillard E., Infante-Blanco L., Lieber J., Nauer E. (2014). Tuurbine: A Generic CBR Engine over RDFS. In *Case-Based Reasoning Research and Development*, vol. 8765, p. 140 - 154. Cork, Ireland.
- Gaillard E., Lieber J., Nauer E. (2011). Adaptation knowledge discovery for cooking using closed itemset extraction. In *The Eighth International Conference on Concept Lattices and their Applications - CLA 2011*, p. 87–99.
- Gaillard E., Lieber J., Nauer E., Cordier A. (2015, September). How Managing the Knowledge Reliability Improves the Results of a Reasoning Process. In *European conference on knowledge management*. Udine, Italy.
- Gaillard E., Nauer E., Lefevre M., Cordier A. (2012). Extracting Generic Cooking Adaptation Knowledge for the TAAABLE Case-Based Reasoning System. In *Cooking with Computers workshop @ ECAI 2012*. Montpellier, France.
- Ganter B., Wille R. (1999). *Formal Concept Analysis*. Heidelberg, Springer.
- Hanft A., Newo R., Bach K., Ihle N., Althoff K.-D. (2010). Cookiis - a successful recipe advisor and menu advisor. In S. Montani, L. Jain (Eds.), *Successful case-based reasoning applications*, p. 187–222. Springer.
- Herrera P. J., Iglesias P., Romero D., Rubio I., Díaz-Agudo B. (2008). JadaCook: Java Application Developed and Cooked Over Ontological Knowledge. In M. Schaaf (Ed.), *Workshop Proceedings of the 9th European Conference on Case-Based Reasoning*, p. 209–218. Trier.
- Jo T. C., Seo J. H., Hyeon K. (2008). Topic Spotting on News Articles with Topic Repository by Controlled Indexing. In S. B. . Heidelberg (Ed.), *Intelligent Data Engineering and Automated Learning — IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*, vol. 1983, p. 89-99.
- Krötzsch M., Schaffert S., Vrandecic D. (2007). Reasoning in semantic wikis. In G. Antoniou *et al.* (Eds.), *Reasoning web*, vol. 4636, p. 310-329. Springer.

- Lieber J. (2002). Strong, Fuzzy and Smooth Hierarchical Classification for Case-Based Problem Solving. In F. van Harmelen (Ed.), *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02), Lyon, France*, p. 81–85. IOS Press, Amsterdam.
- Minor M., Bergmann R., Görg S., Walter K. (2010). Adaptation of cooking instructions following the workflow paradigm. In C. Marling (Ed.), *ICCBR 2010 workshop proceedings*, p. 199–208.
- Pearl J. (1984). *Heuristics – Intelligent Search Strategies for Computer Problem Solving*. Reading, MA, Addison-Wesley Publishing Co.
- Smyth B., Keane M. T. (1996). Using adaptation knowledge to retrieve and adapt design cases. *Knowledge-Based Systems*, vol. 9, n° 2, p. 127–135.
- Szathmary L., Napoli A. (2005). Coron: A framework for levelwise itemset mining algorithms. In B. Ganter, R. Godin, E. M. Nguifo (Eds.), *Third International Conference on Formal Concept Analysis (ICFCA'05), France, Supplementary Proceedings*, p. 110–113.
- Uren V., Cimiano P., Iria J., Handschuh S., Vargas-Vera M., Motta E. *et al.* (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, n° 1, p. 14-28.
- Zhang Q., Hu R., Namee B. M., Delany S. J. (2008). Back to the Future: Knowledge Light Case Base Cookery. In M. Schaaf (Ed.), *Workshop Proceedings of the 9th European Conference on Case-Based Reasoning*, p. 239–248. Trier.

