



Raisonner sur des connaissances provenant d'une e-communauté

Emmanuelle Gaillard, Jean Lieber, Yannick Naudet, Emmanuel Nauer

► **To cite this version:**

Emmanuelle Gaillard, Jean Lieber, Yannick Naudet, Emmanuel Nauer. Raisonner sur des connaissances provenant d'une e-communauté. IC - 24èmes Journées francophones d'Ingénierie des Connaissances, 2013, Lille, France. 2013. <hal-00918540>

HAL Id: hal-00918540

<https://hal.inria.fr/hal-00918540>

Submitted on 13 Dec 2013

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.

Raisonner sur des connaissances provenant d'une e-communauté

Emmanuelle Gaillard¹²³⁵, Jean Lieber¹²³⁵,
Yannick Naudet⁴⁶ et Emmanuel Nauer¹²³⁵

¹ Université de Lorraine, LORIA — 54506 Vandœuvre-lès-Nancy, France

² CNRS — 54506 Vandœuvre-lès-Nancy, France

³ Inria — 54602 Villers-lès-Nancy, France

⁴ CRP Henri Tudor, Luxembourg

⁵ nom.prénom@loria.fr, ⁶ nom.prénom@tudor.lu

Résumé : Ce papier présente MKM, un modèle de métaconnaissances pour gérer la fiabilité de connaissances en provenance d'une e-communauté. Le but est d'étendre un système de raisonnement à partir de cas (RàPC), de sorte à pouvoir raisonner sur des connaissances partiellement fiables et non expertes, provenant du Web. La fiabilité des connaissances est formalisée grâce au modèle MKM qui utilise les notions de croyance, confiance, réputation et qualité. L'évaluation de la fiabilité est utilisée pour filtrer les connaissances *suffisamment* fiable et pour ordonner les réponses fournies par le système de RàPC, garantissant ainsi une certaine qualité des réponses.

Mots-clés : raisonnement à partir de cas, métaconnaissances, fiabilité, filtrage, classement, personnalisation

1 Introduction

Le Web social génère, au travers des communautés en ligne, une quantité importante de connaissances. La connaissance construite par de telles communautés de personnes ayant les mêmes intérêts et communiquant par Internet, appelées *e-communautés*, a un potentiel d'exploitation. Toutefois, elle est influencée par différents facteurs humains tels que la croyance ou la confiance, lesquels agissent directement sur sa fiabilité.

Dans ce contexte, le travail présenté ici traite de l'exploitation de connaissances partiellement fiables provenant d'e-communautés dans un système de raisonnement à partir de cas (RàPC), en contraste avec l'approche

classique qui exploite des connaissances consensuelles et validées par des experts. Pour préserver une certaine qualité de résultats, un modèle de métaconnaissances, nommé MKM (*Meta-Knowledge Model*), a été conçu pour formaliser les connaissances venant d'utilisateurs (non experts).

La section 2 présente les motivations de cette recherche. Un état de l'art sur l'utilisation des métaconnaissances pour décrire la fiabilité des connaissances, et plus spécialement dans les systèmes de RÀPC, est présenté en section 3. La section 4 détaille le modèle MKM et la section 5 montre comment étendre un système de RÀPC pour manipuler des connaissances partiellement fiables et illustre l'intérêt de MKM sur un exemple avant de proposer une approche pour évaluer MKM.

2 Motivation

Ce travail est motivé par la volonté d'étendre le système TAAABLE [2], un système de RÀPC qui adapte des recettes de cuisine, pour exploiter des connaissances provenant d'une e-communauté. À l'instar des systèmes de RÀPC classiques, TAAABLE, résout un problème à l'aide d'unités d'expériences passées, appelées cas [12]; le moteur de RÀPC retourne des cas de la base de cas, et les adapte pour répondre à une requête. Ce processus utilise plusieurs types de connaissances, notamment : des cas, des connaissances du domaine et des connaissances d'adaptation. De tels systèmes sont traditionnellement utilisés dans des mondes fermés où la base de connaissances est produite par des experts et les connaissances sont consensuelles et validées. Adapter un système de RÀPC, et plus généralement tout système de raisonnement, à l'exploitation de données du Web, nécessite d'intégrer une gestion de la fiabilité des connaissances.

Certains travaux, comme ceux menés dans le projet Kolflow (<http://kolflow.univ-nantes.fr/>), ont étudiés la manière de gérer des connaissances produites par une e-communauté. L'approche Kolflow consiste à améliorer la collaboration homme-machine pour assurer la construction collaborative de connaissances. Cette approche permet de collecter une quantité importante de connaissances éditées par les utilisateurs ; elle utilise des tests de non-régression pour assurer que les nouvelles connaissances ne dégradent pas les résultats du processus de raisonnement qui les exploite [14]. Pour notre part, nous proposons d'associer des métaconnaissances sur la fiabilité à chaque unité de connaissances (UC) provenant de l'e-communauté. Pour cela, nous avons établi un modèle de métaconnaissances qui va permettre de formaliser les connaissances provenant d'une e-communauté et de raisonner dessus.

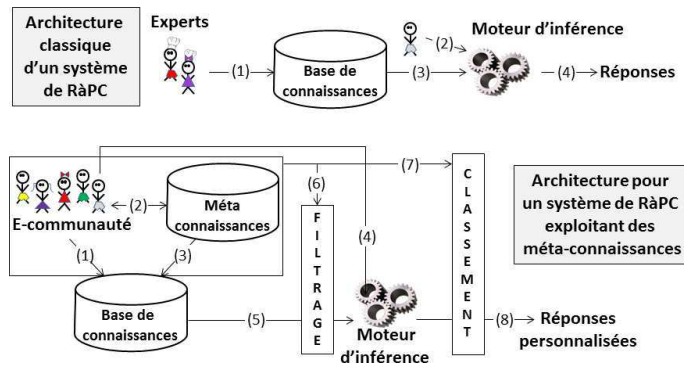


FIGURE 1 – Architecture d'un système de RÀPC classique vs. architecture utilisant les métaconnaissances.

La FIGURE 1 montre la différence d'architecture entre un système de RÀPC *classique* et un système de RÀPC basé sur les métaconnaissances. Dans le système classique, la base de connaissance est produite par les experts du domaine (1). Un utilisateur interroge le moteur RÀPC (2), qui utilise les connaissances validées par les experts (3) pour calculer ses réponses (4). Dans le système qui exploite des métaconnaissances, la base de connaissances est produite par l'e-communauté (1). Des métaconnaissances sont associées aux utilisateurs (2) et aux connaissances (3). Quand un utilisateur interroge le moteur RÀPC (4), ce dernier utilise les connaissances provenant de l'e-communauté (5), filtrées par le modèle de métaconnaissances (6). Les réponses finales du moteur de RÀPC (8) peuvent également être ordonnées (7) grâce à ces métaconnaissances. Par ailleurs, si l'utilisateur qui interroge le système appartient à l'e-communauté, les opérations de filtre et de classement produisent des résultats personnalisés.

3 État de l'art

La fiabilité des connaissances est influencée par plusieurs facteurs interdépendants, comme montré dans [11] où les auteurs proposent un modèle générique pour représenter la connaissance générée par des e-communautés. Ce travail fournit une base pour exploiter les connaissances partiellement fiables dans un processus de raisonnement. Dans le but d'étudier l'impact des différents facteurs qui peuvent servir à évaluer la fiabilité des connaissances, nous nous limitons aux métaconnaissances suivantes : *qualité*, *croyance*, *confiance* et *réputation*.

La qualité est considérée par les chercheurs comme une notion complexe, et est perçue comme un concept multidimensionnel [15]. La na-

ture d'une donnée et la mesure de sa qualité ont un impact important sur le succès d'une prise de décision. Plusieurs critères de qualité sont listés dans [15], comme la consistance, la précision ou la date de création. [6] propose une mesure de qualité des données qui prend en compte leur contexte d'utilisation et leur utilité. Dans MKM, la mesure de qualité se concentre sur la satisfaction d'un utilisateur.

La confiance, la croyance et la réputation sont des termes étroitement liés. La confiance est largement étudiée dans la littérature et de multiples points de vues existent. Elle est généralement définie comme une relation ternaire, valide dans un contexte donné, entre un acteur qui donne sa confiance, un acteur qui reçoit la confiance et l'objet de la confiance (par exemple, être un bon cuisinier), comme proposé par Cook [5]. Selon Grandison [8], l'objet de la confiance correspond à la réussite d'une action spécifique réalisée par la personne qui reçoit la confiance ou à ses compétences à faire cette action. La confiance est un processus social, et évolue avec l'historique des relations entre la personne qui fait confiance et la personne qui la reçoit.

Plusieurs modèles ont été proposés, pour représenter et évaluer la confiance. Dans le domaine de l'interaction homme-machine, les systèmes les plus populaires exploitant la confiance sont basés sur la réputation [4, 1]. Dans les e-communautés, la confiance est utilisée dans les applications de réseaux sociaux comme dans [7], où Golbeck a établi un système de recommandation de films. Pour un utilisateur donné, les scores de recommandation dépendent des évaluations de films pondérées par la réputation des utilisateurs. Certains de ces systèmes prennent aussi en compte la croyance. Par exemple, le modèle proposé dans [9] montre la relation entre confiance, croyance et réputation dans un réseau social. Ce modèle lie la confiance à un ensemble de croyances pertinentes, par exemple, si un utilisateur est un expert, ou s'il est honnête.

Dans les systèmes de RÀPC, les auteurs de [10] proposent d'intégrer la provenance d'un cas comme métaconnaissance pour guider la maintenance de la base de cas et augmenter la qualité des futurs résultats. Cependant, la qualité d'un cas initial est toujours considérée comme maximale.

Dans [13], les auteurs intègrent la confiance et la provenance dans une approche RÀPC pour proposer un modèle de recherche collaborative sur le Web. Pour une requête donnée, les pages Web sont filtrées et classées selon leur similarité avec la requête et la réputation des utilisateurs ayant déjà sélectionné la page concernée. Ici, l'indicateur de qualité agit seulement sur la base de cas et sur les cas qui ont déjà été retournés par le système.

En conclusion, l'état de l'art montre que, pour le RÀPC, les métaconnaissances sont seulement utilisées pour décrire les cas qui ont participé au processus de raisonnement. Les autres types d'UC ne sont pas associés à des métaconnaissances. L'originalité du modèle MKM est de représenter la fiabilité de toutes les UC : les cas, les connaissances du domaine et les connaissances d'adaptation. Cette fiabilité servira à filtrer les UC, à classer et à personnaliser les réponses retournées par le système.

4 Gestion des métaconnaissances dans un système de RÀPC

Cette section présente MKM, le modèle de métaconnaissances, utilisé pour représenter et calculer la fiabilité de connaissances provenant d'une e-communauté. MKM se base sur des modèles de métaconnaissances dont celui de Golbeck [7], mais contrairement à ce modèle, MKM est intégré dans un système de raisonnement et non dans un système de recommandation. L'objectif de MKM est de calculer la fiabilité de chaque UC pour un utilisateur donné, dans le but de garantir une certaine qualité dans les résultats retournés par le système de RÀPC et de personnaliser les réponses. La fiabilité est représentée par un score qui dépend de plusieurs métaconnaissances présentées dans cette section. Certaines métaconnaissances sont inférées par le système tandis que d'autres sont entrées dans le système par les utilisateurs qui évaluent les UC ou les autres utilisateurs de la communauté. Ces évaluations sont les fondations du modèle MKM.

Interactions et évaluations d'utilisateurs. Un utilisateur u peut interagir dans le système en produisant des UC et en évaluant des UC ou d'autres utilisateurs. Soit U , l'ensemble de tous les utilisateurs du système et UC l'ensemble de toutes les UC utilisées dans le système. La fonction ku_from retourne les UC produites par l'utilisateur, pour $u, v \in U$, $ku_from(u) \cap ku_from(v) = \emptyset$

$$ku_from : U \rightarrow 2^{UC}$$

Les utilisateurs peuvent évaluer les UC ou les autres utilisateurs avec un système d'évaluation, par exemple à étoiles comme sur le site d'e-Bay. Dans le modèle MKM, ces évaluations sont normalisées dans l'intervalle $[0, 1]$.

- Quand un utilisateur u évalue une $uc \in UC$, u assigne un score de croyance à uc , $belief(u, uc)$. Ce score de croyance représente le degré d'acceptation, pour u , que uc soit vraie, selon sa connaissance.

$$belief : U \times UC \rightarrow [0, 1] \cup \{?\}$$

où ? représente la valeur inconnue ($\text{belief}(u, uc) = ?$ signifie que u n'a pas évalué uc).

- Quand un utilisateur u évalue un autre utilisateur v , u assigne un score de confiance a priori à l'utilisateur v . Ce score de confiance a priori représente le degré d'acceptation que l'utilisateur v soit un utilisateur digne de confiance pour u selon des informations subjectives indépendantes des UC produites par v . Plus un score de confiance est élevé, plus u considérera qu'une UC uc produite par v est (ou sera) vraie.

$$\text{a_priori_trust} : U \times U \rightarrow [0, 1] \cup \{?\}$$

Les scores de croyance (resp. confiance a priori) sont directement donnés par les utilisateurs. Les autres métaconnaissances sont ensuite calculées à partir de ces deux scores.

Confiance. La confiance d'un utilisateur u envers un utilisateur v représente le degré d'attente qu'une connaissance apportée par v à la communauté soit vraie pour u . Le score de confiance dépend du score de confiance a priori assigné par u pour v , et de tous les scores de croyance assignés par u aux connaissances produites par v .

Le multiensemble des scores de croyance que u assigne aux connaissances produites par v , est retourné par la fonction $\text{user_belief_scores}(u, v)$; il résulte des fonctions $\text{ku_from}(v)$ et $\text{belief}(u, uc)$ où uc est la connaissance que v a produit :

$$\begin{aligned} \text{user_belief_scores} : U \times U &\rightarrow 2^{[0,1]} \\ (u, v) &\mapsto \{\text{belief}(u, uc) \mid \\ &uc \in \text{ku_from}(v)\} \setminus \{?\} \end{aligned}$$

$\text{trust}(u, v)$ est la mesure de la confiance que u a envers v .

$$\text{trust} : U \times U \rightarrow [0, 1] \cup \{?\}$$

La confiance est calculée comme suit (pour $u, v \in U$) :

- Si u n'a jamais évalué de UC produites par v (i.e., $\text{user_belief_scores}(u, v) = \emptyset$) alors, $\text{trust}(u, v) = \text{a_priori_trust}(u, v) \in [0, 1] \cup \{?\}$
- Sinon, soit $n = |\text{user_belief_scores}(u, v)|$, $n \neq 0$.
 - Si u n'a pas assigné de score de confiance a priori envers v (i.e., $\text{a_priori_trust}(u, v) = ?$) alors, la confiance est la moyenne

des scores de croyance que u a assigné aux UC produites par v .

$$\text{trust}(u, v) = \frac{1}{n} \sum_{s \in \text{user_belief_scores}(u, v)} s \quad (1)$$

- Sinon, la confiance est calculée par la combinaison de la confiance a priori et des croyances de u :

$$\begin{aligned} \text{trust}(u, v) &= \alpha_n \text{a_priori_trust}(u, v) \\ &+ (1 - \alpha_n) \frac{1}{n} \sum_{s \in \text{user_belief_scores}(u, v)} s \\ \text{où } \alpha_n &= \frac{1}{n + 1} \end{aligned} \quad (2)$$

Plus le nombre de scores de croyance assignés par u aux UC produites par v est important, moins le score de confiance a priori a d'influence (car $\lim_{n \rightarrow \infty} \alpha_n = 0$) : asymptotiquement, les expressions (1) et (2) sont équivalentes.

Réputation. La réputation d'un utilisateur v est la perception qu'a la communauté envers v , basée sur leurs précédentes interactions. La réputation permet de guider l'estimation de la véracité de la connaissance produite par v . Le score de réputation dépend des scores de confiance de la communauté envers v .

Le multiensemble des scores de confiance de la communauté envers v , est retourné par la fonction $\text{community_trust_scores}(v)$:

$$\begin{aligned} \text{community_trust_scores} : \mathbb{U} &\rightarrow 2^{[0,1]} \\ v &\mapsto \{\text{trust}(u, v) \mid u \in \mathbb{U}\} \setminus \{?\} \end{aligned}$$

$\text{reputation}(v)$ est une estimation de la mesure de confiance qu'a toute la communauté envers v .

$$\text{reputation} : u \rightarrow [0, 1] \cup \{?\}$$

Pour $v \in \mathbb{U}$, la réputation est calculée comme suit :

- Si $|\text{community_trust_scores}(v)| < \tau$
alors $\text{reputation}(v) = \text{default_reputation}$,

$$\text{– Sinon } \text{reputation}(v) = \frac{\sum_{u \in \mathbb{U}, u \neq v} \text{trust}(u, v)}{|\text{community_trust_scores}(v)|}.$$

Le score de réputation de v est la moyenne des ensembles $\text{trust}(u, v)$. Pour les utilisateurs qui ont obtenu un nombre d'évaluations inférieur à un seuil donné τ , le score de réputation est assigné à une valeur par défaut entre 0 et 1, notée `default_reputation`.

Qualité. Le score de qualité d'une UC uc représente sa qualité du point de vue de la communauté. Ce score est indépendant du producteur de l'UC et peut-être considéré comme une estimation du degré moyen de satisfaction d'un utilisateur après l'utilisation de uc . Le score de qualité de uc dépend des scores de croyance assignés à uc .

Le multiensemble des scores de croyance d'une UC uc , noté $\text{ku_belief_scores}(uc)$, représente toutes les évaluations que la communauté a assigné à uc :

$$\begin{aligned} \text{ku_belief_scores} : UC &\rightarrow 2^{[0,1]} \\ uc &\mapsto \{\text{belief}(u, uc) \mid u \in U\} \setminus \{?\} \end{aligned}$$

La qualité d'une UC uc , notée $\text{community_quality}(uc)$, est la moyenne de $\text{ku_belief_scores}(uc)$:

$$\begin{aligned} \text{community_quality} : UC &\rightarrow [0, 1] \cup ? \\ uc &\mapsto \frac{\sum_{s \in \text{ku_belief_scores}(uc)} s}{|\text{ku_belief_scores}(uc)|} \end{aligned}$$

Fiabilité. La fiabilité d'une UC uc quantifie la faculté de uc à être non seulement vraie et fiable, mais aussi réellement fiable pour un utilisateur u . Pour un utilisateur u , la fiabilité d'une UC uc produite par un utilisateur v , dépend du score de la réputation de v , du score de la confiance que u a envers v , et du score de la qualité de uc . Puisque la fiabilité est basée sur la confiance de u envers v , la fiabilité d'une connaissance est propre à un utilisateur (le score de confiance varie d'un utilisateur à l'autre).

$$\begin{aligned} \text{reliability} : U \times UC &\rightarrow [0, 1] \\ (u, uc) &\mapsto w_{\text{reputation}} \text{reputation}(v) \\ &\quad + w_{\text{trust}} \text{trust}(u, v) \\ &\quad + w_{\text{quality}} \text{community_quality}(uc) \\ &\text{où } w_{\text{reputation}} + w_{\text{trust}} + w_{\text{quality}} = 1 \end{aligned}$$

Cette fonction suppose que $\text{trust}(u, v) \neq ?$ et $\text{community_quality}(uc) \neq ?$. Si $\text{trust}(u, v) = ?$ et/ou $\text{community_quality}(uc) = ?$, alors ils ne sont pas pris en compte dans la fonction.

5 Intégration du modèle MKM dans un système de RÀPC

Deux fonctions permettent de modifier un système de RÀPC classique pour prendre en compte la fiabilité des connaissances (voir FIGURE 1).

Une fonction de filtrage est utilisée pour sélectionner l'ensemble de connaissances *suffisamment fiables* selon la requête et l'utilisateur qui l'émet. Cette fonction, notée `to_be_filtered(u, uc)`, dépend de `reliability(u, uc)` ; elle retourne vraie, ssi le score de fiabilité `uc` pour `u` est supérieur ou égal à un seuil donné $\beta \in [0, 1]$. La façon de fixer ce seuil peut être envisagée de plusieurs façons. Ceci peut être laissé à la charge de l'utilisateur qui pourra ainsi paramétrer le système de RÀPC selon son propre degré d'exigence. Il pourrait également être fixé expérimentalement par l'évaluation de la qualité du système de RÀPC (cf. section 5) avec l'utilisation de différents seuils. Dans tous les cas, l'impact du seuil β sur la qualité du système est un élément qui devra être étudié.

$$\begin{aligned} \text{to_be_filtered} : U \times UC &\rightarrow \{\text{true}, \text{false}\} \\ (u, uc) &\mapsto (\text{reliability}(u, uc) \geq \beta) \end{aligned}$$

Si `to_be_filtered(u, uc)`, alors `uc` est utilisée dans les inférences du moteur RÀPC, sinon `uc` n'est pas considérée.

Une fonction de classement est utilisée pour ordonner l'ensemble des réponses grâce aux métaconnaissances associées aux UC impliquées dans le calcul des résultats. L'idée est d'associer à un utilisateur `u` et une inférence, un score, nommé la *fiabilité inférée*. Soit $\{uc_1, \dots, uc_n\} \vdash uc$, une inférence effectuée par le moteur RÀPC : les `uci`'s sont les différentes UC servant à inférer une UC `uc`. Par exemple, si un cas `c` est adapté grâce à la règle d'adaptation `a` en un cas `c'`, $\{c, a\} \vdash c'$ dénote cette adaptation. La fiabilité inférée est calculée grâce à la fonction d'agrégation \otimes appliquée à la fiabilité des `uci`'s :

$$\begin{aligned} \text{inferred_reliability}(u, \{uc_1, \dots, uc_n\} \vdash uc) \\ \mapsto \otimes_{1 \leq i \leq n} \text{reliability}(u, uc_i) \end{aligned}$$

Plusieurs fonctions d'agrégation \otimes sont possibles : par exemple, une fonction d'agrégation moyenne (correspondant à une approche probabiliste) ou encore un minimum (correspondant à une mesure de la nécessité de l'approche possibiliste). L'ensemble des réponses retournées peut être trié par ordre décroissant de `inferred_reliability`.

Illustration de l'apport de MKM pour le système TAAABLE

Pour une requête donnée, TAAABLE retourne des réponses de la forme « recette + adaptation ». Supposons que pour la requête « Je voudrais un gratin avec du poireau » posée par Tom, TAAABLE renvoie dans cet ordre : la recette R_1 (*Gratin de poireaux*) qui ne nécessite pas d'adaptation, la recette R_2 (*Gratin de carottes*) avec l'adaptation A_1 = « remplacer l'oignon par du poireau », et la recette R_3 (*Endives au gratin*) avec l'adaptation A_2 = « remplacer l'endive par du poireau ». Sans l'extension MKM, les résultats retournés par TAAABLE sont les mêmes pour tous les utilisateurs.

Avec l'extension MKM, les résultats retournés par TAAABLE seront personnalisés pour Tom. Supposons que dans le modèle MKM, la fiabilité, pour Tom, de R_1 soit de 0.3 (parce que la recette est mauvaise), et celles de R_2 et de R_3 soient de 0.7. Supposons également que la fiabilité de A_1 et A_2 , soit respectivement de 0.5 et 0.9.

Avec un seuil de filtre $\beta = 0.5$ et en utilisant la fonction d'agrégation moyenne, le résultat serait le suivant : $\{R_3, A_2\} \vdash R'_3$ avec une fiabilité inférée de 0.8 précéderait $\{R_2, A_1\} \vdash R'_2$ de fiabilité inférée plus faible (0.6). R_1 , qui bien que ne nécessitant aucune adaptation, ne serait plus proposée en raison de sa fiabilité de 0.3, inférieure à β (0.5). Pour chaque proposition, l'utilisateur visualise le nom de la recette et son adaptation si la recette doit être adaptée pour répondre à sa requête. Avec le modèle MKM, le score de fiabilité inféré est également présenté. Les MKM intermédiaires (qualité, confiance et réputation) ne sont, quant à elles, pas affichées à l'utilisateur lors de la présentation des résultats mais sont consultables en tant que propriétés d'UC.

Évaluation

L'apport de MKM par rapport à un système de RÀPC classique reste à être étudié. Cependant, nous pouvons d'ores et déjà présenter la méthodologie que nous avons adoptée. Ainsi, la mise en place de cette étude est en cours et a lieu à travers le système TAAABLE avec l'utilisation des connaissances communautaires dans le domaine de la cuisine encodées dans le site Web de cuisine ATAAABLE (<http://ataaable.loria.fr/>). Cet espace web collaboratif permet de manipuler les cas, les connaissances d'adaptation, les utilisateurs, et dans lequel les utilisateurs peuvent évaluer les UC et les autres utilisateurs. Deux hypothèses vont être étudiées : (H1) raisonner sur les connaissances les plus fiables donne de meilleurs résultats que de raisonner sur toutes les connaissances, et (H2) la fiabilité des connaissances peut être représentée grâce à un ensemble de pièces de mé-

taconnaissances. L'évaluation et la validation de ces hypothèses se fera de façon continue à travers des interactions intégrées au système TAAABLE permettant à l'utilisateur de donner des retours sur les résultats, et de façon ciblée à travers des campagnes de tests-utilisateurs. (H1) pourra être testée à travers les critères utilisés classiquement dans l'évaluation de systèmes de recherche d'information, tels que la précision et le rappel. Dans ce cadre, des jeux de tests sont actuellement en cours d'élaboration ; un test est ici constitué des réponses pertinentes et non pertinentes attendues pour une requête donnée. La mise en place d'un système de tests automatisés est également prévu à travers BEGOOD, un système générique de tests [3], qui permettra de tester la proportion de bonnes/mauvaises réponses obtenues avec et sans MKM. L'utilisation de ce système automatique de tests permettra également de déterminer la meilleure façon de positionner les poids des trois métaconnaissances : confiance, réputation et qualité, dans le calcul de la fiabilité et de valider ou invalider (H2).

6 Conclusion

Dans cet article, nous avons présenté une approche pour raisonner avec un système de RÀPC sur des connaissances partiellement fiables provenant d'une e-communauté. Celle-ci propose d'associer des métaconnaissances concernant la fiabilité des connaissances, basées sur des évaluations d'utilisateurs, et un modèle a été présenté.

La mise en oeuvre opérationnelle du modèle MKM est en cours à travers le système TAAABLE. L'acquisition des scores de croyance et de confiance a priori est déjà possible mais il reste à intégrer les fonctions permettant de calculer les autres scores de métaconnaissances. Enfin, une modification du fonctionnement du moteur de RÀPC, avec l'intégration des fonctions de filtrage et de classement, est également prévue à court terme afin de réaliser l'évaluation du système présenté en section 5.

Références

- [1] ARTZ D. & GIL Y. (2007). A survey of trust in computer science and the Semantic Web. *Web Semantics : Science, Services and Agents on the World Wide Web*, **5**(2), 58–71.
- [2] BADRA F., BENDAOU D., BENTEBITEL R., CHAMPIN P.-A., COJAN J., CORDIER A., DESPRÉS S., JEAN-DAUBIAS S., LIEBER J., MEILENDER T., MILLE A., NAUER E., NAPOLI A. & TOUSSAINT Y. (2008). Taaable : Text Mining, Ontology Engineering, and Hierarchical Classification for Tex-

- tual Case-Based Cooking. In *ECCBR Workshops, Workshop of the First Computer Cooking Contest*, p. 219–228.
- [3] CANALS G., CORDIER A., E. D., L. I.-B. & NAUER E. (2013). Collaborative knowledge acquisition under control of a non-regression test system. In *Semantic Web Collaborative Spaces @ ESWC*.
- [4] CASTELFRANCHI C. & FALCONE R. (1998). Principles of Trust for MAS : Cognitive Anatomy, Social Importance, and Quantification. In *Third International Conference on Multi Agent Systems (ICMAS98)*, p.72.
- [5] COOK K., HARDIN R. & LEVI M. (2007). *Cooperation Without Trust ? The Russell Sage Foundation Series on Trust*. Russell Sage.
- [6] EVEN A. & SHANKARANARAYANAN G. (2007). Utility-driven assessment of data quality. *SIGMIS Database*, **38**(2), 75–93.
- [7] GOLBECK J. (2005). *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland.
- [8] GRANDISON T. & SLOMAN M. (2003). Trust management tools for internet applications. In *1st international conference on Trust management*, p. 91–107.
- [9] KNAP T. & MLYNKOVÁ I. (2011). Revealing beliefs influencing trust between members of the czech informatics community. In *Third international conference on Social informatics*, p. 226–239.
- [10] LEAKE D. & WHITEHEAD M. (2007). Case provenance : The value of remembering case sources. In *7th international conference on Case-Based Reasoning : Case-Based Reasoning Research and Development*, p. 194–208.
- [11] NAUDET Y., LATOUR T., VIDOU G. & DJAGHLOUL Y. (2010). Towards a novel approach for high-stake decision support system based on community contributed knowledge base. In *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, p. 730 –736.
- [12] RIESBECK C. & SCHANK R. (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey : Lawrence Erlbaum Associates, Inc.
- [13] SAAYA Z., SMYTH B., COYLE M. & BRIGGS P. (2011). Recommending case bases : Applications in social web search. In *Case-Based Reasoning Research and Development*, p. 274–288. Springer Berlin Heidelberg.
- [14] SKAF-MOLLI H., DESMONTILS E., NAUER E., CANALS G., CORDIER A. & LEFEVRE M. (2012). Knowledge Continuous Integration Process (K-CIP). In *21st World Wide Web Conference - Semantic Web Collaborative Spaces workshop*, p. 1075–1082.
- [15] WANG R. (1998). A product perspective on total data quality management. *Commun. ACM*, **41**(2), 58–65.