



**HAL**  
open science

## Case-Based Reasoning on E-Community Knowledge

Emmanuelle Gaillard, Jean Lieber, Yannick Naudet, Emmanuel Nauer

► **To cite this version:**

Emmanuelle Gaillard, Jean Lieber, Yannick Naudet, Emmanuel Nauer. Case-Based Reasoning on E-Community Knowledge. ICCBR 2013 - 21st International Conference on Case-Based Reasoning, Jul 2013, Saratoga Springs, NY, United States. pp.104-118, 10.1007/978-3-642-39056-2\_8. hal-00918518

**HAL Id: hal-00918518**

**<https://inria.hal.science/hal-00918518>**

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.

# Case-based reasoning on e-community knowledge

Emmanuelle Gaillard<sup>1235</sup>, Jean Lieber<sup>1235</sup>,  
Yannick Naudet<sup>46</sup>, and Emmanuel Nauer<sup>1235</sup>

<sup>1</sup> Université de Lorraine, LORIA 54506 Vandœuvre-lès-Nancy, France

<sup>2</sup> CNRS 54506 Vandœuvre-lès-Nancy, France

<sup>3</sup> Inria 54602 Villers-lès-Nancy, France

<sup>4</sup> CRP Henri Tudor, Luxembourg

<sup>5</sup>firstname.lastname@loria.fr, <sup>6</sup>firstname.lastname@tudor.lu.

**Abstract.** This paper presents MKM, a meta-knowledge model to manage knowledge reliability, in order to extend a CBR system so that it can reason on partially reliable, non expert, knowledge from the Web. Knowledge reliability is considered from the point of view of the decision maker using the CBR system. It is captured by the MKM model including notions such as belief, trust, reputation and quality, as well as their relationships and rules to evaluate knowledge reliability. We detail both the model and the associated approach to extend CBR. Given a problem to solve for a specific user, reliability estimation is used to filter knowledge with high reliability as well as to rank the results produced by the CBR system, ensuring the quality of results.

**Keywords:** case-based reasoning, meta-knowledge, reliability, filtering, ranking, personalization

## 1 Introduction

The social Web is known to generate an enormous amount of information, constituting virtually a big knowledge base about almost any kind of subject. This knowledge base is mainly created by e-communities, consisting of people sharing common ideas, goals or interests, communicating over the Internet or any other technological communication network. The knowledge asset built by e-communities on the Web has a high exploitation potential ranging from knowledge management to, e.g., data-mining applications for detecting interesting trends or tracking users for personalization purposes. Such knowledge is influenced by different human factors such as belief, confidence, or trust, which influence its reliability, not only from a human perspective but also for automated reasoning and decision making [1].

In this context, the work presented here focuses on the exploitation of partially reliable e-community knowledge for Case-Based Reasoning (CBR), in contrast to classical approaches that rely on consensual and validated expert knowledge. In order to preserve the quality of reasoning, we have designed a meta-knowledge model, called MKM, to manage knowledge coming from (non-expert)

users. MKM represents knowledge reliability using meta-knowledge such as belief, trust, reputation, quality, so that the CBR engine can take it into account in its reasoning process.

This paper is organized as follows. Section 2 gives the motivations for this research. We present the state of the art related to the use of meta-knowledge to characterize knowledge reliability and especially in CBR systems in Section 3. Section 4 details the global process for meta-knowledge management, and Section 5 shows how to extend a classical CBR system to handle partially reliable knowledge. Section 6 presents a use-case in the cooking domain and Section 7 concludes this paper.

## 2 Motivation

The research work presented here is motivated by the will to extend the traditional application field of CBR systems, and especially adapt the latter for processing web-originated data. Indeed, traditional CBR systems are used in a closed world where the manipulated knowledge base is fed with expert data, consensual and validated. Because of this, CBR systems can hardly be used when expert data is sparse and especially on the web, where knowledge reliability is not guaranteed and difficult to measure.

Knowing this, an important question is to evaluate if reasoning on not fully reliable knowledge coming from e-communities can be exploited in a CBR approach and if at least results of a similar quality than when reasoning on expert knowledge can be obtained.

Classical CBR systems are usually composed of four knowledge containers: the cases, the domain knowledge (i.e., ontology of the domain), the adaptation knowledge, and the similarity (i.e., retrieval) knowledge. Knowledge is validated by experts; this entails to reason with a limited amount of knowledge. Some studies, like the Kolflow project (<http://kolflow.univ-nantes.fr/>), have investigated ways to manage knowledge provided by an e-community. The Kolflow approach consists in improving the man-machine collaboration to ensure a collaborative knowledge construction. It allows to collect big amount of knowledge from users, and uses non-regression tests to ensure that new incoming knowledge does not affect the results of the reasoning process that exploits the knowledge [2]. More particularly in the CBR domain, Richards [3] has investigated decision support from a knowledge base constituted of experts and informal knowledge gathered from collective tools. She has proposed a collaborative knowledge engineering approach mixing CBR and rules, named Collaborative Multiple Classification Ripple Down Rules (C-MCRDR). C-MCRDR proved being efficient to build collaboratively a knowledge base. However, partial reliability of knowledge is handled implicitly in a manual human-driven review and negotiation process. Knowledge reliability related to truth, belief, trust or reputation, is not considered and not directly exploited in the reasoning process.

We propose to associate meta-knowledge about reliability to each knowledge unit (KU) in order to allow inferences on e-community originated knowledge,

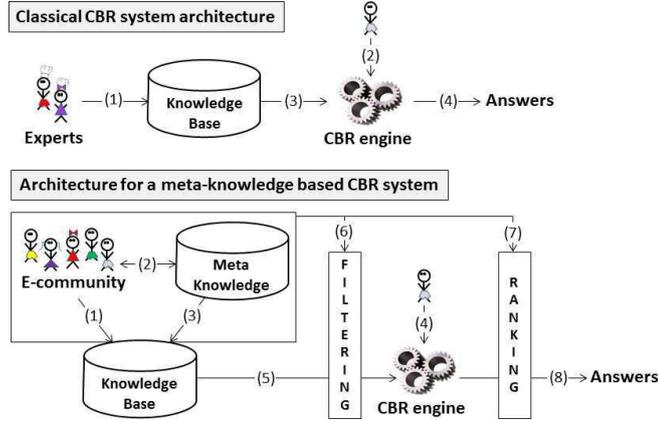


Fig. 1. CBR system classical architecture vs. architecture using meta-knowledge.

while ensuring the quality of results. Our solution consists in establishing a meta-knowledge model to describe e-communities knowledge. A new container, the meta-knowledge base, is thus added to the classical CBR architecture.

Fig. 1 shows the difference of architecture between a *classical* CBR system and a CBR system based on meta-knowledge. In the classical CBR system architecture, (1) the knowledge base is produced by experts of the domain. (2) A user queries the CBR engine which (3) uses the expert validated knowledge for (4) computing its answers. In a meta-knowledge based CBR system, (1) the knowledge base is produced by the e-community and (2) users and (3) knowledge are linked to meta-knowledge. When (4) a user queries the CBR engine, the latter (5) uses the e-community knowledge, (6) filtered by the meta-knowledge model. The final answers of the CBR engine (8) may be ranked (7) using also the meta-knowledge. Moreover, if the user who queries the system is a member of the e-community, the filtering and ranking operations produce personalized results since some meta-knowledge are user-specific.

### 3 State of the arts

#### 3.1 Meta-knowledge about reliability

While philosophical studies (e.g, [4]) associate knowledge reliability to safety and robustness, using reliable knowledge element in a knowledge-based system allows to infer knowledge with an acceptable level of trustworthiness. Knowledge reliability is influenced by several factors, sometimes interrelated, as discussed in [1], where a generic model for representing knowledge generated by e-communities is proposed. In an effort to provide a basis for exploiting partially reliable knowledge in a reasoning process, this work identifies the following dimensions for knowledge reliability: origin, context, truth, belief, value, quality, and trust. Origin (or provenance) is the source of a KU; context concerns validity conditions

associated to knowledge use; truth is the knowledge validity in the considered world, while belief is the knowledge truth from the author, provider or community perspective; value stands for the importance of the KU for its consumer; quality accounts for characteristics of the KU (e.g., precision, completeness) and finally trust, impacting many relationships in the model and more particularly between knowledge, author/provider and community.

In order to assess at best the impact of these parameters on knowledge reliability, it is necessary to first work with a reduced set of meta-knowledge. In this work we focus on knowledge quality, belief, trust and reputation. Context, as a complex notion, is not considered for the moment, as well as value, since it can be considered that a knowledge of high quality is valuable for the community. Finally, truth is highly correlated with belief notion, and they can be considered as synonyms in a first approach.

Quality is considered by researchers as a complex notion, and it is perceived as a multi-dimensional concept [5]. The nature of data and measurement of its quality have an important impact on the success of decision making. So it is important to evaluate the quality of knowledge, when creating a knowledge base. Several criteria of quality are listed in [5], such as completeness, consistency, freshness, and accuracy. Quality assessment is obtained by using a set of formulas aggregating the different criteria. [6] proposes a data quality measurement framework that takes into account the context of use and the utility of data. The quality measurement in MKM focuses on the satisfaction a user (measured thanks to a 5-star rating system) will have when using a KU.

Trust, belief and reputation are closely related terms. Trust is largely studied in literature and multiple viewpoints exist, because trust is both a component of our everyday life and of each application domain [7]. Trust is generally defined as a ternary relation, valid in a given context, between a truster, a trustee and an object, as originally proposed by Cook et al. [8]. As for Grandison [9], the trusted object is often related to an action performed by the trustee or its ability to do it. In other words, independently of whether the trustee is a person or not, the trustee is viewed as some entity that will actually perform the action the truster expects him to do. Trust is a social process, and evolves dynamically following the history of the relationship. In the human computer interaction domain, Golbeck [10] asserts that “A trusts B if A commits to an action based on the belief that B’s future actions will lead to a good outcome.” She used this definition in her recommending system for movies, where users can rate both movies and other users. For a given user, movie recommendation scores are computed by taking into account the community opinion: scores depends on movies ratings weighted by user reputation.

Several models have been proposed, both for conceptualizing and evaluating trust. The most common systems exploiting trust are based on reputation [11–14] and some of them take also belief into account. For example, the model proposed in [15] shows the relation between trust, belief and reputation in a social network. This model relates trust to a set of relevant beliefs on the evaluated user, for example if a user is an expert, or if a user is honest.

In [16], a movie is recommended to a user  $u$  if this movie has been well rated by users belonging to the trust network of  $u$ . Trust is computed in [16] from the proportion of common ratings users have put on movies they evaluated. The cold start problem is handled by finding in the available users one that is similar to a newcomer, where similarity is computed from demographic data. In our work, the MKM solution relies on predefined default values.

Trust and reputation management has been addressed in multiple domains like in multi-agent systems (MASs) (see, e.g., [17]), network security and Service Oriented Architectures [18]. Regarding e-communities, trust is used in social networks applications and addressed in the Web of Trust initiative.

### 3.2 Meta-knowledge in CBR systems

Usually, user feedback on proposed answers in CBR systems allows to judge the quality of the new cases and to repair a failed adaptation [19]. However, some authors think that the feedback approach is insufficient (by observing missing or delayed feedback) and that meta-knowledge is more relevant to improve the reasoning results [20]. In [20], the authors propose to integrate the provenance of a case, as a meta-knowledge, in a CBR system to guide the case base maintenance and increase confidence in future results. For example, a repair is propagated through generated cases from the initial case and the quality of a case is measured by the length of the adaptation path. However, quality of initial cases is set to a same maximum value. The quality does not depend on external factors and additional meta-knowledge like, e.g., provenance of initial cases is not taken into account.

In [21], authors integrate trust in addition to provenance in a CBR approach to propose a model of collaborative web search. During a user search, web pages are filtered and ranked using their relevance to the query and the reputation of users having already selected the pages. Reasoning is guided by preferences of users and not by reliability of cases. Besides, the indicator of quality acts only on the case base containers and on cases which have already been found.

In conclusion, the state of the art shows that meta-knowledge is introduced to provide explanations on CBR results or recommendations. To the best of our knowledge, during the reasoning of a CBR system, meta-knowledge like trust, belief and reputation are only used for cases created by the adaptation process or cases which have already been retrieved during a previous reasoning. Meta-knowledge representing reliability of cases are not yet used for representing new external cases. Besides, reliability of knowledge of the other containers is not represented. The novelty of MKM is to represent reliability of KUs of all the containers: cases (not only learned cases), domain, adaptation and similarity KUs. Reliability is then used to filter KUs and to rank personalized answers returned by the CBR system.

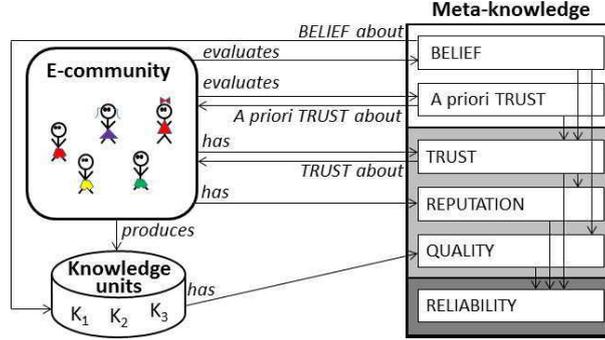


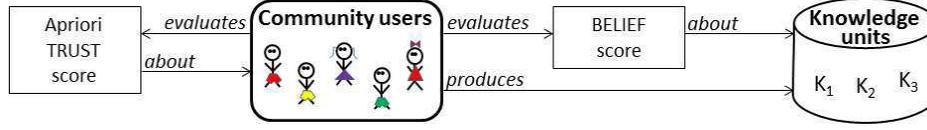
Fig. 2. Dependencies between users, knowledge units and meta-knowledge.

#### 4 Meta-knowledge management for a CBR system

This section presents MKM, the meta-knowledge model, used to represent and to compute the reliability of knowledge coming from an e-community. The objective is to compute the reliability of each KU (of the four containers) for a given user, in order to improve the CBR system reasoning and personalize the answers returned to a user. The reliability is represented by a score that depends on several meta-knowledge elements presented in the following. Some meta-knowledge is inferred by the system while some other is entered in the system by the users who evaluate KUs or by other community members. These evaluations are the foundations of MKM. Fig. 2 introduces the links between users, KUs and meta-knowledge:

- A user  $u$  may evaluate a KU  $ku$  of a knowledge container by a belief score which represents the belief  $u$  has in  $ku$ .
- A user  $u$  may evaluate another user  $v$  by an a priori trust score which represents the trust  $u$  has towards  $v$ .
- A trust score from a user  $u$  towards a user  $v$ , which represents how  $u$  trusts  $v$ , is inferred from the a priori trust score that  $u$  has assigned to  $v$  and from the belief scores that  $u$  has assigned to KUs produced by  $v$ .
- A reputation score of a user  $u$ , which represents the reputation of  $u$  in the e-community, is inferred from all the trust scores about  $u$ .
- A quality score of a KU  $ku$ , which represents the global quality of  $ku$  for the e-community, is inferred from all the belief scores about  $ku$ .
- Finally, a reliability score of a KU  $ku$  for a user  $u$ , which represents the personalized reliability of  $ku$  for  $u$ , is inferred from quality, reputation and trust scores.

To summarize, the meta-knowledge represented on a white background in Fig. 2 (belief and a priori trust) are initially entered by the users. Sayya et al. [22] show that collecting such items works well when there is a small number of users who rate frequently, leading other users of the community to give feedback.



**Fig. 3.** A priori Trust and Belief: Evaluation of knowledge and users.

The meta-knowledge on the light-grey background (trust, reputation and quality) are computed meta-knowledge, that change dynamically according to the community inputs. And finally, reliability (dark-grey background) is the meta-knowledge that will be used by the CBR system to filter knowledge and to rank answers with respect to the users trusts and beliefs. In the following, we detail all these elements and the way they are computed.

**User interactions and evaluations.** Fig. 3 illustrates how users may interact with the system, by editing some KUs, by evaluating some KUs, and by evaluating other users. In the model, let **User** be the set of all the users of the system and let **KU** be the set of all KUs (of all the containers) used in the system.

The `ku_from` function returns the KUs edited by a user, for  $u, v \in \mathbf{User}$ ,  $\text{ku\_from}(u) \cap \text{ku\_from}(v) = \emptyset$  (a  $ku \in \mathbf{KU}$  has only one producer):

$$\begin{aligned} \text{ku\_from} : \mathbf{KU} &\rightarrow 2^{\mathbf{KU}} \\ u &\mapsto \text{ku\_from}(u) \end{aligned}$$

Users may evaluate KUs or other users with, e.g., a numerical scale or a star system. In MKM, these evaluations are normalized in  $[0, 1]$ .

- When a user  $u$  evaluates a  $ku \in \mathbf{KU}$ ,  $u$  assigns a belief score to the KU  $ku$ . This belief score represents the degree of acceptance, for  $u$ , that  $ku$  may be true, according to his/her own knowledge.

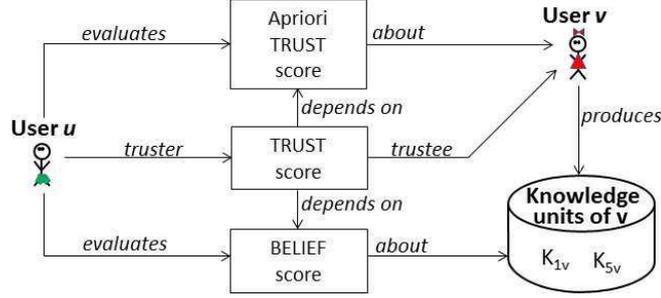
$$\begin{aligned} \text{belief} : \mathbf{User} \times \mathbf{KU} &\rightarrow [0, 1] \cup \{?\} \\ (u, ku) &\mapsto \text{belief}(u, ku) \end{aligned}$$

where ? stands for the unknown value ( $\text{belief}(u, ku) = ?$  means that  $u$  has not evaluated  $ku$ ).

- When a user  $u$  evaluates another user  $v$ ,  $u$  assigns an a priori trust score for  $v$ . This a priori trust score represents the degree of acceptance that  $v$  is a trustworthy user for  $u$  according to subjective information independently from the KUs produced by  $v$ . The higher this a priori trust score is, the higher  $u$  expects that a KU  $ku$  produced by  $v$  is true.

$$\begin{aligned} \text{a\_priori\_trust} : \mathbf{User} \times \mathbf{User} &\rightarrow [0, 1] \cup \{?\} \\ (u, v) &\mapsto \text{a\_priori\_trust}(u, v) \end{aligned}$$

Belief scores and a priori trust scores are directly given by users when evaluating KUs or other users. The other meta-knowledge is inferred from these two scores.



**Fig. 4.** Trust of a user  $u$  towards a user  $v$ , computed from the a priori trust score of  $u$  towards  $v$ , and belief scores of  $u$  towards KUs produced by  $v$ .

**Trust.** The trust of user  $u$  (the truster) regarding a user  $v$  (the trustee) represents the degree of expectation that the knowledge brought by  $v$  to the community is true for  $u$ . The trust score depends on the a priori trust score assigned by  $u$  to  $v$ , and on all the belief scores assigned by  $u$  to the knowledge produced by  $v$  (see Fig. 4).

The multi-set of belief scores that  $u$  assigns to the knowledge produced by  $v$ , is returned by the function `user_belief_scores( $u, v$ )`. This multi-set is inferred thanks to the function `ku_from( $v$ )` and `belief( $u, ku$ )` where  $ku$  is the KU that  $v$  has produced:

$$\text{user\_belief\_scores} : \text{User} \times \text{User} \rightarrow 2^{[0,1]}$$

$$(u, v) \mapsto \{\text{belief}(u, ku) \mid ku \in \text{ku\_from}(v)\} \setminus \{?\}$$

`trust( $u, v$ )` is a measure of the trust the user  $u$  has towards the user  $v$ .

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

Trust is computed as follows (for  $u, v \in \text{User}$ ):

- If  $u$  has never evaluated any KUs produced by  $v$  (i.e., `user_belief_scores( $u, v$ ) =  $\emptyset$` ) then, `trust( $u, v$ ) = a_priori_trust( $u, v$ )  $\in [0, 1] \cup \{?\}$`
- Else, let  $n = |\text{user\_belief\_scores}(u, v)|$ ,  $n \neq 0$ .
  - If  $u$  has not assigned an a priori trust score to  $v$  (i.e., `a_priori_trust( $u, v$ ) = ?`) then, trust is the average of the belief scores that  $u$  assigned to KUs produced by  $v$  (i.e., `user_belief_scores( $u, v$ )`).

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

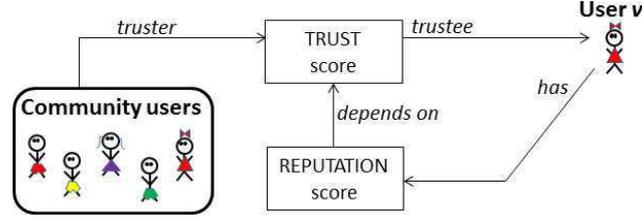


Fig. 5. Reputation of a user  $v$  is inferred from all trust scores assigned to  $v$ .

- Else, the trust is computed by a combination of the a priori trust and the belief scores given by  $u$ :

$$\begin{aligned} \text{trust}(u, v) = & \alpha_n \text{a\_priori\_trust}(u, v) \\ & + (1 - \alpha_n) \frac{1}{n} \sum_{s \in \text{user.belief.scores}(u, v)} s \end{aligned} \quad (2)$$

$$\text{where } \alpha_n = \frac{1}{n + 1}$$

The more scores have been assigned by  $u$  to the KUs provided by  $v$ , the less is the influence of the a priori trust score (since  $\lim_{n \rightarrow \infty} \alpha_n = 0$ ): asymptotically, the expressions (1) and (2) are equivalent.

**Reputation.** The reputation of a user  $v$  is the perception all the users in the community have of  $v$ , based on their previous experience with  $v$ . Reputation provides an indicator of the truth of knowledge produced by  $v$ . It depends on the inferred trust scores of the community towards  $v$  (see Fig. 5).

The multi-set of trust scores inferred for the community towards  $v$ , is returned by the function `community_trust_scores(v)`:

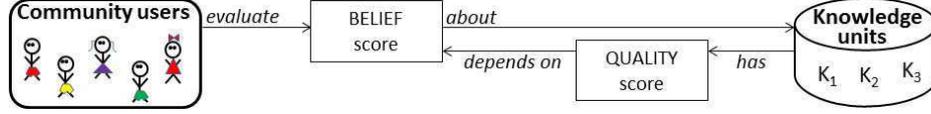
$$\begin{aligned} \text{community\_trust\_scores} : \text{User} & \rightarrow 2^{[0,1]} \\ v & \mapsto \{\text{trust}(u, v) \mid u \in \text{User}\} \setminus \{?\} \end{aligned}$$

`reputation(v)` is an estimation of the measure of the trust that has all the community towards  $v$ .

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

For  $v \in \text{User}$ , reputation is computed as follows:

- If  $|\text{community\_trust\_scores}(v)| < \tau$   
then  $\text{reputation}(v) = \text{default\_reputation}$ ,
- Else  $\text{reputation}(v) = \frac{\sum_{u \in \text{User}, u \neq v} \text{trust}(u, v)}{|\text{community\_trust\_scores}(v)|}$ .



**Fig. 6.** Quality of a knowledge computed from its belief scores.

The reputation score of  $v$  is the average of the set of  $\text{trust}(u, v)$ . For users that have obtained a number of evaluation less than a given threshold  $\tau$ , the reputation score is assigned to a default value between 0 and 1 (for example, 0.5) and is denoted by `default_reputation`.

**Quality.** The quality score of a KU  $ku$  is the computed community quality of  $ku$ . This score is independent from the KU's producer and represents the estimation of the degree of satisfaction of users after the use of  $ku$ . The quality score of  $ku$  depends on all the belief scores assigned to  $ku$  (see Fig. 6).

The multi-set of belief scores of a KU  $ku$ , denoted by `ku_belief_scores(ku)`, represents all the evaluations the community has assigned to  $ku$ :

$$\begin{aligned} \text{ku\_belief\_scores} : \text{KU} &\rightarrow 2^{[0,1]} \\ ku &\mapsto \{\text{belief}(u, ku) \mid u \in \text{User}\} \setminus \{?\} \end{aligned}$$

The quality knowledge of  $ku$ , denoted by `community_quality(ku)`, is the average of `ku_belief_scores(ku)`:

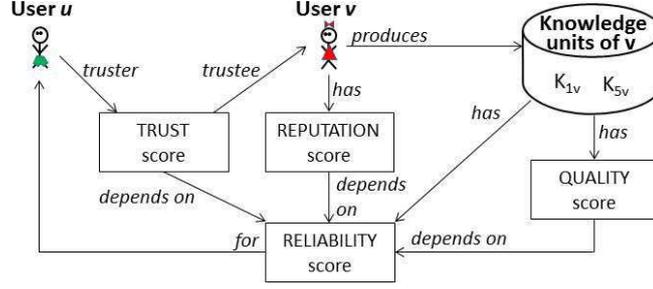
$$\begin{aligned} \text{community\_quality} : \text{KU} &\rightarrow [0, 1] \cup ? \\ ku &\mapsto \frac{\sum_{s \in \text{ku\_belief\_scores}(ku)} s}{|\text{ku\_belief\_scores}(ku)|} \end{aligned}$$

**Reliability.** The reliability corresponds to how much a user  $u$  can rely on a KU  $ku$ , or how  $ku$  is actually useful for him/her. This reliability score will be used by the CBR system for filtering knowledge, as well as for ranking.

For a user  $u$  (see Fig. 7), the reliability of a KU  $ku$  produced by a user  $v$ , depends of the reputation of  $v$ , the trust score of  $u$  towards  $v$  (if it exists), and the quality of  $ku$  (if it exists). Because reliability is based on the trust score of  $u$  towards  $v$ , the reliability of knowledge is a personalized score (the trust score varies from one user to another).

$$\begin{aligned} \text{reliability} : \text{User} \times \text{KU} &\rightarrow [0, 1] \\ (u, ku) &\mapsto w_{\text{reputation}} \text{reputation}(v) + w_{\text{trust}} \text{trust}(u, v) \\ &\quad + w_{\text{quality}} \text{community\_quality}(ku) \\ &\text{where } w_{\text{reputation}} + w_{\text{trust}} + w_{\text{quality}} = 1 \end{aligned}$$

This function assumes that  $\text{trust}(u, v) \neq ?$  and  $\text{community\_quality}(ku) \neq ?$ . If  $\text{trust}(u, v) = ?$  and/or  $\text{community\_quality}(ku) = ?$  then they are not taken into account.



**Fig. 7.** Reliability of a KU  $ku$  for a user  $u$  is computed from the quality score of  $ku$ , the reputation score of  $v$ , the producer of  $ku$ , and the trust of  $u$  towards  $v$ .

## 5 Plugging the meta-model into a CBR system

This section presents how a classical CBR system can be modified in order to take into account the reliability of knowledge (see Fig. 1). A filter function is used to select the *more reliable* set of knowledge according to the query and to the user who queries the system. A ranking function is used to order the set of answers according to the meta-knowledge associated to the KUs involved in the computation of the results.

### 5.1 Filtering

According to MKM, the knowledge which is not sufficiently reliable for a user is filtered according to the reliability score. The filtering function,  $\text{to\_be\_filtered}(u, ku)$ , depends on  $\text{reliability}(u, ku)$ . It returns true iff the reliability score  $ku$  for  $u$  is higher than a given threshold  $\beta \in [0, 1]$ .

$$\begin{aligned} \text{to\_be\_filtered} : \text{User} \times \text{KU} &\rightarrow \{\text{true}, \text{false}\} \\ (u, ku) &\mapsto (\text{reliability}(u, ku) \geq \beta) \end{aligned}$$

If  $\text{to\_be\_filtered}(u, ku)$ , then  $ku$  is used in inferences by the CBR engine, else it is not considered.

### 5.2 Ranking

The answers computed by the CBR engine will be ranked according to MKM. The idea is to associate to a user  $u$  and an inference, a score, called the *inferred reliability*. Let  $\{ku_1, \dots, ku_n\} \vdash ku$  denote an inference performed by the CBR engine: the  $ku_i$ 's are the KUs taken from the knowledge containers and  $ku$  is an inferred knowledge unit. For example, if a case  $c$  is adapted thanks to an adaptation rule  $a$  into a case  $c'$ ,  $\{c, a\} \vdash c'$  denotes this adaptation inference. The

inference reliability is computed thanks to an aggregation function  $\otimes$  applied to the reliability of the  $ku_i$ 's:

$$\text{inferred\_reliability}(u, \{ku_1, \dots, ku_n\} \vdash ku) \mapsto \bigotimes_{1 \leq i \leq n} \text{reliability}(u, ku_i)$$

There are several possible aggregation functions  $\otimes$ : it can be an average aggregation function (corresponding to a probabilistic approach) or a minimum (corresponding to a necessity measure of a possibilistic approach).

By abuse of notation,  $\text{inferred\_reliability}(u, \{ku_1, \dots, ku_n\} \vdash ku)$  is denoted by  $\text{inferred\_reliability}(u, ku)$ . Thus, given several inferred knowledge units  $ku^1, \dots, ku^p$  and a user  $u$ , the  $ku^j$ 's can be ranked according to decreasing  $\text{inferred\_reliability}(u, ku^j)$ .

### 5.3 Prerequisites

To apply the filter and ranking functions, a classical CBR system that would be adapted to reason on an e-community knowledge, must:

- provide the possibility for users to manage KUs of the different containers in order to assign them a belief score;
- be able to filter KUs it will use according to their reliability;
- return the set of KUs that are involved in the computation of each answers, in order to rank these answers.

## 6 Use case: adapting cooking recipes

This section presents a use case, in the framework of TAAABLE (<http://taaable.fr/>). TAAABLE is a CBR system which retrieves and creates cooking recipes by adaptation [23].

TAAABLE is based on the four classical knowledge containers. The domain knowledge container contains an ontology of the cooking domain that is used to retrieve the source cases that are the most similar to a target case (i.e. the query). This ontology includes several hierarchies (about food, dish types, etc.). A generalization cost is associated to each edge in the hierarchy, and is used to compute similarity between cases and the query. For example, cost associated on the edge connecting *GreenOnion* and *Onion* is 0.3.

The case base is composed of recipes. Each recipe  $R$  is transformed into  $idx(R)$ , the index of the recipe  $R$  which is a conjunction of concepts of the domain ontology. Four recipes are given in example in Table 1.  $idx(R_1)$  is a formal and abstracted representation of the recipe  $R_1$  which is a gratin dish and whose ingredients are green onion, leek, béchamel sauce, and ham (and nothing else).

According to a query entered in the system by a user, and also represented by a conjunction of concepts, the system searches in the case base for some cases (recipes) satisfying the query. For example,  $Q = \textit{GratinDish} \wedge \textit{Leak} \wedge \textit{Ham} \wedge$

**Table 1.** Four examples of recipes and their indexes.

<i>Id</i>	<i>Title</i>	<i>idx(R<sub>i</sub>)</i>
<i>R</i> <sub>1</sub>	Leeks gratin	<i>GratinDish</i> $\wedge$ <i>Leek</i> $\wedge$ <i>Potato</i> $\wedge$ <i>Bechamel</i> $\wedge$ <i>Ham</i>
<i>R</i> <sub>2</sub>	Carrots gratin	<i>GratinDish</i> $\wedge$ <i>GreenOnion</i> $\wedge$ <i>Carrot</i> $\wedge$ <i>Bechamel</i> $\wedge$ <i>Ham</i>
<i>R</i> <sub>3</sub>	Endives gratin	<i>GratinDish</i> $\wedge$ <i>Endive</i> $\wedge$ <i>Lemon</i> $\wedge$ <i>Bechamel</i> $\wedge$ <i>Ham</i>
<i>R</i> <sub>4</sub>	Salmon paste	<i>GratinDish</i> $\wedge$ <i>Pasta</i> $\wedge$ <i>Salmon</i> $\wedge$ <i>Bechamel</i> $\wedge$ <i>Ham</i>

**Table 2.** Reliability inferred for Tom, for recipes and adaptations.

<i>Id</i>	<i>Title</i>	Reliability	<i>Id</i>	$\sigma$	Reliability
<i>R</i> <sub>1</sub>	Leeks gratin	0.4	<i>A</i> <sub>1</sub>	<i>GreenOnion</i> $\rightsquigarrow$ <i>Leek</i>	0.5
<i>R</i> <sub>2</sub>	Carrots gratin	0.6	<i>A</i> <sub>2</sub>	<i>Endive</i> $\rightsquigarrow$ <i>Leek</i>	0.6
<i>R</i> <sub>3</sub>	Endives gratin	0.9	<i>A</i> <sub>3</sub>	<i>Salmon</i> $\rightsquigarrow$ <i>Leek</i>	0.8
<i>R</i> <sub>4</sub>	Salmon paste	0.8			

*Bechamel* represents the query “I would like a recipe of gratin dish with leak, ham and béchamel sauce.” Recipes matching exactly the query, if they exist, are returned to the user and are ranked at the first place (for example *R*<sub>1</sub> matches exactly *Q*). The system searches also similar recipes, using the hierarchies to generalize the user query. Recipes that have a similarity measure higher than a given threshold are retrieved and ranked by similarity. Adaptation consists in substituting some ingredients of the source cases by the ones required by the user, and is encoded by a substitution  $\sigma = A \rightsquigarrow B$ , meaning that “*A* has to be substituted by *B*”. An adaptation has a cost, denoted by  $cost(A \rightsquigarrow B)$ . The lesser the cost of adaptation of *R*<sub>1</sub> is, the higher is the similarity of *R*<sub>1</sub> with *Q*. The final result of TAAABLE for *Q*, ordered by increasing similarity, is *R*<sub>1</sub> (no adaptation), *R*<sub>2</sub> with the adaptation *GreenOnion*  $\rightsquigarrow$  *Leek*, *R*<sub>3</sub> with the adaptation *Endive*  $\rightsquigarrow$  *Leek*, *R*<sub>2</sub> is retrieved before *R*<sub>3</sub> because *GreenOnion* is closer to *Leek* than *Endive* (i.e.  $cost(GreenOnion \rightsquigarrow Leek) < cost(Endive \rightsquigarrow Leek)$ ). *R*<sub>4</sub> is not retrieved because the cost of substitution of *Salmon* by *Leek* is too high to adapt *R*<sub>4</sub> to *Q*.

TAAABLE uses also adaptation knowledge (AK) of the form  $(context, \sigma)$  [24]: *context* is the recipe or the class of recipes on which the substitution  $\sigma$  can be applied. For example, the AK where *Salmon* could be replaced with *Leek* in *R*<sub>4</sub> is represented by  $(R_4, Salmon \rightsquigarrow Leek)$ . Using this AK, *R*<sub>4</sub> which was not proposed by the query generalization process is now proposed.

### Example of reasoning with the meta-knowledge approach

A meta-knowledge container is added to take into account reliability. It will be used to modify the list of answers returned by TAAABLE. Table 2 shows the reliability of recipes and reliability of adaptations for a specific user: Tom.

Using MKM, unreliable KUs (e.g. which are lower than a given reliability threshold) will be filtered and will not participate to the reasoning process. For

example, with a threshold fixed to 0.6, the recipe  $R_1$  and the adaptations  $A_1$  are eliminated, for the user Tom.

The answers can be ranked according to the reliability of the KUs that are involved, for example, when using the average aggregation function, in the following order:  $R_4$  with  $A_3$ ,  $R_3$  with  $A_2$ . Indeed, with the average aggregation function, the ranking score of  $R_4$  with  $A_3$ , which is 0.8, is higher than the ranking score of  $R_3$  with  $A_2$ , which is 0.75.  $R_2$  with  $A_1$  is not proposed because  $A_1$  cannot be applied anymore. These ranked answers are specific to Tom, because the reliability of a knowledge for Tom is influenced by the trust score he has towards users of the system who produced the knowledge.

## 7 Conclusion

In this paper, we presented an approach for reasoning on partially reliable e-community knowledge by contrast to consensual and validated knowledge in classical CBR systems. The approach proposes to associate meta-knowledge about reliability of knowledge, based on user evaluations, and a model has been presented.

We have illustrated the interest of the approach with an example in the community cooking domain. Ongoing work consists in implementing the use case presented in section 6 in the framework of the TAAABLE project. A collaborative web space which allows to manage cases, AK, and users, and in which users may evaluate these KUs and other users has already been developed. The functions for computing the different meta-knowledge scores must yet be integrated in this collaborative work space. Experiments with the TAAABLE e-community will be driven at short term in order to evaluate if using MKM and knowledge from an e-community provides similar or better results than those obtained using the classical architecture exploiting consensual knowledge.

## References

1. Y. Naudet, T. Latour, G. Vidou, and Y. Djaghoul. 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)*, pages 730–736, 2010.
2. H. Skaf-Molli, E. Desmontils, E. Nauer, G. Canals, A.e Cordier, and M. Lefevre. Knowledge Continuous Integration Process (K-CIP). In *21st World Wide Web Conference - Semantic Web Collaborative Spaces workshop*, pages 1075–1082, 2012.
3. D. Richards. A social software/web 2.0 approach to collaborative knowledge engineering. *Information Sciences*, 179(15):2515 – 2523, 2009.
4. V.F. Hendricks and D. Pritchard. *New waves in epistemology*. New Waves in Philosophy. Palgrave Macmillan, 2008.
5. R.Y. Wang. A product perspective on total data quality management. *Commun. ACM*, 41(2):58–65, 1998.
6. A. Even and G. Shankaranarayanan. Utility-driven assessment of data quality. *SIGMIS Database*, 38(2):75–93, 2007.

7. S.P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.
8. K.S. Cook, R. Hardin, and M. Levi. *Cooperation Without Trust?* The Russell Sage Foundation Series on Trust. Russell Sage, 2007.
9. T. Grandison and M. Sloman. Trust management tools for internet applications. In *First international conference on Trust management*, pages 91–107, 2003.
10. J.A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, 2005.
11. C. Castelfranchi and R. Falcone. Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In *Third International Conference on Multi Agent Systems (ICMAS98)*, page 72, 1998.
12. D. Artz and Y. Gil. A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the WWW*, 5(2):58–71, 2007.
13. A. Jø sang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, pages 324–337, 2002.
14. A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. In *HICSS*, page 9, 2000.
15. T. Knap and I. Mlynková. Revealing beliefs influencing trust between members of the czech informatics community. In *Proceedings of the Third international conference on Social informatics*, pages 226–239, 2011.
16. L. Quijano-Sánchez, D. Bridge, B. Díaz-Agudo, and J. Recio-García. A case-based solution to the cold-start problem in group recommenders. In *International Conference for Case-Based Reasoning*, pages 342–356, 2012.
17. T.D. Huynh, N. R. Jennings, and N.R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.
18. R. Neisse, M. Wegdam, M. Van Sinderen, and G. Lenzini. Trust management model and architecture for context-aware service platforms. In *On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS*, pages 1803–1820, 2007.
19. A. Cordier, B. Fuchs, L. Lana De Carvalho, J. Lieber, and A. Mille. Opportunistic acquisition of adaptation knowledge and cases - The IaKa Approach. In *Proceedings of the 9th European Conference on CaseBased Reasoning*, pages 150–164, 2008.
20. D. Leake and M. Whitehead. Case provenance: The value of remembering case sources. In *Proceedings of the 7th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, pages 194–208, 2007.
21. Z. Saaya, B. Smyth, M. Coyle, and P. Briggs. Recommending case bases: Applications in social web search. In *Case-Based Reasoning Research and Development*, pages 274–288. Springer Berlin Heidelberg, 2011.
22. J.B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. The adaptive web. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *Collaborative filtering recommender systems*, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.
23. F. Badra, R. Bendaoud, R. Bentebitel, P.-A. Champin, J. Cojan, A. Cordier, S. Després, S. Jean-Daubias, J. Lieber, T. Meilender, A. Mille, E. Nauer, A. Napoli, and Y. Toussaint. Taaable: Text Mining, Ontology Engineering, and Hierarchical Classification for Textual Case-Based Cooking. In *ECCBR Workshops, Workshop of the First Computer Cooking Contest*, pages 219–228, 2008.
24. E. Gaillard, J. Lieber, and E. Nauer. Adaptation knowledge discovery for cooking using closed itemset extraction. In *The Eighth International Conference on Concept Lattices and their Applications - CLA 2011*, 2011.