User Assistance for Collaborative Knowledge Construction

Pierre-Antoine Champin[1] Amélie Cordier[1] Élise Lavoué[2]
Marie Lefevre[1] Hala Skaf-Molli[3]

[1]Université de Lyon
Université de Lyon 1, LIRIS
UMR5205, F-69622, France
{first.last}@liris.cnrs.fr

[2]Université de Lyon, CNRS
Université Lyon 3, MAGELLAN
LIRIS, UMR5205, France
elise.lavoue@liris.cnrs.fr

[3]LINA, Nantes University
2, rue de la Houssiniere
44322 Nantes, France
hala.skaf@univ-nantes.fr

ABSTRACT

In this paper, we study tools for providing assistance to users in distributed spaces. More precisely, we focus on the activity of collaborative construction of knowledge, supported by a network of distributed semantic wikis. Assisting the users in such an activity is made necessary mainly by two factors: the inherent complexity of the tools supporting that activity, and the collaborative nature of the activity, involving many interactions between users. In this paper we focus on the second aspect. For this, we propose to build an assistance tool based on users interaction traces. This tool will provide a contextualized assistance by leveraging the valuable knowledge contained in traces. We discuss the issue of assistance in our context and we show the different types of assistance that we intend to provide through three scenarios. We highlight research questions raised by this preliminary study.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems*; I.2 [**Computing Methodologies**]: Artificial Intelligence—*Systems*

Keywords

Distributed semantic wikis, User assistance, Traces

1. INTRODUCTION

The Social Semantic Web [14] generates a huge amount of data, produced by human-machine collaboration. Humans participate by contributing structured and unstructured data, while machines perform computation to ensure consistency and to infer new data. Data are continuously updated by humans and machines.

Semantic wikis [17] are very suitable social semantic spaces to support man-machine collaboration. They allow to mix structured, machine-readable data and less structured information aimed at humans. They are very successful for collaboratively producing semantically annotated documents. However, most of them are still missing some important features of collaborative tools: they do not support offline work or multi-synchronous editing [31]. Multi-synchronous collaboration [10] allows involved people to work in isolation on their private copies of the shared data, possibly diverging. Then, in order to converge, they synchronize their copies. Distributed semantic wikis such as DSMW [27] were proposed to support multi-synchronous collaboration in semantic wikis.

This increased flexibility comes with a price, though. It is not always easy for users to anticipate the consequences (in the computations performed by the machine) of their own semantic annotations. In a collaborative context, this is even more sensitive, as one's own modifications may impact others. It is important then to enable users to choose which of the modifications made by others they are ready to accept. The success of distributed semantic wikis therefore relies on the availability of powerful assistants that will help the users to harness their powerful functionalities. Those assistants can act at two levels: helping the users in using the tools, and helping them in interacting with each other in order to reach a consensus. This position paper focuses on the second aspect, and aims at identifying the research questions that need to be addressed in order to provide such assistants.

In the next section, we motivate this work by presenting the context of the Kolflow project in which it was initiated. Then we present the state of the art in the domain of assistance in Section 3. Section 4 then explores three use cases of our motivating example, and identifies the different kinds of assistance needed, and how they can be provided. Finally, we conclude and discuss the following steps of this work.

2. MOTIVATING EXAMPLE

This work is done in the Kolflow[1] project. Kolflow aims at building a collaborative semantic space where humans and smart artifical agents can collaborate to build knowledge. The collaborative space is built over a network of wikis. More precisely, we use specific wiki engines called DSMW (Distributed Semantic Media Wiki). DSMW are built over the well known MediaWiki architecture. They embed the Semantic Media Wiki extension, and they implement mechanisms to support communication between wikis in a distributed architecture. DSMW enable multi-synchronous collaborative activities.

In DSMW, semantic wiki pages can be replicated over the network of interconnected semantic wiki servers. Changes issued on one server are local but they can be published to other servers. Remote servers can subscribe to these changes, pull them and integrate them to their local pages. Changes propagation remains under the control of the users and is supported by a *publish-suscribe* mechanism (*i.e.* users

[1]http://kolflow.univ-nantes.fr

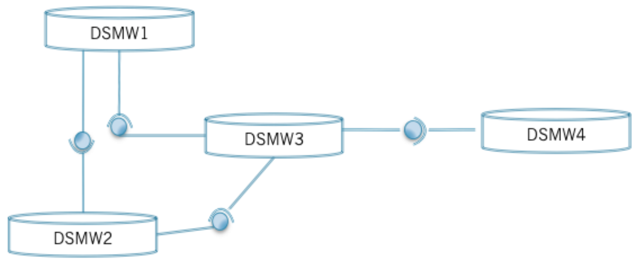


Figure 1: DSMW deployment Scenario

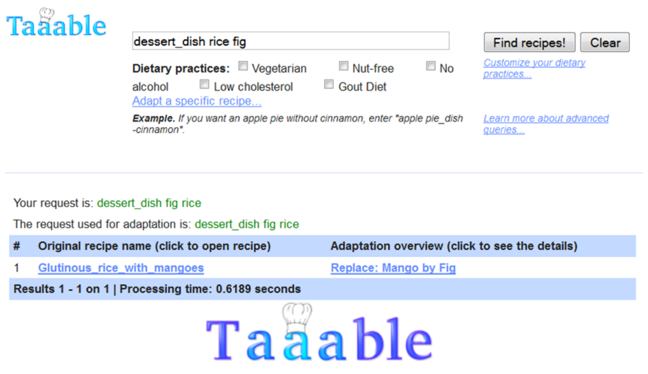
can perform push and pull operations). Concurrent changes on a page issued by different servers are handled by an automatic merge procedure. DSMW uses the Logoot algorithm to synchronize concurrent changes [27]. Logoot guarantees the consistency of the shared pages.

Figure 1 illustrates the basic principles of DSMW and introduces the graphical notation for push and pulls. When a user performs changes on a wiki, he can push them on the network. Users of other wikis can pull these changes on demand. In this example, changes made on DSMW3 have been pushed, and the manager of DSMW4 pulls these modifications.

In current implementations of DSMW, merging of changes is performed by an automatic synchronization algorithm. If a problem occurs during the merging, the operation is aborted and an error is thrown. There is no way that allows the user to correct the problem that caused the error. Therefore, it is hard for users to take into consideration the error feedback they receive. Users must manually manage the problem before attempting a new merge of their resources. This limitation makes the management of wiki content difficult for users. The problem becomes even more critical when viewed on a wider scale, *i.e.* in a distributed environment with many users and communities.

The objective of the work presented in this paper is to build an assistance tool that could help users better manage the merging of resources in a distributed context. To illustrate the importance of this issue, we present the example of WikiTaaable, a semantic wiki used for the distributed knowledge management application Taaable [1].

Taaable is a web-based application that solves cooking problems. When a user asks for "a dessert with rice and figs" to Taaable, the system returns dessert recipes containing rice and figs. If Taaable does not have this kind of recipe in its cookbook, it builds one by adapting an existing recipe. For example, Taaable can retrieve a dessert recipe with rice and mangoes, and recommend the user to replace mangoes by figs to obtain a recipe with rice and figs. For this, Taaable relies not only on its cookbook, but also on a set of additional knowledge. All the knowledge used by the reasoning engine (recipes, domain knowledge, and recipe adaptation knowledge) is represented in a Semantic Wiki called WikiTaaable (more precisely, WikiTaaable uses the DSMW engine). WikiTaaable allows users to visualize the knowledge used in the system and to navigate in the recipe book. The reasoning is performed by a dedicated case-based reasoning engine. The interface of Taaable and WikiTaaable are presented in Figure 2 and 3 respectively.


Taaable
Find recipes!
Clear
Customize your dietary practices...
Adapt a specific recipe...
Learn more about advanced queries...
Taaable

dessert_dish rice fig

Dietary practices: Vegetarian Nut-free No alcohol Low cholesterol Gout Diet

Example. *If you want an apple pie without cinnamon, enter "apple pie_dish -cinnamon".*

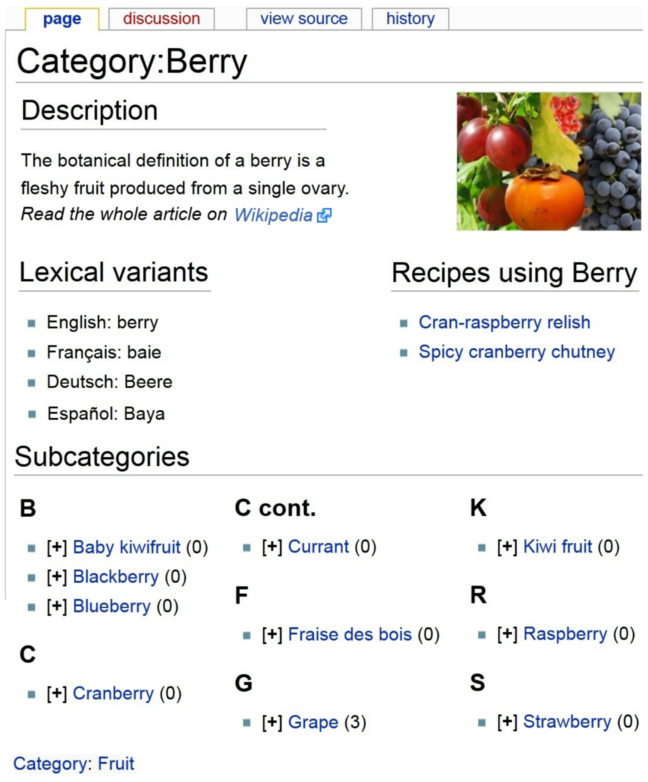
Your request is: dessert_dish fig rice

The request used for adaptation is: dessert_dish fig rice

#	Original recipe name (click to open recipe)	Adaptation overview (click to see the details)
1	Glutinous_rice_with_mangoes	Replace: Mango by Fig

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

Figure 2: Taaable



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
- [+] Baby kiwifruit (0)
- [+] Blackberry (0)
- [+] Blueberry (0)

C
- [+] Cranberry (0)

C cont.
- [+] Currant (0)

F
- [+] Fraise des bois (0)

G
- [+] Grape (3)

K
- [+] Kiwi fruit (0)

R
- [+] Raspberry (0)

S
- [+] Strawberry (0)

Category: Fruit

Figure 3: WikiTaaable

TAAABLE designers have chosen to use a Semantic Wiki to represent TAAABLE knowledge for several reasons:

- A semantic wiki allows users to view easily all the knowledge used by the system (recipes and cooking knowledge).
- A semantic wiki is easy to edit. As a consequence, users can modify, update, or enrich knowledge represented in the system through a quite simple user interface, as if they were editing a classical wiki page.
- In a semantic wiki, knowledge is already formalized. Its usage in a reasoning engine is straightforward.

Historically, WIKITAAABLE was set up by TAAABLE designers in order to simplify the knowledge construction workflow. By using a single tool to build, modify and enrich the knowledge used in the reasoning process, they were able to immediately see the impact of their modifications on the results produced by the reasoning engine. Although the introduction of WIKITAAABLE in the TAAABLE project constitutes a major improvement, significant problems remain:

- The user interface of WIKITAAABLE is not easy to handle, especially for novice users.
- As aforementioned, the knowledge represented in WIKITAAABLE is immediately available to the reasoning engine. Formalization errors (spelling mistakes, categorization problems, etc.) are not checked for the moment. Thus a formalization error may lead to reasoning problems, and the causes of such reasoning problems may be hard to identify.
- For the moment, WIKITAAABLE operates according to a simple centralized architecture. However, the next step is to scale the approach and to implement a distributed version of the system using DSMW. Then, the collaborative dimension will add complexity to the user task.

The first two problems are out of the scope of this paper. However, they are addressed in the Kolflow project. In this paper, we focus on the third problem. How to assist users with the complex task of merging knowledge bases in a collaborative context? In the remaining of the paper, all the use cases will be presented with WIKITAAABLE.

3. STATE OF THE ART

The concept of assistance in the computing domain has been studied extensively over the years. Unfortunately, this concept is often misunderstood. The main reason it that the idea of assistance is too often assimilated to that of this small wizard always popping up at the wrong time and that does not help. In this section we clarify what we mean by assistance.

We can provide help to end users of computer system in a lot of different ways. Assistance is a form of help that we can provide. In this section, we start by presenting a the definition of help. Then, we focus on the notion of assistance. We present the various properties of the assistance. Last, we go one step further by discussing the notion of intelligent assistance, such as we want to implement in Kolflow.

3.1 The concept of the help

From a cognitive science perspective, the concept of help can be defined as an asymmetrical and instrumented relationship, between a human performing and action (desired, suggested or imposed) with special modalities of realization (which can be ignored or forgotten) and a technology supposed to make explicit the modalities, in such a way that they are appropriated by the person seeking help [12].

Gapenne [12] classify four modalities for this “human / technology” coupling:

- substitution: when the technology supports independently the totality or a part of a task;
- supplementation: when the use of the technology increase the possibilities of action for the user;
- assistance: when the technology is not crucial for the main activity, but facilitates or improves the use of the main tool;
- support: when the technology allows to support the appropriation and the use of a new schema by the human.

3.2 Properties of assistance

We now focus on assistance. According to [11], the artifacts performing assistance can be distinguished according to several criteria:

- advisor system vs assistant systems [30]. In this distinction, the advisors provide information, offers solutions, but are not directly involved in the task. Conversely, the assistants are dedicated to the execution of repetitive tasks.
- conversational systems vs. autonomous systems [18, 30]. Conversational systems require the user to express a question or query, which is associated with a logical expression. Autonomous systems operate in the background and can proactively provide suggestions.
- the ability of the system to improve itself [19, 36].

When defining an assistance system, several dimensions must be considered. These dimensions relate to two separate problems: the presentation of assistance to the user and the way to define assistance algorithms [28].

The presentation of the assistance may be differentiated by the following characteristics [28].

- When to assist: If every click by the user indicates a potential action, the question arises of when the user should be assisted. Assistance can be generated proactively (*i.e.*, before an action), during actions, or after actions. Assistance can be provided on user demand or on system request.
- How to assist: As modern computers often represent multi-media work environments, the form of media used by the assistance can be differentiated. Currently, assistance can be presented textually, visually, acoustically, or as a video.
- Where to assist: The information offered by the assistance system might be wrapped in tooltips, pop-ups, tables, specific sound effects, blinking effects, sidebars

of a document, or specific marked spaces. Furthermore, it can be presented within the active tool, a specific third-party tool, or in the operating system itself.

- Why to assist: Assistance may be proposed for many reasons, *e.g.* during the identification of a lack of user's competence, when the task is complex, when the new feature is available, etc.

Similarly, for the definition of assistance algorithms, several features are considered [28] : assistance for whom? Assistance about what? Assistance in which process? Assistance in which tool environment? The various properties of assistance show that, in order to provide contextual assistance, it is necessary to have at least information about the task and the tool at hand, as well as about the skills and preferences of the user.

Kolflow implements an approach based on a network of Wikis. Therfore, we need to implement an assistant suitable for web-based applications. Assistance to end user on the Web may include Web form input operations [7, 41]; adaptive presentation of content [40]; navigation within a website [29, 34, 35]; information or website retrieval [8, 20, 21]. This assistance can be client-side or server-side, depending on user profiles; the history of user actions (previous input, logs, more complex traces); and the content of web pages.

3.3 Towards a trace-based intelligent assistant?

Given the "human/technology" coupling, the combinatory of unpredictable situations users can face is huge. This makes it really difficult to define any *a priori* assistance strategy [23]. Therefore, to develop assistance tools able to adapt themselves to changing needs of users, we must implement "intelligent assistance strategies". Intelligent assistants rely on assistance knowledge that they leverage to perform specific assistance tasks. In order to ensure that intelligent assistants are able to adapt themselves to changing situations and evolving needs of users, we must provide them with the ability to acquire additional knowledge on the fly [5].

Intelligent assistants can be likened to some adaptive tools. Adaptive tools are designed to adapt themselves to users [39]. These tools are able to automatically change their characteristics depending on the needs of users [24]. These tools are particularly relevant in contexts where users need to quickly appropriate environments. Adaptive tools exploit the knowledge they have about the user to adapt their behavior. They also use knowledge of the domain and the application in order to make inferences and identify the elements of the application that can be suited to the user. Thus, adaptive tools focus on how the user interacts with the system and how the interfaces can be adapted to facilitate these interactions. When defining an intelligent assistant, we can draw inspiration from many principles implemented in adaptive tools.

Trace-Based Reasoning (TBR) [22] is a resoning principle inspired from Case-Based Reasoning. It proposes to use traces as a source of knowledge that can be advantageously used in a dynamic reasoning process. In [6], the autors argue that TBR makes it possible to overstep the limits of traditional assistance approaches by developing tools able to adapt themselves to user needs as well as context changes[3, 25].

In the TBR paradigm, traces of the interactions between the user and the application are collected and stored for future reuse. These traces are used as knowledge containers that keep experiences "in context". Reasoning mechanisms extract from the trace the necessary knowledge. As such, TBR will naturally follow the changes in the way the user interacts with the application, helping answering the problem of adaptation [6].

However, when implementing TBR for user assistance, a new problem arises. The question is: how to get from traces (digital record of observations) to knowledge that makes sense for a human user? This question raises the issue of enabling the co-construction of meaning between users and systems [33], a necessary step in the process of providing a trace-based assistance. Co-construction of meaning entails a negotiation process between agents (here, humans and smart agents). The negotiation of meaning between agents is an important topic in AI research. In 2000, Steels proposed a survey of this area [32]: the emergence of a shared language between two agents (specifically, a grammar) is possible through language games allowing interaction between the two agents.

These principles were re-used by Stuber to allow a user to negotiate, via a graphical interface, meaning with his assistant [33]. This interface allows the user to manipulate symbolic interpretations of the relevant parts of his traces of interaction with the system. The interface helps to negotiate the common meaning of the symbols to lead to a consensus between machine and human. The agreement of shared meaning between human and machine allows the assistant to be more effective.

Enabling co-construction of meaning is a required step in the development of our trace-based assistant. In addition, it must be noted that this question will be studied more widely within the context of the Kolflow project. Kolflow aims at facilitating the construction of knowledge shared by humans and smart agents. Obviously, this is also a problem of negotiation of meaning between agents. It must be noted that such negotiation between humans can be seen as a collective decision-making [2]: no attempt is the best solution but the solution that suits the majority. Negotiation is an unpredictable collaborative mechanism and the mediating of this process has also been the subject of much research [37, 38]. These remarks also apply to negotiation of meaning between humans and smart agents. This enhances the need to provide intelligent and adaptive assistants capable of supporting users in this complicated process.

4. USE CASES

In this section, we explore three use cases of growing complexity, involving users collaborating by means of WikiTaaable. For each one, we identify the problems encountered by end users and we explain what assistance could be offered.

4.1 Personal fusion

In this first use case, Charles and David both maintain a personal instance of WikiTaaable with their own recipes and the corresponding ingredient ontologies. Charles wishes to augment his own wiki with recipes from David's (see Fig-

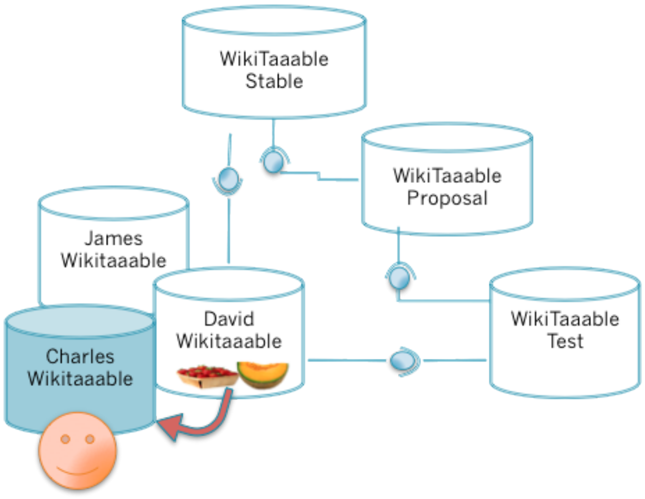


Figure 4: Personal fusion

ure 4). He may encounter two kinds of problems that would require an assistant to support him in this task:

- selecting, from David's recipes, which ones he wants to import[2],
- adapting the resulting ingredient ontology if inconsistencies arose from the fusion.

For the first problem, it is important to recall that the imported recipes will have an impact on the overall behaviour of Charles' system. First, because they draw on parts of David's ingredient ontology, which will also be imported in Charles' ontology. Second, because the imported recipes will become available to the reasoning engine, hence changing the set of recipes that can be retrieved to answer a user query.

For the second problem, two kinds of inconsistencies have to be considered: formal and informal ones. Formal (or logic) inconsistencies are easy to detect, as reasoning algorithms are dedicated to that task. Fixing them is not as easy, but is not the purpose of this paper[3]. We are more concerned with informal inconsistencies, *i.e.* problems that are not captured by the formal semantics of the ontology, but nevertheless need fixing from the user's point of view. People may for example disagree about eggs being a vegetarian ingredient.

Different knowledge sources can be used to assist Charles in solving those problems. We describe them from the easiest to exploit to the most challenging:

- an explicit declaration of Charles' constraints: for example, Charles may have explicitly stated to the assistant: "exclude all recipies with eggs". While this is the most obvious way to efficiently assist Charles, it is not always feasible. Charles may not have enough insight to elicit such knowledge, nor enough technical skill to express it formally.

[2]Note that the current implementation of DSMW does not provide fine grained selection of what is to be imported. This functionality is however planned for the future and is taken for granted in our assistance scenarios.
[3]It is addressed by another part of the Kolflow project.

- traces from previous imports: the assistant may spot recurring patterns (*e.g.* Charles already excluded all recipes containing eggs when importing recipes from James' wiki), and try to reproduce them.
- traces of Charles' use of WikiTaaable: the assistant may evaluate the impact of the imported elements (recipes and ingredients) based on what in his wiki Charles actually uses, and in which way he uses it. He may for example already have some recipes with eggs, but never use them. This again can be used as a clue to prevent importing more recipes of this kind.
- traces of the reasoning engine: beyond Charles' explicit use of WikiTaaable, the assistant can benefit from information about what elements are used by the reasoning engine, especially during recipe adaptation. A change in the ontology may make a frequent adaptation impossible, or on the contrary reinforce adaptations that have proven unsuccessful. For example, the assistant could notice that the results provided by the reasoning engine are always refused by Charles when the reasoning process involved eggs as ingredients.
- traces shared by David: Charles' traces are not the only one that the assistant could use. If David chooses to share parts of his own traces, those can be used as well to compare his usage with Charles', and help decide which parts of David's wiki are worth importing.

Those different sources of knowledge can be used to provide many different kinds of assistance discussed in Section 3. The assistant may pro-actively select for import the most relevant subset of what David is exporting. It may otherwise ask Charles questions about his goals and guide him through the import process. Finally, it may simply react to Charles' decisions, commenting them or advising the next action.

Another dimension is to make the decisions and suggestions of the assistant intellegible to the users, by carefully setting the amount and forms of explanations provided by the assistant. Some users may not be familiar with formal ontology descriptions and the associated reasoning mechanisms, so they will prefer simplified description, while skilled users may prefer a precise account of the decision mechanism. The good thing about using trace-based reasoning for assistance is that it provides an easy way to explain its conclusions by linking them back to actual past experiences. But the assistant could even go a step further by proposing the user to validate a rule inferred from the traces, in order to improve both its performance and the intelligibility of its decision. For example, the assistant could ask Charles: *I notice that you never import recipes containing eggs; shall I add this to your preferences as "exclude any recipe with eggs"?* This has the advantage of creating knowledge of the first kind from the list above, while sparing the users the trouble of eliciting it themselves.

Those kinds of assistance are of course not exclusive, nor are they to be fixed once and for all. The assistant can decide which one to use based on different factors such as the confidence it has in its own prediction, the amount of data available for import, the user's expertise and preferences.

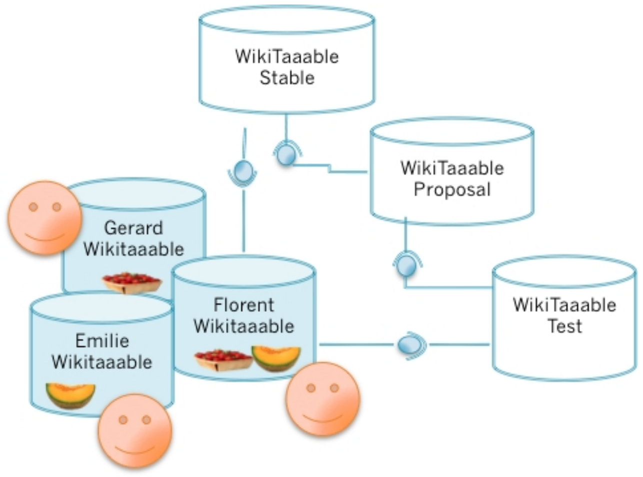


Figure 5: Collaborative fusion

4.2 Collaborative fusion

In this second use case, we are interested in assisting a group of people in building a stable version of their common knowledge in a common instance of WikiTaaable. Note that the goal is not for them to collaboratively edit the same recipe (for that, they would need a synchronous tool) but rather to gather several users' cooking recipes (and their corresponding ingredient ontologies) to build a common set of recipes (and the corresponding common ontology).

A *collaborative* context [9] is defined by the fact that the participants:

- know each other,
- have a well defined task to carry out,
- can negotiate with the others due to the small number of people,
- have the same status (level of trust, authority, etc.).

For instance, Émile, Florent and Gérard each have their own instance of WikiTaaable on their distributed wiki (see Figure 5). As they are lovers of chocolate recipes, they want to gather all their recipes related to chocolate. The issue to overcome here is to reach an agreement on chocolate recipes (and the associated ingredient ontology) to produce a stable instance of WikiTaaable dedicated to chocolate recipes that Émile, Florent and Gérard will be able to share online with others. The problem here is that these users have probably not the same knowledge, neither in their ingredient ontology nor in their recipes. Some of these differences will be compatible or automatically solved, whereas others will induce conflicts that can only be solved by negotiating and finding an agreement among the three users.

Émile, Florent and Gérard will of course encounter the same kind of problems as Charles did in the first use case; the same kind of assistant will therefore be needed. However, the collaborative situation brings other difficulties.

Mainly, the conflicts due to the merging will not appear in each user's wiki, but in the common one. Hence, the cause of the conflicts that may arise will be more complex to spot. Indeed, for Charles, conflicts were necessarily David's fault, while in this case, they can result from non-trivial interaction between the contribution of three (or more) users. It will therefore be necessary to present the users with the *history* of the different resources (recipes and concepts of the ingredients ontology), and assist them in discovering in that history when and how the conflicts arose (and then of course, to help them solve them).

Another problem is that a large part of the negotiation between the users will happen outside the wiki, through different mediums (e-mail, online chat, phone...). While the content of those negotiations might be valuable to document the decision and inform assistants for further decisions, it is not easy or even technically possible to keep track of it.

There are several research issues related to the development of an assistance for this use case.

- How to present the history of the resources to the users involved in the building and negotiation of a set of recipes? But a resource can have a complex history, involving several users, and intermingled with the history of other resources. For example, an ingredient created by Émile may be modified by Florent in order to fit a recipe from Gérard. So it is important to keep all the traces of use of WikiTaaable related to every resource, so as to be able to present them to the users in a contextualized way.
- Which data produced during the negotiation should be added to the history of the resource? It is easy to record all the pull/push operations that occurred during the negotiation. However, the decisions (resources added or deleted) are probably taken outside the wiki. So we can imagine to keep traces of these exchanges if the communication tools used allow it. Another way is to interpret the decisions that were taken apart from the wiki, based on the interaction traces on the wiki before and after the decisions.
- How to integrate all the knowledge produced during the negotiation so as to document a decision? It implies to be able to retrieve the order of all the knowledge and to place them in their context. For example, it should be relevant to integrate into the push/pull traces all the discussions that occurred between these operations. The discussions could have taken place in a separate tool or in the discussion page associated to each resource of the wiki (ingredient and recipe).

4.3 Community fusion

Our third use case focuses on a virtual community of interest. A virtual community emerges in an online environment and gather people who have common interests and/or goals, and so develop social relationships by exchanging information and sharing their knowledge [13, 16, 26]. Compared to the collaborative context described above, a community of interest is defined as a large amount of people who:

- do not necessarily know each other,
- have a common interest,
- have different status (different level of trust, authority, etc.) [4, 15].

Compared to the previous use case, the number of participants can now be much higher, while their goals are much

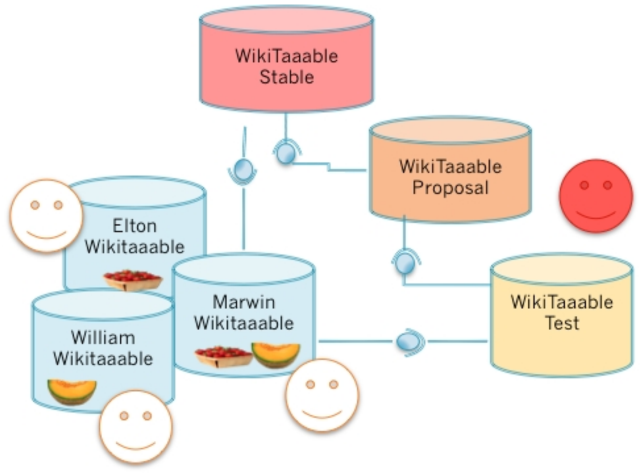


Figure 6: Community fusion

less converging (they merely *share a common interest*). It follows that, unlike in the previous use case, those people can usually not negotiate with each other individually; a moderator is responsible of making decisions, contacting the members, etc.

For instance, Hector is the moderator of a community of people interested in chocolate recipes, each of them contributing to their own instance of WikiTaaable (see Figure 6 where Hector is represented by the red face). Hector is in charge of integrating all the users' suggestions in a stable instance of WikiTaaable. While Hector is ultimately making the decisions alone, his situation is very different from Charles' in section 4.1, with respect to the amount of data to merge, and with respect to Hector's need to negotiate with the members of the community (or help them negotiate with each other).

As a moderator, Hector will typically use a dashboard to have a synthetic view of the community activity and manage the numerous changes pushed by the users. He has to decide which knowledge to validate, according to the information that he has on this knowledge. If he is not able to decide by himself, he has to identify which users to contact in order to find a solution to solve the conflict. A dedicated assistant can greatly help him in those tasks.

This third use case is concerned by the research issues developed in the previous section. There are also specific research issues.

- Which information to present to the moderator on a community management dashboard? The moderator needs warnings when there are conflicts with new knowledge imported in the wiki. In that case, the moderator should be able to access an analytical view with several kinds of information. For example, he should know who has imported which resource, how many people agree on it, etc. The dashboard should also reflect the way the community works. For example, the moderator could detect the most active users, the users that are "expert" on a subject.

- How to determine the level of trust in a member of the community? There are several possibilities. People can declare their level of expertise in their profile. Other people can vote or rate other members. This rating can be explicit, or based on use traces (*e.g.* based on the fact that many people import from that user's wiki).

- In case of a conflict, which knowledge to validate? How to determine the level of trust in a piece of knowledge? For example, we can state that the majority is right. Another possibility is to trust the people deemed "expert" in their sub-domain. But what if the majority and the expert disagree? When the information is insufficient for the moderator to take a decision, it is necessary to assist him in putting the appropriate users in contact.

5. CONCLUSION

In this position paper, we have highlighted the importance of assisting users, in their collaboration to build knowledge using a distributed semantic wiki. We have illustrated the needs for assistance in three use cases, ranging from interpersonal interaction to communities of practice. We have identified the challenges raised by each of these use cases, what kind of assistant can help the user to overcome those challenges, and the sources of knowledge that such assistants will need to tap. In particular, we have shown the value of interaction traces for that purpose. This outlines a research agenda that will be followed in the course of the Kolflow project.

While those use cases are applied to WikiTaaable, they straightforwardly apply to any task involving a distributed wiki. We also trust that they can apply to other aspects of the social semantic web, as soon as they emphasize collaboration and consensus building between users.

6. ACKNOWLEDGEMENT

This work is supported by the French National Research agency (ANR) through the KolFlow project (code: ANR-10-CONTINT-025), part of the CONTINT research program. More information about Kolflow is available on the project website: http://kolflow.univ-nantes.fr/

7. REFERENCES

[1] F. Badra, J. Cojan, A. Cordier, J. Lieber, T. Meilender, A. Mille, P. Molli, E. Nauer, A. Napoli, H. Skaf-Molli, and Y. Toussaint. Knowledge acquisition and discovery for the textual case-based cooking system WIKITAAABLE. In S. J. Delany, editor, *8th International Conference on Case-Based Reasoning - ICCBR 2009, Workshop Proceedings*, pages 249–258, Seattle, United States, July 2009.

[2] C. Brassac and P. Fixmer. La décision collective comme processus de construction de sens. In C. Bonardi, N. Grégori, J.-Y. Menard, and N. Roussiau, editors, *Psychologie sociale appliquée. Emploi, travail, ressources humaines*, pages 111–118. InPress, 2004.

[3] P.-A. Champin. ARDECO: an assistant for experience reuse in CAD. In B. Fuchs and A. Mille, editors, *From Structured Cases to Unstructured Problem Solving Episodes For Experience-Based Assistance (Workshop 5 of ICCBR'03)*, June 2003.

[4] C. Chiu, M. Hsu, and E. Wang. Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories. *Decision Support Systems*, 42(3):1872–1888, Dec. 2006.

[5] A. Cordier, M. Lefevre, S. Jean-Daubias, and N. Guin. Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques, enjeux et étude de cas. In S. Despres, editor, *Acte des 21èmes Journées Francophones d'Ingénierie des Connaissances*, pages 119–131, France, 2010. Ecole des Mines d'Alès.

[6] A. Cordier, B. Mascret, and A. Mille. Extending Case-Based Reasoning with Traces. In *Grand Challenges for reasoning from experiences, Workshop at IJCAI'09*, July 2009.

[7] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification - W3C Recommendation 16 April 2002. `http://www.w3.org/TR/P3P/`.

[8] E. Desmontils and C. Jacquin. Indexing a web site with a terminology oriented ontology. In *The Emerging Semantic Web*, pages 181–197. IOS Press, 2002.

[9] P. Dillenbourg. What do you mean by collaborative learning? In *P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches*, pages 1–19. Elsevier, Oxford, 1999.

[10] P. Dourish. The parting of the ways: Divergence, data management and collaborative work. In *4th European Conference on Computer Supported Cooperative Work*, 1995.

[11] E. Egyed-Zsigmond. *Gestion des Connaissances dans une base de documents multimedias.* These de doctorat en informatique, Institut National des Sciences Appliquees de Lyon, 2003.

[12] O. Gapenne, C. Lenay, and D. Boullier. Defining categories of the human/technology coupling : theoretical and methodological issues. In *Adjunct Proceedings of the 7th ERCIM Workshop on User Interface for All*, 2002.

[13] E. Garrot, S. George, and P. Prevot. Supporting a virtual community of tutors in experience capitalizing. *International Journal of Web Based Communities*, 5(3):407–427, 2009.

[14] T. Gruber. Collective knowledge systems: Where the social web meets the semantic web. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 2008.

[15] S. Hung, C. Chen, and M. J. Lin. Fostering the determinants of knowledge sharing in professional virtual communities. *Computers in Human Behavior*, 25(4):929–939, 2009.

[16] J. Koh and Y. Kim. Knowledge sharing in virtual communities: an e-business perspective. *Expert Systems with Applications*, 26(2):155–166, 2004.

[17] M. Krötzsch, D. Vrandecic, M. Völkel, H. Haller, and R. Studer. Semantic wikipedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4):251–261, 2007.

[18] H. Lieberman. Autonomous interface agents. In *ACM Conference on Human-Computer Interface (CHI-97)*, March 1997.

[19] H. Lieberman and D. Maulsby. Instructible agents : Software that just keeps getting better. *IBM Systems Journal*, 35(3/4):539 – 556, 1996.

[20] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering. In *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, 2001.

[21] S. E. Middleton, H. Alani, N. R. Shadbolt, and D. C. D. Roure. Exploiting synergy between ontologies and recommender systems. In *Semantic Web Workshop at WWW2002*, May 2002.

[22] A. Mille. From case-based reasoning to traces-based reasoning. *Annual Reviews in Control*, 30(2):223–232, Oct. 2006. Journal of IFAC.

[23] A. Mille, M. Caplat, and M. Philippon. Faciliter les activites des utilisateurs d'environnements informatiques : quoi, quand, comment ? *INTELLECTICA*, 2(44):121–143, Dec. 2006. Revue de l'Association pour la Recherche Cognitive Publiee avec le concours du CNRS.

[24] R. Oppermann. Adaptively supported adaptability. *Int. J. Hum.-Comput. Stud.*, 40:455–472, March 1994.

[25] M. Philippon, A. Mille, and G. Caplat. Aide a l'utilisateur: savoir quand intervenir. In *IHM*, volume 264 of *ACM International Conference Proceeding Series*, pages 155–162. ACM, 2005.

[26] J. Preece. *Online Communities: Designing Usability and Supporting Socialbilty.* John Wiley & Sons, Inc., New-York, USA, 2000.

[27] C. Rahhal, H. Skaf-Molli, P. Molli, and S. Weiss. Multi-synchronous collaborative semantic wikis. In *10th International Conference on Web Information Systems Engineering - WISE '09*, volume 5802 of *LNCS*, pages 115–129. Springer, October 2009.

[28] J. Rech, E. Ras, and B. Decker. Intelligent assistance in german software development: A survey. *IEEE Softw.*, 24:72–79, July 2007.

[29] B. Richard and P. Tchounikine. Enhancing the adaptivity of an existing website with an epiphyte recommender system. *The New Review of Hypermedia and Multimedia*, 10(1):31–52, 2004.

[30] T. Selker. Coach : A teaching agent that learns. *ACM Communications*, 37(7):92 – 99, 1999.

[31] H. Skaf-Molli, G. Canals, and P. Molli. DSMW: Distributed Semantic MediaWiki. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, editors, *ESWC (2)*, volume 6089 of *Lecture Notes in Computer Science*, pages 426–430. Springer, 2010.

[32] L. Steels. Language as a complex adaptive system. In M. Schoenauer, editor, *Proceedings of PPSN VI*, Lecture Notes in Computer Science, Berlin, Germany, September 2000. Springer-Verlag.

[33] A. Stuber, S. Hassas, and A. Mille. Language games for meaning negotiation between human and computer agents. *Engineering Societies in the Agents World VI*, pages 275–287, 2006.

[34] P. Sukaviriya. Dynamic construction of animated help from application context. In *Proceedings of the 1st*

annual ACM SIGGRAPH symposium on User Interface Software, UIST '88, pages 190–202, New York, NY, USA, 1988. ACM.

[35] P. Sukaviriya and J. D. Foley. Coupling a ui framework with automatic generation of context-sensitive animated help. In *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*, UIST '90, pages 152–166, New York, NY, USA, 1990. ACM.

[36] B. Trousse, M. Jaczynski, and R. Kanawati. Une approche fondee sur le raisonnement a partir de cas pour l'aide a la navigation dans un hypermedia. In *Product, Tools and Methods (H2PTM'99)*, 1999.

[37] P. Verillon and P. Rabardel. Cognition and artifacts: A contribution to the study of though in relation to instrumented activity. *European Journal of Psychology of Education*, 10:77–101, 1995. 10.1007/BF03172796.

[38] L.-S. Vygotski. *Myslenie i rec'*. Messidor, 1934.

[39] S. Weibelzahl. *Evaluation of Adaptive Systems.* Ph.d. thesis, University of Trier, 2002.

[40] C. Wiecha, W. Bennett, s. Boies, and J. Gould. Generating highly interactive user interfaces. *SIGCHI Bull.*, 20:277–282, March 1989.

[41] A. Yoshinori, S. Masahide, and N. Amane. Rule-based interactive web forms for supporting end users(special issue on high speed networks and multimedia applications). *Transactions of Information Processing Society of Japan*, 44(3):722–741, 2003-03-15.