Extraction of Adaptation Knowledge from Internet Communities*

Norman Ihle, Alexandre Hanft and Klaus-Dieter Althoff

University of Hildesheim
Institute of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
{ihle|hanft|althoff}@iis.uni-hildesheim.de

Abstract

[1] Acquiring knowledge for adaptation in CBR is an demanding task. This paper describes an approach to make user experiences from an Internet community available for the adaptation. We worked in the cooking domain, where a huge number of Internet users share recipes, opinions on them and experiences with them. Because this is often expressed in informal language, in our approach we did not semantically analyze those posts, but used our already existing knowledge model to find relevant information. We classified the comments to make the extracted and classified items usable as adaptation knowledge. The first results seem promising.

1 Introduction

Adaptation is a central part of the case-based reasoning process model [Kolodner, 1993]. A good adaptation is of very high importance if the case base is restricted to a small number of cases or if the variation of the problems to be solved is very high. Adaptation has not been the most current topic in recent years of CBR research [Greene *et al.*, 2008], but in the last year the research effort increased again [Cojan and Lieber, 2008; Cordier *et al.*, 2008; Leake and Dial, 2008]. Often adaptation means justifying values in a bounded range [Cojan and Lieber, 2008] and is done via rules created and maintained by a domain knowledge or system developer [Leake *et al.*, 1995].

Knowledge acquisition for adaptation (Adaptation Knowledge acquisition: AKA) is a cost intensive task since it is highly domain dependent and the hard-to-get experts are needed for acquiring and maintaining the necessary knowledge. To solve this problem, research on automated adaptation knowledge acquisition has been done, but mainly focused on automated AKA from cases in the case base [Wilke *et al.*, 1996; Hanney and Keane, 1997; d'Aquin *et al.*, 2007].

Besides the case base, the Internet and the Web 2.0 with its user-generated content are a large source of any kind of knowledge and experience. Following the Web 2.0 paradigm of user-interaction people provide their experience, opinions and advice on any kind of topic. Although the people are not necessarily experts in the domain, the hope is that the mass of users will correct mistakes as practiced for example in the Wikipedia project.

[1]This is an extended version of a paper which was accepted at the Workshop "WebCBR: Reasoning from Experiences on the Web" at ICCBR'09

In this paper we present an approach to make knowledge from Internet communities accessible as adaptation knowledge using the domain model that usually exists in structured CBR applications [Bergmann, 2002]. In the first part of the paper the adaptation background inside CBR is introduced before we describe the domain and the existing application we worked with in our approach. After a short introduction of the tool we used, we will explain the approach in detail before we close with results of the evaluation, related work and an outlook.

2 The Cooking Domain

For most people cooking is "everyday" knowledge. Almost everybody has encountered the problem of preparing a meal with a restricted amount of ingredients. This explains why a huge number of people are willing to share their recipes as well as their experience with the preparation of these recipes. Cooking communities on the Internet offer a platform for this. They provide the possibility to share recipes and also the chance to review and comment them. Usually they also offer additional information like cooking hints, background information on groceries or preparation methods. Besides the fact of the existence of active cooking communities, we chose the cooking domain for the investigation on adaptation knowledge because it has some advantages compared to other areas of interest discussed in communities like computer problems for example. First of all, it is relatively easy to describe the (near) complete context of preparing a meal. Hence, it is possible to reconstruct the experience of others by preparing the meal and trying the adaptation suggestions oneself. The context can be described according to the following characteristics:

1. all ingredients can be listed with exact amount and quality
2. ingredients can be obtained in standardized quantities and in comparable quality
3. kitchen machines and tools are available in a standardized manner
4. (in case of a failure) the preparation of a meal can start all over again every time from the same initial situation (except that we have more experience in cooking after each failure).

The latter one is not given in many other domains, for example setting up a computer costs much time and a certain installation situation cannot always be restored. Additionally, cooking and the adaptation of recipes is (some basic understanding presumed) relatively uncritical. In contrast to medical applications it does not endanger human health, except for some rare meals like fugu (pufferfish).

The costs of a failure are low. It is mostly covered by the price of the ingredients plus the preparation time. Cooking is also an appropriate application domain for adaptation, because cooking mastery depends on the variation and creativity, not only on following strictly preparation advices for a meal [This-Benckhard, 2001].

3 CookIIS and Adaptation in CookIIS

CookIIS [Hanft *et al.*, 2008] is a CBR-based recipe search engine that competes in the Computer Cooking Contest (CCC). When the user provides possible ingredients, it searches for suitable recipes in a case base. Doing that it considers ingredients the user does not want or cannot use because of a certain diet. If recipes with unwanted ingredients are retrieved, CookIIS offers adaptation suggestions to replace these ingredients. According to the CCC the set of recipes is restricted. Besides the retrieval and adaptation of recipes CookIIS also offers recipes for a complete three course menu from the given ingredients (and maybe some additional ones).

CookIIS is using a very detailed domain model which is described in [Hanft *et al.*, 2008]. It was created using the empolis:Information Access Suite (e:IAS) [empolis GmbH, 2008], which offers a knowledge modeling tool called Creator and with the Knowledge Server a component to build client-server based applications. It also provides a rule engine for the completion of cases and queries and for the adaptation of cases after the retrieval. Some more technical details are described in [Hanft *et al.*, 2009a].

3.1 Adaptation with the empolis:Information Access Suite

As stated above, the e:IAS offers the possibility to use completion rules which are executed before building the case index or before the retrieval to extend cases or queries with meta-information and adaptation rules, which are executed after the retrieval, to modify retrieved cases. The Creator offers a rule editor to model completion and adaptation rules with an own syntax. The rules follow the classic IF ... THEN ... schema. They have read and write access to all modeled objects and their values, but only adaptation rules have access to the retrieved cases since they are executed after the retrieval. A large amount of predefined functions help to manipulate single values. Both rule types use the same e:IAS specific syntax, which after compilation is stored in the format of the Orenge Rule Markup Language (ORML), an XML-language.

3.2 Case Representation and Similarity for Adaptation in CookIIS

The case representation is based on structured CBR. 11 classes of ingredients (e.g. Vegetables, Fruit, etc.) plus some classes for additional information (e.g. Type of Meal, Tools, etc.) are modeled, which represent about 2000 concepts of the cooking domain. As an example both concepts carrot and cucumber belong to the class Vegetable. The aim of differentiating ingredient classes is that we want to restrict the replacement of ingredients during adaptation to the same class. A case consists of 11 attributes, one for each possible ingredient class. Each attribute of ingredients can have multiple values per recipe (sets). Most concepts of the different classes are ordered in specific taxonomies. These and some custom similarity measures are used to compute the similarity between the query and the cases. Thereby the different attributes have different weights corresponding to their importance for a meal. Additional meta-information like the type of cuisine of a recipe is established during the indexing process of the recipes and also stored in the case.

The approach for adaptation that was first realized in CookIIS is to replace forbidden ingredients (according to a certain diet oder explicitly unwanted) with some similar ingredients of the same class. While executing a query unwanted (forbidden) ingredients are collected in extra attributes. Besides the explicit exclusion, four different methods can be distinguished to handle dietary practices, where more conditions have to be considered [Hanft *et al.*, 2009a]. One of those methods is the same approach as above: ingredients that have to be avoided due to a diet are replaced by similar ones.

Adaptation rules take these forbidden ingredients and check if at least one of them is an ingredient used in the retrieved case (recipe). Then, making use of the taxonomies and a similarity-aware set-function offered by the rule engine, the most similar ingredients to the unwanted one are retrieved and offered as replacement. The functions of the rules are described in detail in [Hanft *et al.*, 2009a]. If no similar ingredient is found that can be used for following the diet, then the suggestion is to just omit that ingredient.

Shortcomings of the Existing Adaptation Approach

Since the used adaptation approach makes use of the modeled taxonomies the results are often inappropriate. The method returns sibling concepts to the unwanted one as well as parent and child concepts. Only the siblings are the ones who are interesting for adaptation, but the others cannot be avoided with the provided rule functions. Also the number of siblings is often too high. For one unwanted ingredient one or two ingredients as an adaptation suggestion would be preferable. A detailed analysis of the problems with the adaptation and the ways to handle it with the e:IAS Rule mechanism is described in [Hanft *et al.*, 2009b].

4 CommunityCook: A System to Extract Adaptation Knowledge from Cooking Communities

In this chapter we will present our approach to extracting adaptation knowledge from a German cooking community. For this purpose we use our existing knowledge model from the CookIIS application and the TextMiner provided by e:IAS to extract ingredients from recipes and comments on those recipes and classify them. One of the classes can then be used as adaptation knowledge.

4.1 Idea behind the Approach

Our idea is to make knowledge from a cooking community accessible for our CookIIS application to have better adaptation suggestions in case a recipe contains an unwanted or forbidden ingredient. We were especially interested in comments that people posted in reply to provided recipes. In these comments users express their opinion on the recipe, before as well as after cooking it. They write about their experience with the preparation process and also tell what they changed while preparing the recipe. Thereby they express their personal adaptation of the recipe and frequently give reasons for this. Since this is written down in natural language text, often using informal language, we had the idea not to semantically analyze what people said, but to just find the occurrences of ingredients in the comment texts and then compare them to the ingredients mentioned in the actual recipe. We propose to classify them into three classes, depending on whether the ingredients mentioned in a comment appear in the recipe or not. The classification idea is described in the following sections.

4.2 Analysis of Example Cooking Communities

In Germany, *chefkoch.de*[2] is a well known cooking community with a large number of active users. So far, over 131'000 recipes have been provided by the users with an even larger amount of comments on them. The users also have the possibility to vote on the recipes, send them to a friend per email or even add pictures of their preparation. Besides the recipes, chefkoch.de features an open discussion board for all kinds of topics on cooking with more than 7.8 million contributions. Their English partner site *cooksunited.co.uk*[3] is unfortunately much smaller with only about 2200 recipes and 3500 posts.

But with *allrecipes.com*[4] a big platform with a huge amount of recipes and over 2.4 millions reviews is available in English. It has representable big localizations for the United States, Canada, the United Kingdom, Germany, France and others. Allrecipes.com explicitly provides variants of an existing recipe. Hence it also seems to be also a good source candidate. Another large cooking German community is *kochbar.de*[5] with over 160'000 recipes. Besides these large communities a number of smaller communities exist in the Web with more or less similar content. For our approach we decided to use a large German community since the recipes and the corresponding comments are presented on one page with a standardized HTML-code template, which makes it easier to crawl the site and extract relevant information items.

4.3 Extraction of Information Items from a Cooking Community

From a large German community we collected about 70'000 recipes with more than 280'000 comments by crawling the site. This way we got one HTML source-code page for each recipe with the corresponding comments. From this source code we extracted the relevant information entities using customized HTML-filters which we built using the HTML Parser tool[6]. For the recipes these entities were primarily the recipe title, needed ingredients and the preparation instructions, but also some additional information on the preparation of the recipe (e.g. estimated time for the preparation, difficulty of the preparation, etc.) and some usage statistics (e.g. a user rating, number of times the recipe has been viewed, stored or printed, etc.). If users commented on the recipe, we extracted the text of the comment, checked if the comment was an answer to another comment and if the comment has been marked as helpful or not. We also remembered the recipe ID of the related recipe. All this information we stored in a database to have an efficient access to it.

In the next step we used the e:IAS and indexed all recipes and all comments into two different case bases using a slightly extended CookIIS knowledge model. One case base consists of the recipes and one of the comments. For each recipe and each comment we extracted the mentioned ingredients and stored them in the case using our knowledge model and the e:IAS TextMiner during the indexing process. Since our knowledge model is bilingual (English and German) we were also able to translate the originally German ingredient names from the comment text into English terms during this process and this way had the same terms in the case bases that we use in our CookIIS application.

[2]http://www.chefkoch.de, last visited 2009-04-22

[3]http://www.cooksunited.co.uk, last visited 2009-04-23

[4]http://allrecipes.com, last visited 2009-04-23

[5]http://www.kochbar.de, last visited 2009-05-22

[6]http://htmlparser.sourceforge.net, last visited 2009-04-18

4.4 Classification of Ingredients

Having built up the two case bases we first retrieved a recipe and then all of the comments belonging to the recipe and compared the ingredients of the recipe with the ingredients mentioned in the comments. We then classified the ingredients mentioned in the comments into the following three categories:

- *New*: ingredients that are mentioned in the comment, but not mentioned in the recipe
- *Old*: ingredients that are mentioned in the comment as well as in the recipe
- *OldAndNew*: two or more ingredients of one class of our knowledge model, of which at least one was mentioned in the recipe and in the comment and at least one other one was only mentioned in the comment, but not in the recipe

We interpret the classification as follows:

- *New*: New ingredients are a variation of the recipe. A new ingredient (for example a spice or an herb) somehow changes the recipe in taste or is a tryout of something different or new.
- *Old*: If an ingredient of a recipe is mentioned in the comment it means that this ingredient is especially liked or disliked (for example the taste of it), that a bigger or smaller amount of this ingredient has been used (or even left out), or it is a question about this ingredient.
- *OldAndNew*: This is either an adaptation (e.g. instead of milk I took cream) or an explanation/specialization (e.g. Gouda is a semi-firm cheese).

For the adaptation the last class is the interesting one. For each ingredient classified as *OldAndNew* we also stored whether it is the new or the old one. We tried to distinguish between adaptation and specialization by looking for hints in the original comment text and by using the taxonomies of our knowledge model. Therefore we tried to find terms in the comment during the text-mining process that confirm if it is an adaptation (e.g. terms like: instead of, alternative, replaced with, ...) and stored those terms in the corresponding case. Additionally we looked in the taxonomy of the ingredient class whether the one ingredient is a child of the other (or the other way around). If an ingredient is a child of the other we interpreted this as specialization or explanation, because one ingredient is a more general concept than the other. This way we could avoid adaptations like: "instead of semi-firm cheese take Gouda".

For the classes *Old* and *New*, which we consider as variations of the recipe, we also tried to find terms in the comment that closer describe the function of the mentioned ingredient. For example, if an ingredient was classified as *Old*, we looked for terms like 'more', 'less' or 'left out'. If the ingredient of the comment is of the supplement class of our CookIIS knowledge model, and the recipe did not contain any supplement, then we took this as a suggestion for a supplement (e.g. bread for a soup recipe).

For each classified ingredient we assigned a specific score, which depends on the following factors:

- the number of ingredients found in the comment text
- whether the comment was marked as helpful or not
- whether a term was found that indicates the classification assigned or not
- whether a term was found that indicates a different classification or not

After assigning the score we aggregated our classification results. We did this in two steps: First we aggregated all classified ingredients of all comments belonging to one recipe. Thereby we counted the number of the same classifications in different comments and added up the score of the same classifications. For instance a specific recipe has 12 comments in which 3 of them mention milk and cream.

Then we aggregated all classifications without regarding the recipe they belong to. In our dataset we found comments with milk and cream belonging to 128 different recipes. This way we could select the most common classifications out of all classifications. Since we are using a CBR tool and have cases, we also checked if similar recipes have the same ingredients with the same classification mentioned in the comments. We did this for each recipe first with a similarity of at least 0.9, then with a similarity of 0.8. If many of the same classified ingredients exist in similar recipes, this supports our results.

4.5 Usage as Adaptation Knowledge

OldAndNew-classified ingredients can be used to generate adaptation suggestions. This can be done in two different ways: independent from the recipe or with regard to the recipe. Considering the fist way, we look in the database table for the ingredient to adapt and use the result where the ingredient that needs to be adapted is categorized as old and appears in the most recipes or has the highest score. It is possible to retrieve two or more adaptation suggestions to be more manifold. Using this approach we got more than 6200 different adaptation suggestions of which we only used the most common (regarding the number of appearances in the comments and the score) per ingredient. Figure 1 shows some of these suggestions, e.g. in the first line a suggestion to replace cream with milk which appears in comments to 128 different recipes.

We integrated this approach into our CookIIS application: at first we look for two adaptation suggestions from CommunityCook. If no suggestions are provided, the set of more general adaptation rules (see section 3.2) determine adaptation suggestions.

5 Evaluation of the Results

A first look at the results of the most common adaptation suggestions is promising. Only the ingredient class "supplement" reveals problems which are due to the fact that too many different ingredients are integrated into this class. This can be changed by further improving the modeling.

The evaluation can be divided into two different parts. At first we checked if our classification and the interpretation correspond to the intentions written in the original comments. This was done manually by comparing the classification results and their interpretation to the original comments and match in most of over 400 tests the classification.

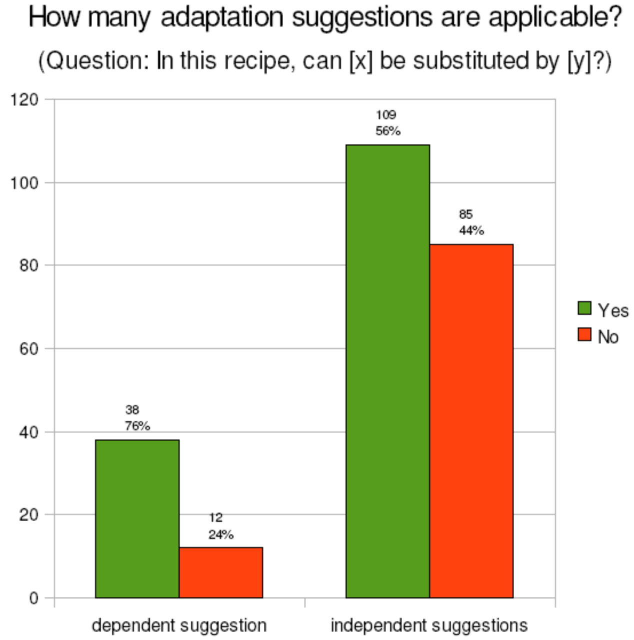

Figure 2: Applicability of dependent and overall independent suggestion

The second evaluation was done on the results of the overall aggregated adaptation suggestions. We examined whether the adaptation suggestions with a high score are good adaptation suggestions for any kind of recipe. Our idea is to take a representative number of recipes and present them with adaptation suggestions to real chefs. These chefs then rate the adaptation suggestions.

Therefore we designed a questionnaire by choosing randomly a recipe and add one adaptation suggestion extracted from comments belonging to that recipe ("dependent") and secondly add two adaptation suggestions without regard of the recipe ("independent") each with two ingredients as replacement suggestion. At the end we present the 50 questionnaires with 50 dependent and 100 pairs of independent ingredients to different chefs, because each chef may have a different opinion.

76% of the dependent and 58% of the independent adaptation suggestions were confirmed as applicable by the chefs (see figure 2). Differentiating the first and second independent suggestion it could be observed that the first one is noteworthy better (see figure 3). Summing up it follows that only by 11 of the 100 independent adaptation suggestions no ingredient can be used as substitution.

In case that the adaptation suggestion was applicable, the chefs should rate it as very good, good and practicable. Here again the dependent suggestions perform better, see figure 4.

6 Related Work

JULIA [Hinrichs, 1992] and CHEF [Hammond, 1986] are early CBR systems giving preparation advice for meals. CHEF is a planning application which builds new recipes in the domain of Szechwan cooking. To satisfy the goals of a request for a new recipe it anticipates and tries to avoid problems. Therefore it stores and retrieves occurred problems and ways of dealing with them. JULIA integrates CBR and constraints for menu design tasks. It uses a large taxonomy of concepts and problem decomposition with fixed decomposition plans. Unlike our approach their

	id integer	ingr_class text	oldingr1 text	oldingr2 text	newingr1 text	newingr2 text	score double precis	specification text	card_recipes integer
1	52	Comment_Ingredient_Milk	cream		milk		79.9583333333	adaptation	128
2	63	Comment_Ingredient_Milk	milk		cream		48.2	adaptation	78
3	254	Comment_Ingredient_Supplement	potatoes		sauce		36.1333333333	adaptation	61
4	238	Comment_Ingredient_Supplement	potatoes		broth		31.025	adaptation	54
5	386	Comment_Ingredient_OilAndfat	margarine		butter		29.4166666666	adaptation	46
6	128	Comment_Ingredient_Meat	bacon		ham		28.175	adaptation	45
7	20	Comment_Ingredient_Supplement	noodle		sauce		27.8166666666	adaptation	48
8	127	Comment_Ingredient_OilAndfat	butter		olive oil		27.0333333333	adaptation	43
9	393	Comment_Ingredient_OilAndfat	butter		margarine		25.55	adaptation	41
10	39	Comment_Ingredient_Vegetable	onion		green onion		21.55	adaptation	34
11	230	Comment_Ingredient_Milk	cream		yogurt		20.35	adaptation	31
12	332	Comment_Ingredient_Milk	creme fraiche		smetana		20.15	adaptation	33
13	1072	Comment_Ingredient_SpiceAndHerb	salt		pepper		18.55	adaptation	32
14	160	Comment_Ingredient_Supplement	rice		broth		17.025	adaptation	29
15	21	Comment_Ingredient_Milk	smetana		sour cream		16	adaptation	25
16	785	Comment_Ingredient_Vegetable	onion		paprika pepper		14.725	adaptation	26
17	57	Comment_Ingredient_Milk	cheese		cream		14.525	adaptation	25
18	60	Comment_Ingredient_Vegetable	onion		leek		14.25	adaptation	23
19	64	Comment_Ingredient_Milk	cream		coconut milk		14.1916666666	adaptation	22
20	73	Comment_Ingredient_Drinks	rum		amaretto		13.9	adaptation	22
21	660	Comment_Ingredient_Milk	cream		cheese		13.85	adaptation	24
22	98	Comment_Ingredient_Meat	ham		bacon		13.85	adaptation	22
23	424	Comment_Ingredient_Milk	yogurt		cream		13.775	adaptation	23
24	249	Comment_Ingredient_Meat	ham		salami		13.65	adaptation	21
25	385	Comment_Ingredient_Minor	honey		maple syrup		13.125	adaptation	21

Figure 1: Some suggestions for adaptation

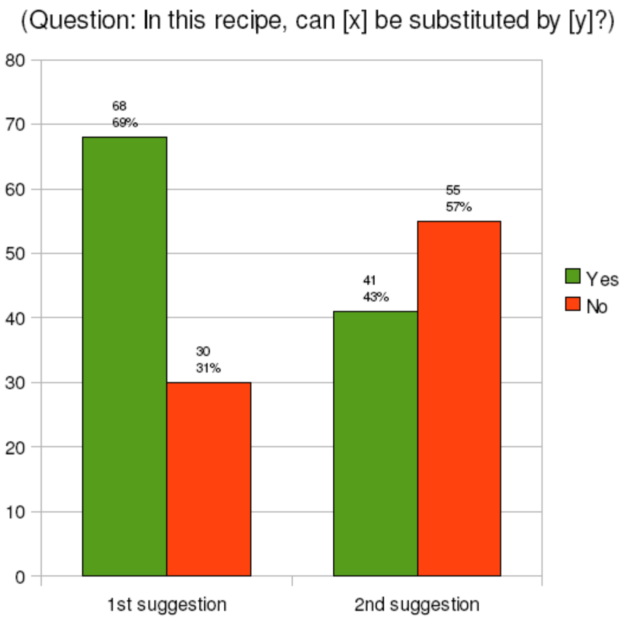

Figure 3: Applicability of the first and second independent suggestion

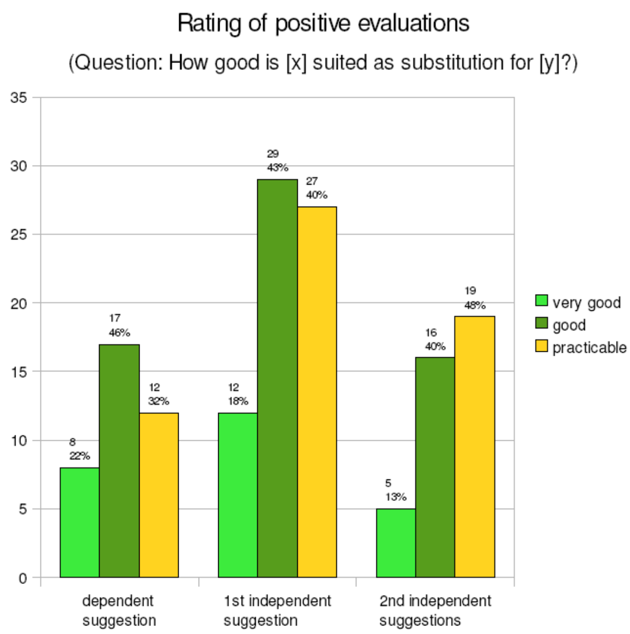

Figure 4: Ratings of applicable adaptation suggestions

knowledge was built by experts and was not captured from communities.

The idea presented here closely relates to the research of Plaza [Plaza, 2008], especially the EDIR cycle, however they concentrate more on gathering cases from web experience. In [Cordier *et al.*, 2008] they use the presented IakA approach for the acquisition of adaptation knowledge (and cases) by asking an oracle, which is described as an "ideal expert", but the presented prototype IakA-NF works (only) for numerical function domains. Furthermore Acquisition of Adaptation Knowledge from cases was done by by [Hanney and Keane, 1997] or with the CABAMAKA System by [d'Aquin *et al.*, 2007].

The procedure of looking at first for concrete adaptation suggestions and apply afterwards, if the first step yields no results, more general rules, was done also by [Leake *et al.*, 1995] with DIAL, which at first attempt to retrieve adaptation cases.

Our approach presented here goes with the vision of Collaborative Multi-Experts Systems (CoMES) [Althoff *et al.*, 2007] and is modelled following the SEASALT architecture [Bach *et al.*, 2007], an instance of CoMES. Mapping this to the CommunityCook System the collection of recipes and comments corresponds to the task of the *Collector Agent*. The further analysis and interpretation match to their role of a *Knowledge Engineer*.

7 Conclusion and Outlook

Adaptation knowledge acquisition is an demanding and expensive task since it needs experts. In this paper we presented an approach to use experience from Internet communities for adaptation knowledge. Our approach is based on the idea of comparing the ingredients mentioned in a recipe to the ones mentioned in the comments that relate to the recipe. From comments which contain ingredients also existing in the recipe and others which are not contained in the recipe the adaptation suggestions are created and aggregated over all comments to 6200 suggestions. The evaluation results are promising and show that adaptation suggestion extracted from the same recipe are more acceptable than the one which are independent and aggregated over all recipes.

The approach described here has a lot of advantages. For finding ingredients we can use our existing CookIIS knowledge model which has the benefit of taking care of synonyms, independence from slang and grammatically deficient language. By using a large number of recipes and comments we hope to balance out wrong classifications. We integrated the extracted adaptation suggestions in our CookIIS application.

In the future we want to be able to use the adaptation suggestions with regard to the recipe they belong to. Therefore we will find similar recipe out of our pool of 70'000 recipes to the one that has to be adapted and consider only comments of these recipes following the principle that similar recipes need similar adaptations.

Following the SEASALT architecture we also want to realize a multi-agent system that continuously monitors the community for new experiences with the recipes and adapts our adaptation knowledge if necessary.

References

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Régis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative multi-expert-systems - realizing knowledge-lines with case factories and distributed learning systems. In Joachim Baumeister and Dietmar Seipel, editors, *KESE*, volume 282 of *CEUR Workshop Proceedings*, 2007.

[Althoff *et al.*, 2008] Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors. *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*, volume 5239 of *LNCS*, Heidelberg, 2008. Springer.

[Bach *et al.*, 2007] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. A domain independent system architecture for sharing experience. In Alexander Hinneburg, editor, *Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement*, pages 296–303, September 2007.

[Bergmann, 2002] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, volume 2432 of *LNAI*. Springer-Verlag, 2002.

[Cojan and Lieber, 2008] Julien Cojan and Jean Lieber. Conservative adaptation in metric spaces. In Althoff et al. [2008], pages 135–149.

[Cordier *et al.*, 2008] Amélie Cordier, Béatrice Fuchs, Léonardo Lana de Carvalho, Jean Lieber, and Alain Mille. Opportunistic acquisition of adaptation knowledge and cases - the iaka approach. In Althoff et al. [2008], pages 150–164.

[d'Aquin *et al.*, 2007] Mathieu d'Aquin, Fadi Badra, Sandrine Lafrogne, Jean Lieber, Amedeo Napoli, and Laszlo Szathmary. Case base mining for adaptation knowledge acquisition. In Manuela M. Veloso, editor, *IJCAI*, pages 750–755. Morgan Kaufmann, 2007.

[empolis GmbH, 2008] empolis GmbH. Technical white paper e:information access suite. Technical report, empolis GmbH, January 2008.

[Greene *et al.*, 2008] Derek Greene, Jill Freyne, Barry Smyth, and Pádraig Cunningham. An analysis of research themes in the cbr conference literature. In Althoff et al. [2008], pages 18–43.

[Hammond, 1986] Kristian J. Hammond. Chef: A model of case-based planning. In *American Association for Artificial Intelligence, AAAI-86, Philadelphia*, pages 267–271, 1986.

[Hanft *et al.*, 2008] Alexandre Hanft, Norman Ihle, Kerstin Bach, Régis Newo, and Jens Mänz. Realising a cbr-based approach for computer cooking contest with e:ias. In Martin Schaaf, editor, *ECCBR Workshops*, pages 249–258, Hildesheim, Berlin, 2008. Tharax Verlag.

[Hanft *et al.*, 2009a] Alexandre Hanft, Norman Ihle, Kerstin Bach, and Régis Newo. Cookiis – competing in the first computer cooking contest. *Künstliche Intelligenz*, 23(1):30–33, 2009.

[Hanft *et al.*, 2009b] Alexandre Hanft, Norman Ihle, and Régis Newo. Refinements for retrieval and adaptation of the cookiis application. In Knut Hinkelmann and Holger Wache, editors, *Wissensmanagement*, volume 145 of *LNI*, pages 139–148. GI, 2009.

[Hanney and Keane, 1997] Kathleen Hanney and Mark T. Keane. The adaptation knowledge bottleneck: How to ease it by learning from cases. In David B. Leake and Enric Plaza, editors, *ICCBR*, volume 1266 of *LNCS*, pages 359–370. Springer, 1997.

[Hinrichs, 1992] Thomas R. Hinrichs. *Problem solving in open worlds*. Lawrence Erlbaum, 1992.

[Kolodner, 1993] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, 1993.

[Leake and Dial, 2008] David B. Leake and Scott A. Dial. Using case provenance to propagate feedback to cases and adaptations. In Althoff et al. [2008], pages 255–268.

[Leake *et al.*, 1995] David B. Leake, Andrew Kinley, and David C. Wilson. Learning to improve case adaptation by introspective reasoning and cbr. In Manuela M. Veloso and Agnar Aamodt, editors, *ICCBR*, volume 1010 of *LNCS*, pages 229–240. Springer, 1995.

[Plaza, 2008] Enric Plaza. Semantics and experience in the future web. In Althoff et al. [2008], pages 44–58. invited talk.

[This-Benckhard, 2001] Herve This-Benckhard. *Rätsel und Geheimnisse der Kochkunst*. Piper, 2001.

[Wilke *et al.*, 1996] Wolfgang Wilke, Ivo Vollrath, Klaus-Dieter Althoff, and Ralph Bergmann. A framework for learning adaptation knowledge based on knowledge light approaches. In *5th German Workshop on CBR*, pages 235–242, 1996.