Case-Based Menu Creation as an Example of Individualized Experience Management

Kerstin Bach[1,2], Pascal Reuss[1], and Klaus-Dieter Althoff[1,2]

[1]Intelligent Information Systems Lab
University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

[2]Competence Center Case-Based Reasoning
German Research Center for Artificial Intelligence (DFKI) GmbH
Trippstadter Straße 122, 67663 Kaiserslautern, Germany
kerstin.bach@dfki.de, pascal.reuss@uni-hildesheim.de, klaus-dieter.althoff@dfki.de

Abstract: This paper presents a Case-Based Reasoning application for individualized experience management. We describe how such intelligent technologies can be used for implementing applications in small and medium sized enterprises by introducing a realized application. The application is called dinerplus and organizes guests and menus for a hobby chef, restaurant or catering service. Further we discuss related work on the aspect of experience management in SMEs using intelligent systems.

1 Introduction

Experience management is described as a special field of knowledge management where specific knowledge required for a certain problem solving task is stored, organized and provided [Ber02]. As pointed out by Minor [Min05], experiences are mainly kept in people's heads and successful experience management systems have to include the user in all development phases of such a system. However, systems that can deal with people's experiences and support regular tasks do not necessarily require its application in large enterprises, the same technologies can be applied in small and medium sized enterprises (SME) in order to make their work more efficient.

Intelligent technologies like Case-Based Reasoning (CBR) can be applied in SMEs or semi-professional environments, because the idea behind CBR is quite easy to explain, which makes people trusting in a system but also provides comprehensive support fulfilling various tasks. Therefore a system first has to collect experiences by "observing" the user before a system can be employed. Observing users can also mean using previously available experiences like documents, databases, emails, etc. and mining those for relevant experiences, which afterwards will be provided.

We have experienced that a very easy way of explaining the ideas of CBR is using a prototype like CookIIS [INH$^+$09], one of the competing systems at the Computer Cooking Contest (CCC). The CCC is a technology competition carried out at the Case-Based Reasoning conferences demonstrating the capabilities of CBR. Those showcase systems further allow the explanation of the technologies using an every day problem: *What do I have in the fridge and how can I create a meal out of it?* Presenting CookIIS and explaining the underlying technologies using easily understandable examples usually gains trust of stakeholders.

Inspired by the prototype, we developed the idea to use CBR in social media applications more intensively, because its strength, working already with a small amount of examples and developing its capabilities while it is applied is very attractive to SMEs. This paper describes an application that collects, organizes and provides pieces of experiences of individuals and provides those experiences when necessary. Section 2 introduces the idea of the application followed by the description of the system's implementation. The following section 3 presents a qualitative evaluation of the resulting menus. Section 4 discusses related approaches and the final section 5 gives a short summary of the paper and provides an outlook on on-going and future work.

2 Application Domain: dinerplus - how to culinary pamper your guests

2.1 Idea and challenges

Anybody who has ever invited guests for dinner, faces several problems. As host you have to remember what your guests like to eat and what not. Also you should not forget, which guests do not like each other. All this information should be considered by the host when creating menus for this invitation. In addition, the search for new recipes and their combination to menus leads to a considerable effort. Eventually the host will not serve his guests always the same dishes. We have designed a system - dinerplus - to support all people who see themselves confronted with these problems. The basic idea is penned by Philippe Bouvet, a web designer from Hildesheim. We combined his idea with techniques from Case-Based Reasoning (CBR) to search for recipes and combine menu variations based on previous experiences. Here, the information about the selected guests can be included into a query and considered in the search for recipes.

The goal of our application is to offer users a platform on the world wide web, where recipes can be created and searched as well as combined to menus. Also, the user can enter the guest's data in order to be reused in the future and maintain the information about their eating preferences. Guests can be assigned to menus and the user can enter information about wanted and unwanted ingredients, the type of cuisine or specify the category of a recipe for a certain course. Using CBR allows finding similar recipes and adapt existing ones to fit the users preferences if necessary. The created menus can be saved and rated by the user after cooking. By this rating it is possible to evaluate the adaptation and the combining process. An expert of the domain, i.e. a (hobby)chef, can view over the rated recipes and revise the solution in order to create cookable and tasty recipes by the CBR engine.

2.2 Concepts of dinerplus

The challenge we faced is to get five courses per menu that have to be suitable for another. Therefore we have to create models representing general knowledge and others that represent specialized knowledge. General knowledge contains all kinds of ingredients that can be used for preparing dishes as well as drinks. However, each person has a different taste, so such a system also has to represent individual preferences for preparing meals (specialized knowledge). Interaction between generalized and specialized knowledge is the key challenge of systems like dinerplus.

The structure is result of the underlying domain: A menu is composed of several recipes of dishes and drinks. These recipes are composed using a whole variety of ingredients. Additionally, recipes and ingredients can be assigned to various categories. The outcome of this are two different structured approaches for modeling the cases. Our first approach was to represent a complete menu by a case. But there are several drawbacks to this model: A case is a combination of recipes for dishes and drinks. Storing every useful combination leads into a huge case base that would be hardly manageable and would have reduced the efficiency of the prototype. Furthermore it is very difficult to maintain a single recipe and it will not be possible to retrieve only one recipe. However this is a use case the application should fit. Eventually, we decided to use an approach that contains redundant information in the description and the solution of our cases allowing more efficient CBR.

We decided to follow another basic approach, where each recipe is represented as a case. A menu is a combination of several retrieved cases and is only stored in our database. The modeling of our cases is based on the modeling of CookIIS, a Case-based Recipe Advisor described in [INH^{+}09]. In contrast to CookIIS we modeled two different case models - one for dishes and one for drinks instead of a single one that covers a dish with drinks. Both case models are similar to each other and differ mainly in the number of attributes. The description of our cases consists of the recipe's name, category, type of cuisine, course and preparation tool. In addition there are several attributes for the ingredients. For each modeled ingredient category two attributes are needed, one for desired ingredients and the other for unwanted ingredients. The preparation description, the time of preparation and a complete list with amounts and units of the ingredients for the recipe are part of the solution. The case has two additional components, an adaptation component and a rating component.

The adaptation component contains the information which ingredients have been previously adapted. These pieces of experiential information are reused for result representation and in general follow this structure:

```
Replace <old_ingredient> through <new_ingredient>
Please skip <old_ingredient>
```

The rating component stores a user's review of the recipe. This approach gives us a better flexibility for composing dishes and drinks in a menu. While requiring up to ten retrieval steps for a menu we gain the possibility to include information from previous steps to the next query. This allows us to find more adequate recipes with the information a user provides [BRA09].

2.3 Implementation

Overall the dish cases have 112 attributes, of which 60 attributes are responsible for handling the ingredients. For each ingredient category five attributes are required to handle our adaptation process. The general case representation can be seen in Figure 1.

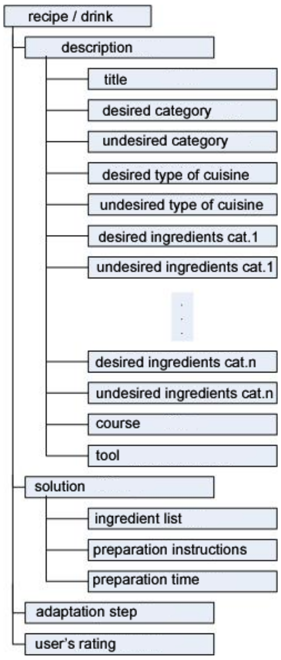


Figure 1: dinerplus Case Representation

Our drink cases have only 62 attributes because of less ingredient categories. Ingredients, categories and type of cuisine are modeled as taxonomies, which are also used for similarity computation. To compute the similarity between query and cases we use a combined symbolic similarity via taxonomies and similarity matrices. In a similarity matrix it is possible to assign different similarity values to certain concepts of an underlying taxonomy[HNB+10]. Thus we are able to work with more specific similarity values. The global similarity is computed from 20 attributes in the dish cases and 11 attributes in drink cases [Reu10]. The weight for each attribute is based on (Figure 2) and has been developed in CookIIS [INH+09] and adapted after discussions with experts. In general, the similarity measure follows the principle the more characteristic a category of ingredients is, the higher it is weighted.

A number of completion and adaption rules have also been defined. Completion rules are meant to determine additional information in cases while indexing the case base or in queries before the retrieval. Furthermore these rules are responsible for setting filters for taxonomies to avoid a group of concepts for the retrieval. Adaptation rules are applied

$$sim_{global} = \frac{1}{\sum weight_{localsim}} x$$

$$(10 \text{ x } (sim_{language}) + 6 \text{ x } (sim_{meat} + sim_{fish}) + 5 \text{ x } (sim_{specie} + sim_{vegetable}) + 4 \text{ x } (sim_{fruit} + sim_{dishcategory} + sim_{typeofmeal}) + 3 \text{ x } (sim_{typeofcuisine} + sim_{supplement}) + 2 \text{ x } (sim_{basic} + sim_{milk}) + 1 \text{ x } (sim_{extradiet} + sim_{methodofpreparation} + sim_{tool} + sim_{fulltext} + sim_{fluid} + sim_{minor} + sim_{oilandfat} + sim_{spiceandherb}))$$

$$Sim_{globaldrink} = \frac{1}{\sum weight_{localsim}} x$$

$$(10 \text{ x } (sim_{language}) + 6 \text{ x } (sim_{Alcohol}) + 4 \text{ x } (sim_{NonAlcohol}) + 3 \text{ x } (sim_{Fruit} + sim_{spiceandherb} + sim_{category}) + 2 \text{ x } (sim_{typeofcuisine}) + 1 \text{ x } (sim_{typeofmeal} + sim_{extradiet} + sim_{title} + sim_{fulltext})$$

Figure 2: global similarity dinerplus

to retrieved cases to fit the users wish. In dinerplus we use model-based adaptation via defined taxonomies. The adaptation consists of the following five steps:

1. First the unwanted concepts are determined.
2. Then fitting concepts for replacement are calculated.
3. If no concepts are found for replacement instructions to omit the ingredients are generated.
4. The next step is to generate instructions for replacing ingredients.
5. Finally desired ingredients replace possible existing ones to fit the users wish.

In our prototype we use a two-layered architecture (Figure 3). We have a database server with PostgreSQL and a web server for our graphical user interface (GUI) and the application logic is based on Java. Using PostgreSQL and Java we gain a highly platform independent system. The bottom layer of our architecture contains the database, where all data about users, guests, recipes and menus is stored. The top layer contains our graphical user interface and the framework for CBR. We used the Information Access Suite (IAS) [emp05] from Attensity Europe GmbH for modeling our case representation structures and rules. There is a communication flow between the GUI and the database to store and query user input. The IAS creates the case base, more detailed a case retrieval net (CRN) [Len99] based on our recipe data. Further the CBR engine IAS provides various interfaces for executing the retrieval and adapting results. For user interaction we have implemented a GUI for communicating with the users that interacts with the retrieval engine IAS and the database itself [Reu10].

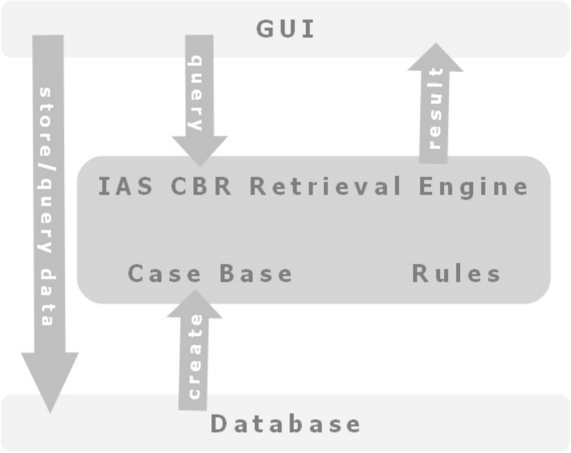


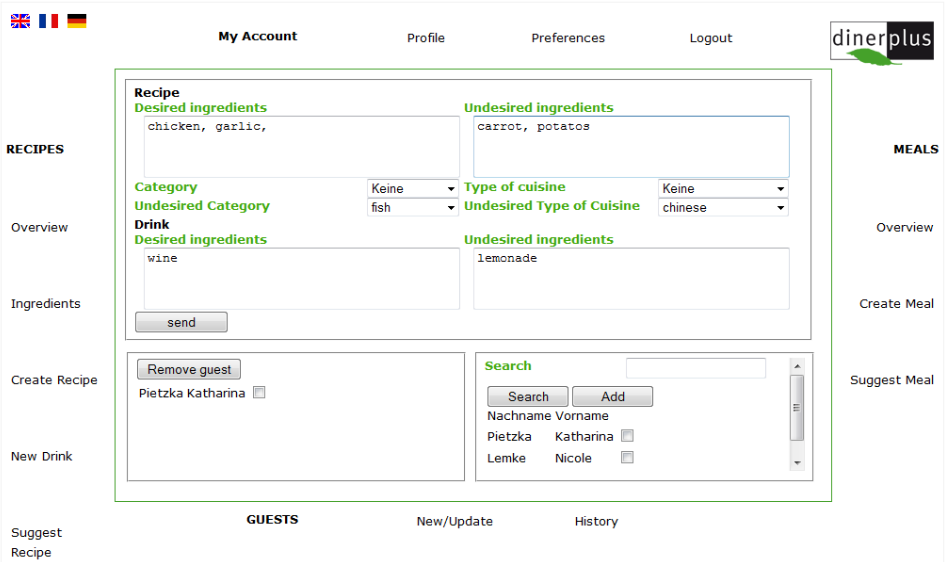
Figure 3: dinerplus prototype architecture

Our prototype is on the one hand able to retrieve a single dish with a drink and on the other hand it can retrieve up to five dishes and drinks for a menu. Two retrieval methods (one for single recipe retrieval, one for menu retrieval) have been implemented to connect the Java GUI and the IAS. Users can enter desired and unwanted ingredients, type of cuisine or categories and assign guests to a recipe or menu. Both methods then start by identifying the dislikes of the assigned guests from the database. These information enhance the query that is sent to the IAS retrieval engine (a sample can be seen in the screen shot in Figure 4). The retrieval is initiated by querying for the main dish (third course), followed by creating subsequent queries for the starter, the intermediate 1 dish (second dish), the intermediate 2 dish (fourth dish) as well as the dessert. Each included recipe has to match the menu's type of cuisine and fit the given preferences (likes and dislikes). After a complete menu has been created, drinks for each course are queried from the case and the most similar is assigned to the according dish. For that reason the menu retrieval is an up to five-time repetition. Between each step the gathered information is included into the next query to find an adequate recipe. In case the system discovers guests who are invited to one event, but do not get along with each other, dinerplus marks the guests and gives a short note [Reu10].

The information gathered during the dinner can be used to improve the quality of the recommendations. The user can enter information about the guests reaction to the menu or to other guests. This information can be used by the next query, in which the guests are involved. Additionally, the user can rate the dishes and drinks, so an expert can adjust the rules of dinerplus or create new ones.

3 Results

The prototypical implementation of dinerplus has been realized as a student thesis, where the aforementioned functionalities have been implemented. The prototype has been presented to the web designer, Philippe Bouvet, and convinced him so we will work on a productive system for the future. The starting page of the GUI can be seen in figure 4 containing links to the previously described as well as user account management functionalities.



Recipe
Desired ingredients: chicken, garlic,
Undesired ingredients: carrot, potatos
Category: Keine
Undesired Category: fish
Type of cuisine: Keine
Undesired Type of Cuisine: chinese
Drink
Desired ingredients: wine
Undesired ingredients: lemonade

Pietzka Katharina □

Search
Nachname Vorname
Pietzka Katharina □
Lemke Nicole □

Figure 4: Screenshot dinerplus

Further, we evaluated the menu retrieval using the CookIIS menu designer as benchmark system. We used the CCC09 exercise and competition queries and performed them on both systems. dinerplus composed menus with the same quality as the CookIIS system did. The evaluation results were reviewed by experts and rated according to its cookability. Therefore 15 users worked with dinerplus and each user answered a questionnaire including five queries for single recipe retrieval and six queries for a complete menu retrieval. The test case based included 1,600 recipes and 14 drinks.

The user evaluation, which is depicted in figure 5 was satisfying as well. 78% of the retrieved recipes were rated as a useful selection by dinerplus. 56% of the adaptation results presented to the test users were also rated as useful. Overall, the assesment of the similarity measure turned out to be sufficient. However, the organization of the concepts did not always fit as expected. The major drawback of the unsatisfying adapation results were caused by pure model-based adapation with no respect to the context of the recipe. This leads us to the conclusion that the adaptation processes has to be refined. More ideas how we will address this problem will be presented in the last section of this paper.

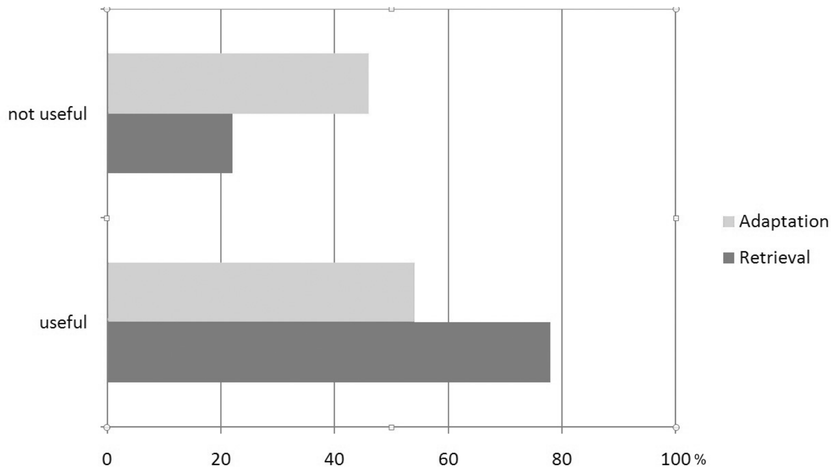


Figure 5: Evaluation Results

4 Discussion of Related Work

The cooking domain has been used applying CBR many years ago – probably because when you are cooking you want to give correct suggestions but still provide some space for experimentation and creativity. Systems that suggested preparation advice for meals are JULIA [Hin92] and CHEF [Ham86]. While CHEF was a planning application that builds new recipes in the domain of Szechwan cooking, JULIA integrated CBR and constraints for menu design tasks. Therefore, JULIA uses a large taxonomy of concepts and problem decomposite with fixed decomposition plans. In comparison to the approach presented in this paper, none of the mentioned systems are integrated in a web community of another type of community of experts. However, dinerplus is developed to be integrated in a web community.

The CCC has brought some new ideas and methodologies to this type of application domain. Along with CookIIS there are a couple of competitors (e.g. [ZHND08, BBB^{+}08, DPDA08, MBGW10]) facing the similar challenges like assigning appropriate similarity measures, recognizing ingredient concepts, determining the type of cuisine, handling unwanted ingredients, organizing work flows, etc.

Given the fact of the fast and broad dissemination of web communities and therewith the availability of huge amounts of experience knowledge, it is very promising integrating experiences in CBR, because the underlying methodology of CBR relies on previously made experiences. It is important to use, combine and further develop technologies that have already been applied on the Web (2.0) together with standard technologies to meet the expectations of today's knowledge management. Therefore software engineers for CBR systems should try to create web-based systems, because that might be the only way to receive feedback [SCBC09].

5 Conclusion and Outlook

In this paper we have presented how CBR can be applied to an idea that was carried out by a SME. We further used previous experiences of experts how to organize a system and which information are relevant for that particular task. Based on this model, we have created a CBR system and according GUI. By using the system, experiences are continuously collected and provided on demand. The related work section shows that the application domain as well as the methodology has been successfully applied in the last years. With the forthcomings of the social web they have become more and more important and there are still some open challenges as it has been pointed out by Plaza [Pla08].

Our supreme goal is to get a productive system on the web to support (hobby)chefs, cooking schools and even gastronomy creating menus. Therefore, the next steps are to reimplement the CBR part of our system with Open Source software and to integrate more knowledge about menu composition in dinerplus. For the development of dinerplus we will further focus on dynamic similarity measures and adaptation to meet different contexts. The challenge is to get the required knowledge into formal rules with a minimum of effort. It also raises the question of what context should be given priority. Possible candidates are different occasions or already used ingredients in selected recipes.

References

[BBB+08] Fadi Badra, Rokia Bendaoud, Rim Bentebitel, Pierre-Antoine Champin, Julien Cojan, Amélie Cordier, Sylvie Desprès, Stéphanie Jean-Daubias, Jean Lieber, Thomas Meilender, Alain Mille, Emmanuel Nauer, Amedeo Napoli, and Yannick Toussaint. TAAABLE: Text Mining, Ontology Engineering, and Hierarchical Classification for Textual Case-Based Cooking. In Schaaf [Sch08], pages 219–228.

[Ber02] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Springer, Berlin, Heidelberg, 2002.

[BRA09] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. A Value Supplementation Method for Case Bases with Incomplete Information. In Lorraine McGinty and David C. Wilson, editors, *Case-based Reasoning in Research and Development, Proceedings of the 8th International Conference on Case-Based Reasoning (ICCBR-09)*, LNCS, pages 389–402. Springer, July 2009.

[DPDA08] Juan DeMiguel, Laura Plaza, and Belén Díaz-Agudo. ColibriCook: A CBR System for Ontology-Based Recipe Retrieval and Adaptation. In Schaaf [Sch08], pages 199–208.

[emp05] Technisches White Paper e:Information Access Suite. Technical report, empolis GmbH, September 2005. `http://www.empolis.de/executive-forum/Downloads/m_Technisches-WP-IAS_V1.1.1_050906.pdf`; letzte Verifikation am 11. Juni 2007.

[Ham86] Kristian J. Hammond. CHEF: A Model of Case-Based Planning. In *American Association for Artificial Intelligence, AAAI-86, Philadelphia*, pages 267–271, 1986.

[Hin92] Thomas R. Hinrichs. *Problem solving in open worlds*. Lawrence Erlbaum, 1992.

[HNB$^+$10] Alexandre Hanft, Régis Newo, Kerstin Bach, Norman Ihle, and Klaus-Dieter Althoff. CookIIS - A successful Recipe Advisor and Menu Advisor. In Stefania Montani and Lakhmi Jain, editors, *Successful Case-based Reasoning applications*, Studies in Computational Intelligence 305, pages 187 – 222. Springer, 2010.

[INH$^+$09] Norman Ihle, Régis Newo, Alexandre Hanft, Kerstin Bach, and Meike Reichle. CookIIS - A Case-Based Recipe Advisor. In Sarah Jane Delany, editor, *Workshop Proceedings of the 8th International Conference on Case-Based Reasoning*, pages 269–278, Seattle, WA, USA, July 2009.

[Len99] Mario Lenz. *Case Retrieval Nets as a Model for Building Flexible Information Systems*. Dissertation, Humboldt University of Berlin, Berlin, 1999.

[MBGW10] Mirjam Minor, Ralph Bergmann, Sebastian Görg, and Kirstin Walter. Towards Case-Based Adaptation of Workflows. In Isabelle Bichindaritz and Stefania Montani, editors, *ICCBR*, volume 6176 of *Lecture Notes in Computer Science*, pages 421–435. Springer, 2010.

[Min05] Mirjam Minor. Introduction Strategy and Feedback from an Experience Management Project. In Klaus-Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas Roth-Berghofer, editors, *Professional Knowledge Management, Third Biennial Conference, WM 2005, Kaiserslautern, Germany, April 10-13, 2005, Revised Selected Papers*, volume 3782 of *Lecture Notes in Computer Science*, pages 284–292. Springer, 2005.

[Pla08] Enric Plaza. Semantics and Experience in the Future Web. In Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors, *ECCBR*, volume 5239 of *Lecture Notes in Computer Science*, pages 44–58. Springer, 2008.

[Reu10] Pascal Reuss. DinerPlus – Menükomposition mit Case-Based Reasoning. Bachelor's thesis, University of Hildesheim, August 2010.

[SCBC09] Barry Smyth, Pierre-Antoine Champin, Peter Briggs, and Maurice Coyle. The Case-Based Experience Web. In Sarah Jane Delany, editor, *ICCBR 2009 Workshop Proc., Workshop Reasoning from Experiences on the Web*, pages 74–82, July 2009.

[Sch08] Martin Schaaf, editor. *ECCBR 2008, Trier, Germany, September 1-4, 2008, Workshop Proceedings*, Hildesheim, Berlin, 2008. Tharax Verlag.

[ZHND08] Qian Zhang, Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Back to the Future: Knowledge Light Case Base Cookery. In Schaaf [Sch08], pages 239–248.