Computer-based adaption of cooking recipes integrated in a speech dialogue assistance system

Karen Insa Wolf, Stefan Goetze and Frank Wallhoff

Abstract Speech input and output allow for a natural and intuitive means to communicate with a technical device or system. This motivates the development of speech dialogue systems for convenient human machine interaction. A speech dialogue system is particularly suitable in scenarios in which the user cannot use his or her hands for interaction, for example while driving a car. In kitchen environments a speech dialogue system promises to be very useful as well. The application CooCo (Cooking Coach) aims at providing assistance under working conditions in a common kitchen. CooCo supports the user to choose recipes best matching the requests of the user and gives advices during the cooking processes. In this paper, the focus lies on the conceptual view on building up CooCo and the integration of a computer-based approach to adapt cooking recipes.

1 Motivation

Nowadays, speech dialogue systems become more and more common in everyday use, even for people with low affinity to new technologies. A speech dialogue system is particularly suitable in scenarios in which the user cannot use his or her hands for interaction, for example while driving a car (cf. e.g. Larsson and Villing (2007)). Speech dialogue systems promise to be very useful as well during daily work at home in the kitchen. The user can ask for recipes while doing the dishes or can get reminders regarding timing and next steps from an assistance system while cooking.

Karen Insa Wolf, Stefan Goetze, Frank Wallhoff
Fraunhofer Institute for Digital Media Technology IDMT, Marie-Curie-Strasse 2, 26129 Oldenburg, Germany, e-mail: {karen.insa.wolf, stefan.goetze, frank.wallhoff}@idmt.fraunhofer.de

Frank Wallhoff
Jade University - Institute for Technical Assistive Systems, Westerstr. 10-12, 26121 Oldenburg, Germany, e-mail: frank.wallhoff@jade-hs.de

Assuming a flexible dialogue management, spontaneous utterance of the user (like e.g. ≪Oops, I do not have ...≫) can be processed.

The nutrition topic itself is of particular relevance in context of the demographic change and, thus, for ambient assisted living. Well-balanced nutrition is an important pillar of a healthy lifestyle and healthy ageing (cf. e.g. Zbeida et al (2014)). Nutrition consulting can contribute to the prevention and therapy of diseases thereby enabling a longer, independent life. With increasing age, the need for a supporting nutrition consultation can grow, as chronic diseases become more frequent. Depending on the health status of a person, cognitive impairments such as memory decline or decline in the capacity to structure ones daily routine increase, both of which affect the ability to independently schedule food intake. Loss of the partner can also induce a less balanced nutrition.

Contributing to healthy ageing and long-lasting independence of older people additionally motivates the development of the speech dialogue system CooCo (Cooking Coach), introduced in Wolf et al (2015b). In this paper, the focus lies on the conceptual view on building up CooCo and the integration of a computer-based approach to adapt cooking recipes. The automatic adaption of recipes can be helpful to enlarge the recipe database for special diet purposes (vegetarians, allergic persons).

2 Concept of CooCo

The application CooCo aims at providing ambient and non-dominant meal preparation assistance under real-world conditions. Keyboard, mouse or touchscreens are no practical user interfaces while cooking. Therefore, CooCo uses automatic speech recognition and speech output as user interface. Two different use cases are considered in the design of the system.

First, in the use case *recipe advice*, CooCo helps the user to choose the right recipes based on her or his requests concerning ingredients, preparation time, complexity of preparation or even tastes (e.g. sweet or hot). Over time CooCo learns the likes and dislikes of the user to optimize the interaction. The recipe advice mode includes generic models of gustatory preferences, e.g. hot or sweet depending on typical amount of ingredients like chili or sugar, and a computer-based adaptation of the cooking recipes, cf. Wolf et al (2015a).

Second, in the use case *cooking support*, CooCo supports the user during the cooking process. The goal thereby is not only to read the cooking steps as shown e.g. in Schäfer et al (2013), but to build up a plan considering expenditure of time and dependencies of single cooking steps. CooCo formulates an action plan considering active time of the user, like cutting, and passive time, like simmering.

Both tasks, recipe advice and cooking support, require a context-based dialogue system including modules for interpreting, planning and re-planning, memorizing and learning. These modules are typical for a cognitive (technical) system. Such systems can represent system-relevant aspects of the environment internally, which is the basis for cognitive functions, cf. Gamrad (2011), and allows flexible behaviors.

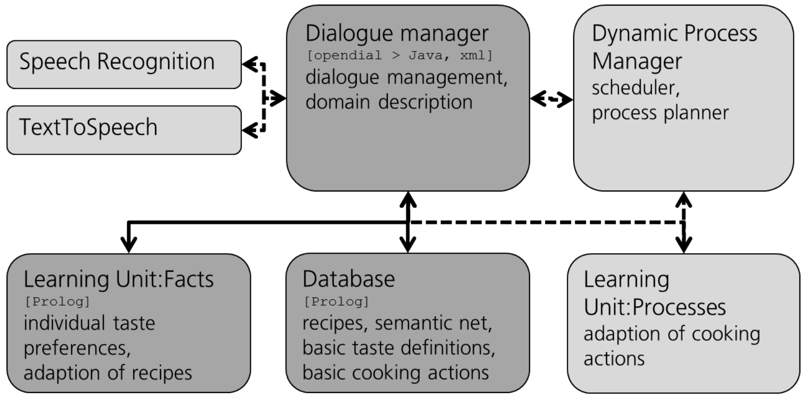

Fig. 1 A schematic diagram of the architecture of CooCo. The modules in dark gray are active in the application discussed in this paper.

In this paper, just the first use case is addressed in more detail to explain the integration of the new feature of automatic adaption of cooking recipes. The concept of the second use case is explained in more detail in (Wolf et al, 2015b).

3 Modules of CooCo

The system of CooCo is based on seven main modules, as explained in the following. A schematic diagram of the architecture of CooCo is shown in Fig. 1. *Speech recognition* and *text to speech* (TTS) modules are introduced as part of a dialogue system in section 3.1. Details for the *dialogue manager* are given in section 3.2. The *database* is introduced in 3.3. The approach of the adaption of cooking recipes as part of the *learning unit:facts* is explained in section 4. The *dynamic process manager* is conceptually based on a scheduler, handling time processes, and a process planner which derives a plan for the cooking process regarding dependencies. The concept of the dynamic process manager is explained in more detail in Wolf et al (2015b). In future versions, the *learning unit:processes* should adapt parameters of the cooking action description to match better the preferences and practices of the user.

3.1 Dialogue systems

In Fig. 2 the architecture of a speech dialogue system is sketched, cf. e.g. Eliasson (2006). The speech recognizer translates from spoken input to written text. The interpretation of the utterance is encapsulated in the module *dialogue manager* based

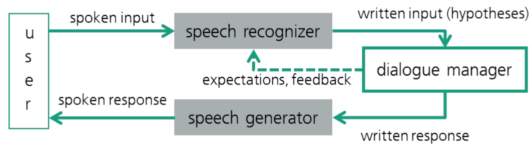

Fig. 2 Architecture of a speech dialogue system with an optional feedback between dialogue manager and speech recognizer.

on written text input. To improve speech recognition results, expectations of the content of the user utterances can be introduced in the speech recognizer. A speech generator (text-to-speech TTS) is used to articulate the response.

Focused on dialogue management, speech recognition is often linked as an off-the-shelf module in the system, cf. e.g. Eliasson (2006). Different systems are available for this purpose, some of them are able to return not just one best hypothesis but a ranked n-best list of hypotheses for further context based analysis, cf. Morbini et al (2013). For CooCo automatic speech recognition (ASR) is used, cf. Goetze et al (2010). A feedback loop between the dialogue manager and the speech recognizer is designed in the concept based on different input hypotheses entered in the dialogue manager to improve speech recognition. The feedback information of the chosen most likely recognition results and expectations are used to weight the probability of the words in the dictionary of the recognizer. This helps optimizing the recognition rate.

3.2 Dialogue manager

The tasks of the dialogue manager can be summarized as

1. interpretation of the (written) utterance
2. definition of next action resulting in a response (question, information, advice)

These tasks include diverse subtasks as parsing, updating actual information knowledge, or planning. There are different approaches to solve these tasks depending on the specific use case and necessary functionality of the dialogue manager, cf. e.g. Morbini et al (2013). For CooCo's first task, giving recipe advice, a slot-filling system would be sufficient. The second task, cooking support, is more demanding if the aim is more than just articulating the cooking steps when the user asks for a next step. Applying the categories of Morbini et al (2013), a negotiation and planning system is needed. Lison (2014) distinguishes between hand-crafted and statistical approaches for a speech dialogue manager and proposes the toolkit OPENDIAL (OpenDial, 2015) to combine both. Pure statistical approaches as partially observable Markov decision processes (POMDPs), described in Williams (2008), need training data sets to build up the dialogue setting. Central element of OPENDIAL is an information-state which is updated cyclically based on the information of the user (and maybe from other sources, e.g. sensors). The dialogue manager of CooCo is based on OPENDIAL. Another option for an information-state approach is the

TRINDI Concept (task oriented instructional dialogue, Larsson and Traum (2000)), as proposed in Wolf et al (2015b), but the basic algorithm does not consider probability and utility values to influence the dialogue policy. Therefore, the approach of Lison (2014) is used.

3.3 Database

In the knowledge database facts are stored: recipes, ingredients, basic taste definitions and basic cooking actions. The *database* contains currently 1.222 recipes (Herz, 2015). The ingredients are stored as a semantic net as sketched in Fig. 3. This network is used for different tasks. Assuming the user requests a recipe with meat, CooCo can resolve this request by looking for all recipes using chicken, turkey, etc. as ingredients. A shopping list can be generated in a sorted form for a well-organized shopping tour, e.g. all vegetables are written in one block of the list. Finally, properties based on the inheritance relations in the network are applied within the algorithm to adapt the cooking recipe, as described in Section 4.

4 Computer-based adaption of cooking recipes

The computer-based variations of cooking recipes addresses topics of artificial intelligence and machine learning approaches, cf. e.g. Cordier et al (2014). The task to derive the consequences of the substitution of an ingredient on the textual description of the preparation steps requires techniques of natural language understanding, e.g. Dufour-Lussier et al (2014). Other approaches aim at replacing ingredients, e.g. by randomizing recipe items (Easierbaking, 2015), by using cognitive super computing (based on IBM's computer system WATSON, (IBM and Institute of Culinary Education, 2015)) or by just enlarging the database (by the help of a community) to find a matching recipe for every combination of ingredients (SousChef, 2015).

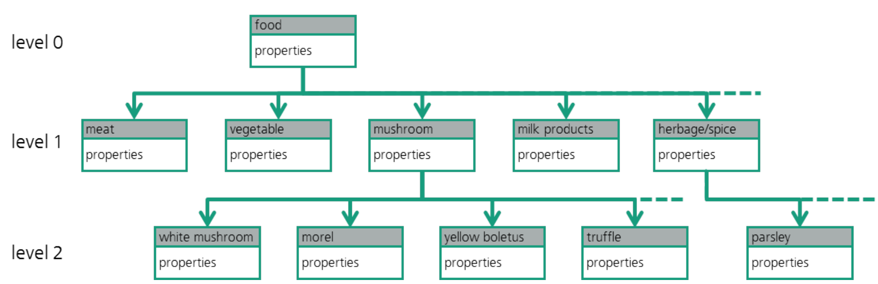

Fig. 3 Ingredients structured in a semantic net.

4.1 Basic approach

In Wolf et al (2015a) a heuristic computer-based approach is described to vary cooking recipes by replacing ingredients. The approach is based on a scoring system. The score value is used to rate different ingredients as candidates to substitute a specific ingredient of a recipe. This substitution score depends on different factors: (1) rating of the similarity between the ingredient which has to be replaced and the substitution candidat, (2) rating how well the substitution candidate fits the recipe, and (3) gustatory preferences of the user. The substitution candidate with the highest score is proposed to the user.

The rating of the similarity between the ingredients is based on the one hand on their level and their distance as elements in the semantic net, shown in Fig. 3 and on the other hand on a comparison of their nutrition values. How well a substitution candidate fits the recipe is derived from statistical numbers based on the recipe database. Finally, users differ in their gustatory preferences, one likes more traditional recipes, while the other is more open to new tastes. Therefore, the gustatory preferences of the user are considered by two user parameters, referred to as experimental levels: The experimental level e_{cd} influences how common or uncommon a substitution candidate should be. The level e_{cb} regulates how common or uncommon the combination of a substitution candidate and all remaining elements of the chosen recipe is. For both levels three adjustment steps can be chosen by the user, ranging from 1 = very common to 3 = very uncommon.

4.2 Exemplary results

The approach described above is conceptually tested, further implementation and evaluation is ongoing work. The starting point is a simple mushroom soup recipe:

250	g	common mushrooms,	40	g	butter,
40	g	flour,	5	dl	bouillon,
5	dl	milk,	-	-	salt,
1	tb	parsley, minced,	-	-	pepper

Numerical results for the different experimental levels e_{cd} and e_{cb} are listed in Tab. 1. The results show, that a user with a low e_{cd} of 1 and a medium or high e_{cb} of 2 or 3 will be recommended a tomato soup. In case a very common combination of ingredients is wanted ($s(1,1)$), morel soup is proposed instead. Reason for this is that the recipes with common mushrooms and morel often share the basic combination of ingredients. Obviously, the results depends strongly on the content of the database. A user who wants uncommon ingredients in an uncommon combination gets truffle as substitution candidate ($s(3,3)$). Elements of the class mushrooms are mostly preferred. A whole class like ”mushrooms” could also be excluded, resulting in recommendations of cauliflower as substitution candidate as a less common ingredient than tomatoes. If a high experimental level e_{cb} for the final combination in the recipe is chosen you get even cucumber as candidate.

Table 1 Numerical results[a] of adapting a cooking recipe. The result $s(j,k)$ means s based on $e_{cd} = j$, $e_{cb} = k$. The respective candidate with the largest score s is marked in bold letters. Only some examples of the possible substitution candidates are listed.

.

	y. boletus	morel	truffle	red pepper	tomato	cucumber	cauliflower
$s(1,1)$	46	**56**	14	44	51	8	43
$s(1,2)$	46	36	34	54	**61**	28	43
$s(1,3)$	46	16	54	64	**71**	48	43
$s(2,1)$	56	**76**	34	24	31	18	53
$s(2,2)$	56	**56**	54	34	41	38	53
$s(2,3)$	56	36	**74**	44	51	58	53
$s(3,1)$	66	**96**	54	4	11	28	63
$s(3,2)$	66	**76**	74	14	21	48	63
$s(3,3)$	66	56	**94**	24	31	68	63

[a] The numbers differ slightly to those published in Wolf et al (2015a) due to modifications of some of the nutrition values of the ingredients.

4.3 Use cases

The central task in the following two use cases is to propose a tasty recipe based on the user's input by replacing ingredients. The intention of the user differs in the scenarios. Both use cases can be extended by including the question of undesired ingredients. In order to enlarge the number of possible recipe candidates, the proposed recipe variation approach can be applied in this case additionally to substitute undesired ingredients.

4.3.1 Use case 1: ≪Surprise me.≫

Based on one chosen recipe the user asks for a variation of this recipe. A similar scenario is that the user realizes that one ingredient is missing but s/he still wants to cook the chosen recipe accepting variations. In both cases, CooCo can choose freely possible substitution candidates. In the first case, the ingredient to be substituted is not defined by the user. In the second case, this ingredient is the missing one.

4.3.2 Use case 2: ≪Work with what I have.≫

The user specifies some ingredients, s/he wants to work with, but no recipe can be found in the database which uses all desired ingredients. The task for CooCo is now to propose one recipe which matches by replacing missing ingredients of the recipe with those defined by the user. For this scenario, a plausibility check is performed since not each combination of ingredients presents a suitable option for a recipe.

5 Implications for the speech dialogue plot

Both use cases, described in Section 4.3, demands different plots of the speech dialogue. The first use case (≪Surprise me.≫, cf. Section 4.3.1) can be integrated in a simple version of a dialogue by adding the question whether a recipe should be adapted if wanted or necessary. As CooCo is more or less free to choose the candidate which substitutes one of the ingredients of the recipe, the chance is large that an appropriate solution will be found. It is different in the second use case (≪Work with what I have.≫, cf. Section 4.3.2) as there, the set of ingredients which can be used as candidate for the substitution is very limited. CooCo has to discuss with the user what would be the best option starting from probably not ideal substitution solutions. An example:

In use case 2 the user desires a recipe with the ingredient set $I_{us} = \{$*butter, flour, parsley, bouillon, red pepper*$\}$. The recipe that matches I_{us} best is mushroom soup, given in Section 4.2, based on the simple rule to look for those recipes with the smallest number of missing ingredients I_{ms}. However, red pepper is not part of the original recipe. Based on the substitution score it is then checked whether red pepper is a suitable substitution candidate c_{sb} for one of the missing ingredients.

Aiming at an efficient dialogue an assumption is introduced that some ingredients are standard ingredients, e.g. $\{$*pepper, salt, bouillon*$\}$. They are supposed to be available also in case the user did not mention them explicitly. But this first guess has to be confirmed by the user. Then, the only missing ingredients left are $I_{ms} = \{$*common mushrooms, milk*$\}$. Considering the experimental levels, the score s is derived for all pairs of c_{sb} with one of the elements of I_{ms}. The highest score $s = 54$ is reached for the experimental level $e_{cd} = 1$. Considering a threshold scheme of $[120 \ldots 80]$ (very good), $]80 \ldots 40]$ (acceptable), $]40 \ldots 0]$ (not recommended) for s, the substitution pair *red pepper - common mushrooms* is evaluated as "acceptable". In no case it is an option to replace *milk* with *red pepper*, the highest score is $s = 29$. *Milk* remains here as missing candidate. Two different last options are possible:

(1) Ask the user explicitly whether there is after all a potential substitution candidate. If yes, repeat the procedure.

(2) Evaluate how well the missing ingredient could be omitted. For this, check if other ingredients could make up for the omission by increasing its quantity based on their similarity. In this specific example, the result of 17.5 for *milk* in relation to *butter* is not promising enough to propose this as solution. As final step, the amount

of liquid within the recipe ingredients is checked leading here to an increase of the amount of bouillon to recover the original amount of liquid.

The final solution with appropriate comments based on the score s is presented to the user. The example shows that there are some intermediate dialogue steps necessary to get the final result of the adapted recipe. Therefore, current work considers user study to support the definition of appropriate user dialogue scripts.

6 Discussion and Conclusion

This paper describes the concept of the speech dialogue assistance system CooCo that aims to support users during daily work at home in the kitchen. An approach to derive recipe variations by replacing ingredients is introduced. Two different use cases are addressed. The presented examples provide reasonable substitution results. However, the test cases have to be enlarged in further evaluation steps. The adaption algorithm is an initial version to include such a feature within the speech dialogue to model a more complex interaction with the user. The algorithm is still limited and may propose uncommon or nonsense substitution suggestions. But its implementation is done easily and with less effort than manually listed potential substitution pairs. The feature provides a good experimental environment to design and test speech dialogue approaches for practical use even based on the simple use case of *recipe advice*.

References

Cordier A, Dufour-Lussier V, Lieber J, Nauer E, Badra F, Cojan J, Gaillard E, Infante-Blanco L, Molli P, Amedeo N, Skaf-Molli H (2014) Taaable: a case-based system for personalized cooking. In: Montani S, Jain LC (eds) Successful Case-based Reasoning Applications-2, Springer, Studies in Computational Intelligence, vol 494, pp 121–162

Dufour-Lussier V, Le Ber F, Lieber J, Nauer E (2014) Automatic case acquisition from texts for process-oriented case-based reasoning. Information Systems 40:153–167

Easierbaking (2015) The Clickable Recipe Maker. www.easierbaking.com, 9.7.2015

Eliasson K (2006) The use of case-based reasoning in a human-robot dialog system. PhD thesis, Linköping Institute of Technology, Department of Computer and Information Science, Linköping University

Gamrad D (2011) Modeling, simulation, and realization of cognitive technical systems. PhD thesis, Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau und Verfahrenstechnik der Universität Duisburg-Essen

Goetze S, Moritz N, Appell JE, Meis M, Bartsch C, Bitzer J (2010) Acoustic user interfaces for ambient-assisted living technologies. Informatics for Health & Social Care 35 (3-4):125–143

Herz J (2015) Kalorio. www.kalorio.de, 27.2.2015

IBM, Institute of Culinary Education (2015) Cognitive Cooking with Chef Watson: Recipes for Innovation from IBM & the Institute of Culinary Education. Sourcebooks

Larsson S, Traum D (2000) Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering 6:323–340

Larsson S, Villing J (2007) The DICO project: A multimodal menu-based in-vehicle dialogue system. In: Proc 7th Intern Workshop on Computational Semantics (IWCS-7, Tilburg, The Netherlands), pp 351–354

Lison P (2014) Structured probabilistic modelling for dialogue management. PhD thesis, Dep Informatics, University of Oslo

Morbini F, Audhkhasi K, Sagae K, Artstein R, Can D, Georgiou P, Narayanan S, Leuski A, Traum D (2013) Which ASR should I choose for my dialogue system? In: Proc SIGDIAL 2013, Metz, France, 22-24 August 2013, pp 394 – 403

OpenDial (2015) Opendial toolkit. www.opendial-toolkit.net, 9.7.2015

Schäfer U, Arnold F, Ostermann S, Reifers S (2013) Ingredients and recipe for a robust mobile speech-enabled cooking assistant for german. In: KI 2013: Advances in Artificial Intelligence, no. 8077 in Lecture Notes in Computer Science (LNCS), Springer, pp 212–223

SousChef (2015) Cook Something Amazing Today. www.acaciatreesoftware.com/, 9.7.2015

Williams JD (2008) The best of both worlds: unifying conventional dialog systems and POMDPs. In: Proc. of the 9th Intern Speech Communication Association (Interspeech 2008), Brisbane, Australia, pp 1173–1176

Wolf KI, Goetze S, Wallhoff F (2015a) Cooco, what can I cook today? In: Kendall-Morwick J (ed) 23. Int Conf on Case-Based Reasoning ICCBR 2015, Workshop Proc, pp 229–236

Wolf KI, Goetze S, Wellmann J, Winneke A, Wallhoff F (2015b) Concept of a nutrition consultant application with context based speech recognition. In: Proc Workshop Kognitive Systeme 2015, Bielefeld

Zbeida M, Goldsmitz R, Shimony T, Vardi H, Naggan L, Shahar DR (2014) Mediterranean diet and functional indicators among older adults in non-mediterranean and mediterranean countries. The Journal of Nutrition, Health & Aging, Springer 18(4):411–418