Cooking Coach Spoken/Multimodal Dialogue Systems

Romain Laroche, Orange Labs, France, romain.laroche@orange.com
Jan Dziekan, Orange Labs, Poland, jan.dziekan@telekomunikacja.pl
Laurent Roussarie, Orange Labs, France, laurent.roussarie@orange.com
Piotr Baczyk, Orange Labs, Poland, piotr.baczyk@orange.com

Abstract

The use of a cookbook is quite awkward. Your working space is dirty, overloaded with kitchen utensils, your hands are probably busy, dirty or both and it is difficult for your eyes to keep track of the step you are working on. As a result, you keep going back and forth between the work surface and the cookbook, you lose time, track and may even forget a component as an unfortunate result of the mess.

Spoken dialogue systems are relevant for helping in a cooking task. Indeed, it can be performed without any device. This demonstration shows and describes Cooking Coach, two dialogue systems (one vocal and one multimodal) helping the user to search for a recipe, to check that she has all the ingredients and to prepare the recipe.

1 Spoken Dialogue System

Section 1 describes the spoken version of Cooking Coach. See Section 2 for the multimodal version.

1.1 Architecture

The architecture is illustrated by Figure 1. An Android client has been developed in order to use the tablet or mobile sound capture and restitution. The role of this client is to replace the Voice Platform that is commonly used for phone-based Interactive Response Services. We interface with the Dragon Mobile SDK, developed by Nuance, in order to get Voice Activity Detection, Automatic Speech Recognition and Text-To-Speech. On the other end, the dialogue application is deployed on an Application Server, which may have access to Information Systems if required.

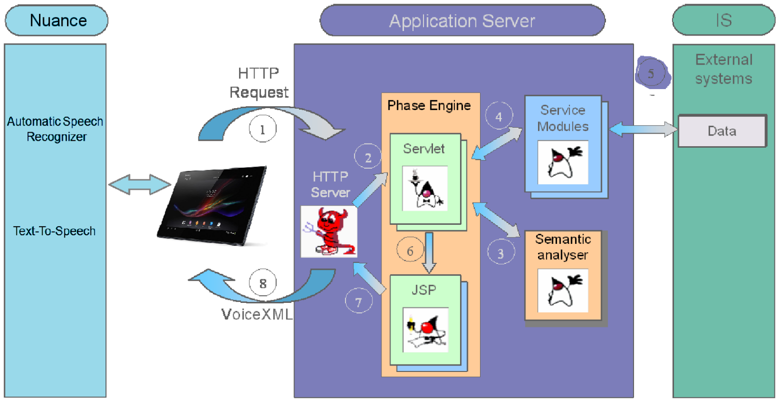


Figure 1: Architecture of the Cooking Coach dialogue system

A dialogue turn is typically processed as follows. Once Dragon Mobile has translated the audio signal into text, the Android client sends an HTTP request to the application server. An HTTP Server receives it and communicates its content to the Phase Engine's servlets. These latter asks the Semantic Analyser to interpret it and then processes the resulting request according to the dialogue logic. The servlets may load user libraries, which in turn, may have access to Information Systems. Once the servlets have planned the system's reaction, it calls a Java Server Page that generates the VoiceXML page to communicate to the client through the HTTP Server. The client parses the VoiceXML file, accesses the Nuance Server to synthesize the text (Text-To-Speech), plays the resulting sound to the user.

1.2 Dialogue Example

S01: Welcome to CookingCoach. What recipe do you want to prepare?

U01: I would like to cook brownies.

S02: You want to cook brownies. For How many people?

U02: For two.

S03: To prepare walnut brownies, you need ...

S04: Do you have all the ingredients?

U04: No.

S05: Which ingredient is missing?

U05: I don't have any walnut or chestnut.

S06: I can propose nutless brownies without your missing ingredients. Can you precise if you agree?

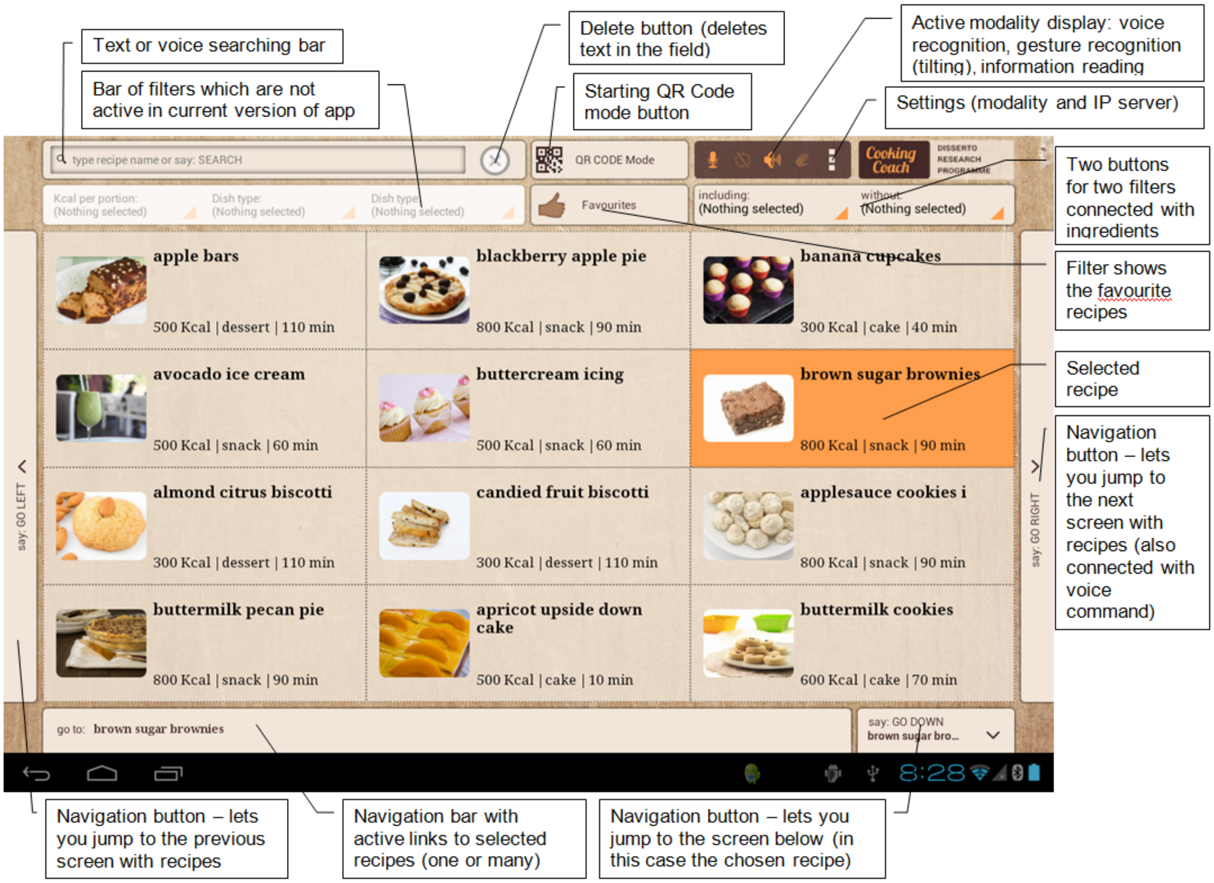
U06: Yes.

S07: Great, we will start the recipe.

S08: First, you'll need...

U08: Next/Repeat/Previous

...


Text or voice searching bar
Bar of filters which are not active in current version of app
Delete button (deletes text in the field)
Starting QR Code mode button
Active modality display: voice recognition, gesture recognition (tilting), information reading
Settings (modality and IP server)
Two buttons for two filters connected with ingredients
Filter shows the favourite recipes
Selected recipe
Navigation button – lets you jump to the next screen with recipes (also connected with voice command)
Navigation button – lets you jump to the previous screen with recipes
Navigation bar with active links to selected recipes (one or many)
Navigation button – lets you jump to the screen below (in this case the chosen recipe)
type recipe name or say: SEARCH
QR CODE Mode
Cooking Coach
DISSERTO RESEARCH PROGRAMME
Kcal per portion: (Nothing selected)
Dish type: (Nothing selected)
Dish type: (Nothing selected)
Favourites
including: (Nothing selected)
without: (Nothing selected)
say: GO LEFT
say: GO RIGHT
go to: brown sugar brownies
say: GO DOWN
brown sugar bro...
8:28

apple bars 500 Kcal | dessert | 110 min	blackberry apple pie 800 Kcal | snack | 90 min	banana cupcakes 300 Kcal | cake | 40 min
avocado ice cream 500 Kcal | snack | 60 min	buttercream icing 500 Kcal | snack | 60 min	brown sugar brownies 800 Kcal | snack | 90 min
almond citrus biscotti 300 Kcal | dessert | 110 min	candied fruit biscotti 300 Kcal | dessert | 110 min	applesauce cookies i 800 Kcal | snack | 90 min
buttermilk pecan pie 800 Kcal | snack | 90 min	apricot upside down cake 500 Kcal | cake | 10 min	buttermilk cookies 600 Kcal | cake | 70 min

Figure 2: Multimodal Cooking Coach graphical user interface.

After the use names one or several missing ingredients, the system goes back to the recipe identification, and sends the same request forbidding these ingredients. The system remembers as well the number of people. Once the recipe has been validated, the system reads the steps one by one, with the possibility for the user to navigate through the steps : go further, repeat the current step and go back to the previous one.

1.3 Database Construction

For the purpose of the demonstrator, the database has been built with a website (www.allrecipes.com) hoover. The following fields are parsed:

- Name of the recipe
- Ingredients: list of quantity, unit and name of ingredients. We implemented an smart rounding algorithm to avoid to prepare 0.67 lemon.
- Number of persons: for how many people?
- Steps: steps to perform the recipe. We had to implement a step splitting method, based on punctuation and step size.
- A bunch of information not used in the application: rating, popularity, origin, poster…

2 Multimodal Cooking Coach

The multimodal Cooking Coach (see Figure 2) follows the same architecture and dialogue logic except if offers several additional multimodal ways to interact with the system:

- Touch screen : item selection / navigation button / . . .
- QR Code : in order to select an ingredient you want in your recipe (filters the recipes with ingredients)
- Waving hand in front of the tablet : it enables to turn pages without touching your device which might be useful if your hands are dirty and the environment is too noisy to use speech recognition.

It also enables to cook several recipes in parallel.

Since a lot of effort has been made into the implementation of the multimodal prototype, we recorded two videos. The first one presents a use-case: `http://vimeo.com/62321504` and the second one shows the scope of the project with all its functionalities which have been done: `http://vimeo.com/61406788`. Password for both is: `muiDemo`.

The next steps of our prototype are to include a better search algorithm, in particular with filters amongst ingredients or type of food, and improve the personalisation of the application, *i.e.* avoid proposing recipes with ingredients that the user does not like.