

# The LIMSI ARISE System\*

*L. Lamel, S. Rosset, J.L. Gauvain, S. Bennacef,*

*M. Garnier-Rizet<sup>†</sup>, B. Prouts<sup>†</sup>*

Spoken Language Processing Group, LIMSI-CNRS, 91403 Orsay, France

<sup>†</sup>VECSYS, 3 r. de la Terre de Feu - Les Ulis, 91952 Courtabœuf, France

## ABSTRACT

The LIMSI ARISE system provides vocal access to rail travel information for main French intercity connections, including timetables, simulated fares and reservations, reductions and services. Our goal is to obtain high dialog success rates with a very open structure, where the user is free to ask any question or to provide any information at any point in time. In order to improve performance with such an open dialog strategy, we make use of implicit confirmation using the callers wording (when possible), and change to a more constrained dialog level when the dialog is not going well. The same system architecture is being used to develop a French/English prototype timetable service for the high speed trains between Paris and London.

## I. INTRODUCTION

The LIMSI ARISE (Automatic Railway Information Systems for Europe) system provides vocal access to rail travel information. The demonstration system, development of which has been partially financed by the EC through the LE-3 ARISE project, provides timetables, simulated fares and reservations, and information on reductions and services for the main French intercity connections. A prototype French/English service for the high speed trains between Paris and London is also under development.

The system providing information for the main intercity connections is largely based on the spoken language system developed for the LE-MLAP RAILTEL project[1, 7]. Callers are able to obtain information taken from the French Railways (SNCF) static timetables and additional information about services offered on the trains, fares and fare-related restrictions and reductions. The system is based on a modular architecture and contains six modules: a medium vocabulary, real-time, speaker-independent, continuous speech recognizer which transforms the acoustic signal into the most probable word sequence; a natural language understanding component which first carries out a literal understanding of the text string output by the speech recognizer, and then reinterprets the query in the context of the ongoing dialog producing a seman-

tic frame; a mixed-initiative dialog manager who, with its goal of providing information to the user, ensures communication between the user and the DBMS; a database retrieval component which accesses the RIHO database provided by SNCF to extract the requested information using a pseudo-SQL request generated from the semantic frame; a generation component which outputs a natural language response based on the dialog state, the caller's query, and the information returned from database access; and a synthesizer which concatenates speech units corresponding to the generated text string, producing very natural sounding synthetic speech.

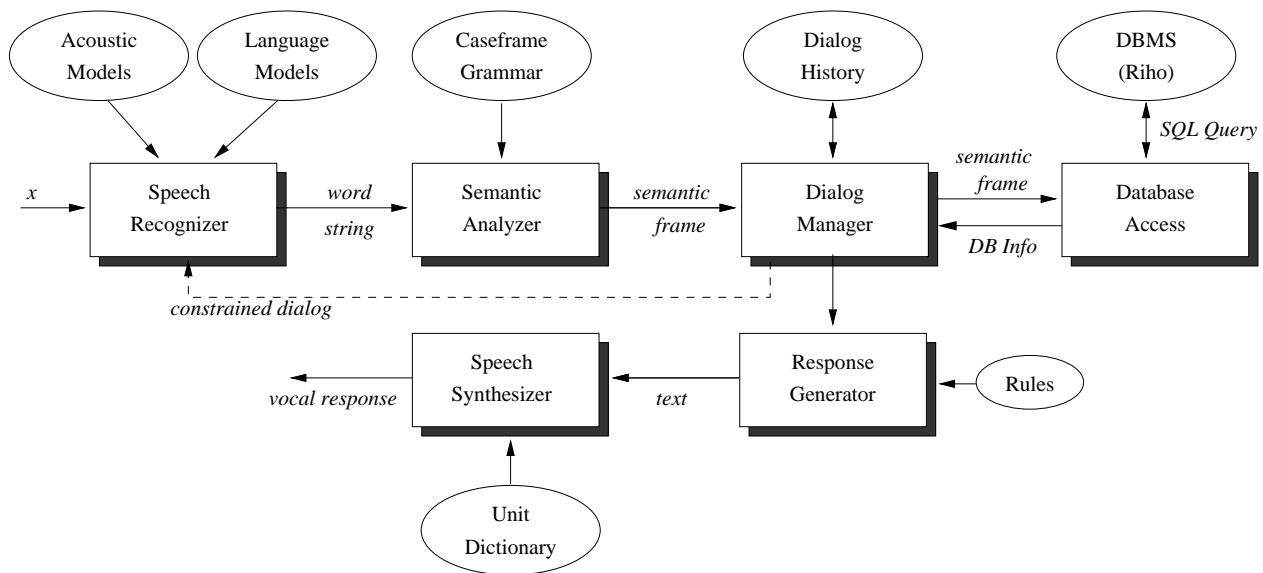
Compared to our RAILTEL system, the main advances in ARISE are in dialog management, confidence measures, an optional spell mode for city/station names, and in the inclusion of a barge-in capability which allows a more natural interaction between the user and the machine. In the next section we provide an overview the system. Section III focuses on the dialog strategy, highlighting recent advancements. In Section IV we report on ongoing user trials carried out with recruited subjects as well as punctual evaluations carried out by the SNCF.

## II. SYSTEM OVERVIEW

The main components of the spoken language system are shown in Figure 1. The system runs on a Unix workstation with a telephone interface. The speaker independent, continuous speech recognizer[3] has a recognition vocabulary of 1800 words, including about 500 station names. Speaker independence is achieved by using acoustic models which have been trained on speech data from a large number of representative speakers, covering a wide variety of accents and voice qualities. The recognizer uses continuous density HMM with Gaussian mixture for acoustic modeling and *n-gram* backoff language models[5]. Context-dependent phone models are used to account for allophonic variation observed in different contextual environments. The *n-gram* statistics are estimated on the transcriptions of spoken queries. Since the amount of language model training data is small, some grammatical classes (such as cities, days, months, etc) are used to provide more robust estimates of the *n-gram* probabilities. In order to reduce the number of understanding errors

---

\*This work was partially financed by the LE-3 project 4223 ARISE.



**Figure 1:** Overview of the ARISE system for spoken language information retrieval.  $x$  is the input speech signal.  $w$  is the word sequence output by the speech recognizer.

due to speech recognition, a confidence score is associated with each hypothesized word. If the score is below an empirically determined threshold, the hypothesized word is marked as uncertain. The natural language understanding component can choose to ignore these uncertain words and the dialog manager can use these to start clarification subdialogs. On average, rejection tends to lead to a longer dialog, since some correct words are ignored, however the overall dialog success rate is improved. An optional spell mode is authorized for station names, to improve recognition performance with a large number of cities since this is critical for the task. In our current implementation the output of the speech recognizer is the best word sequence with a confidence score, however, the recognizer is also able to provide a word lattice.

The text string output by the recognizer is passed to the natural language understanding component. Speech understanding consists of literal understanding followed by a reinterpretation in the context of the ongoing dialog. In literal understanding, the semantic analyzer carries out a caseframe analysis to determine the meaning of the query, and builds an appropriate semantic frame representation[2]. Keywords are used to select an appropriate case structure for the sentence without attempting to carry out a complete syntactic analysis. The major work in developing the understanding component is defining the concepts that are meaningful for the task and the appropriate keywords. The concepts needed to carry out the main ARISE ticketing task concern train times, connections, fares and reservations (including reductions and other constraints). Other concepts are used to handle general information available about reductions and services. The concepts have been determined by analysis of queries in the training corpora.

Contextual understanding consists of interpreting the utterance in the context of the ongoing dialog, taking into account common sense and task domain knowledge. The semantic frames resulting from literal understanding are reinterpreted using default value rules. At the same time qualitative values are transformed into quantitative ones. Semantic frames corresponding to the current utterance are then completed using the *dialog history* in order to take into account all the information previously given by the user, as well as the questions posed by the system.

The dialog manager then either prompts the user to fill in missing information or uses the semantic frame to generate an SQL-like request to the database management system. We require the caller to supply four key items: the departure and arrival stations, the date and approximate time of travel. The day and time can be specified exactly (March 14th) or in a relative manner, such as *next Monday*, *early morning*, *late tomorrow afternoon*. Interpretative and history management rules are applied prior to generation of the DBMS request. These rules are used to determine if the query contains new information, and if so, if this information is in contradictory with what the system has previously understood. If a contradiction is detected, the dialog manager may choose to keep the original information, replace it with the new information, or enter into a confirmation or clarification subdialog.

The database retrieval component uses a copy of the static SNCF train information (database RIHO). Post-processing rules are used to interpret the returned information prior to presentation to the user, taking into account the dialog history and the content of the most recent query. The generation component converts a generation semantic frame into a natural language response, which is played to the user. The form of the natural language re-

S: *Quels renseignements désirez-vous ? (What information do you want?)*  
 U: Je voudrais l'horaire du premier train de Paris à Grenoble vendredi prochain (I'd like to know the time of the first train from Paris to Grenoble next Friday)  
 S: *Veillez patienter... (Please hold on....)*  
 S: *Vendredi 26 septembre, le premier train part à 6 heures 48 et arrive à 9 heures 47 (Friday, September 26th, the first train leaves Paris at 6:48 and arrives in Grenoble at 9:47)*  
 U: et un peu plus tard? (and a little later?)  
 S: *Vous avez un train partant de Paris à 7 heures qui arrive à Grenoble à 10 heures 59 avec un changement à Lyon-Part-Dieu le 26 septembre. Qu'en pensez-vous ? (There is a train leaving Paris at 7:00 the 26th of September which arrives in Grenoble at 10:59, with a change at Lyon-Part-Dieu. Is this ok?)*  
 U: Est-ce qu'il y a un bar dans ce train? (Is there a snack bar on this train?)  
 S: *Oui, vous avez un bar dans ce train. (Yes, there is a snack bar on this train.)*  
 U: Bon, je réserve. (Good. I'd like to reserve.)  
 S: *En quelle classe voulez-vous voyager ? (What class are you traveling in?)*  
 U: en seconde classe (in second class)  
 S: *Zone fumeur ou non fumeur ? (Smoking or non-smoking?)*  
 U: fumeur, si possible (smoking, if possible)  
 S: *Vous avez réservé en seconde classe dans le train partant de Paris à 7 heures, arrivant à Grenoble à 10 heures 59 avec un changement à Lyon-Part-Dieu le vendredi 26 septembre. Au revoir, la SNCF vous souhaite un bon voyage. (You have a second class reservation Friday September 26, on the train leaving Paris at 7:00, arriving in Grenoble at 10:59, with a change of trains at Lyon-Part-Dieu. Goodbye, the SNCF wishes you a good trip.)*

**Figure 2:** Example dialog illustrating highly user-initiated dialog.

sponse depends on the dialog context, and whether or not the same information was already presented to the user. Our aim is to give a direct response to the caller, highlighting the new information. Careful attention has been paid to construction of sentences that contain the appropriate information and the generation of natural-sounding utterances[1]. Messages are synthesized by concatenation of variable-sized speech units stored in the form of a dictionary[6]. The resulting synthetic speech is rated as very natural by users.

A barge-in capability is often considered to be very important for usability. The ability to interrupt the system required modifications to several modules. Primarily, recording and speech recognition must be active at all times, even when the system is synthesizing a response. Software-based echo cancellation, applied to the recorded signal using the known synthesized signal is used to detect if the caller is speaking. When speech is detected, synthesis is stopped. In order to be able to assess the use of this capability, the time and location in the partially played response is noted. Although not yet taken into account, the dialog manager can make use of this partial response. There are also dialog situations in which barge-in is disabled to ensure that the caller hears the entire message.

### III. DIALOG STRATEGY

The dialog strategy was substantially modified from that of our RAILTEL system[1]. In order to be closer to a real dialog situation, representatives from LIMSI and VECSYS visited the Douai SNCF Information Service to

observe how the human-human dialogs are performed and what strategies are used by the human operators. The following main objectives were the basis for the new dialog strategy:

- *To never let the user get lost.* This is a very important rule in dialog systems. The user must always be informed of what the system has understood. Considering that most users will be unfamiliar with talking to a machine, it is important that that s/he is kept aware of what the system has understood.

- *To use direct responses to user questions.* After analyzing dialogs with our system it was apparent that the responses of the system needed to be more accurate and should answer directly to the user's request.

- *To give to the user the opportunity, at each step, to correct the system.* This capability is needed to be able to correct for recognition errors, but also the user may correct what s/he said or change their mind.

- *To avoid misunderstanding.* Even though users should be able to correct the system at any moment, we have observed that they have a tendency to not do so. Therefore it is important to minimize recognition errors, as users can not be expected to correct the system. This was the motivation for rejecting unreliable hypothesized words.

Although not an objective, the SNCF indicated that the overall interaction was too long, and requested that the dialog be shortened.

S: *What information do you want?*  
 U: I want a train to Paris.  
 S: *So, **you want to go to Paris**, where are you leaving from?*  
 U: from Bordeaux...  
 S: ***From Bordeaux-St-Jean to Paris**, what day are you traveling?*  
 U: tomorrow morning  
 S: ***In the morning**, there is a direct train from Bordeaux-St-Jean to Paris at 10:22 arriving at 13:40, on the 21st of September. Is this ok?*  
 U: and the next train?  
 S: ***The next train** leaves 10:42 and arrives at 13:45.*  
 U: what is the price of a first class ticket?  
 S: ***The price of a first class ticket** from Bordeaux-St-Jean to Paris is 549 francs.*  
 U: Ok, I'd like to reserve on this train.  
 S: *Smoking or non-smoking?*  
 U: non smoking, please  
 S: *You have a reservation in first class on the 10:42 train direct from Bordeaux-St-Jean to Paris, arriving at 13:45, Sunday Septembre 21st. Goodbye, the SNCF wishes you a good trip.*

**Figure 3:** Example dialog illustrating immediate feedback of what has been understood by the system. The relevant part of the system response is shown in bold.

S: *Quels renseignements désirez-vous? (What information do you want?)*  
 U: des horaires de train (timetable information)  
 S: *Oui, de quelle ville partez-vous? (Ok, where are you leaving from?)*  
 U: Paris  
 S: *Bon, vous partez de Paris, quelle est votre destination? (You are leaving from Paris, what is your destination?)*  
 U: non (recognition error: user said Dinan)  
 S: *Je suis d'esolée, je n'arrive pas à comprendre, indiquez votre destination, par exemple Lille. (I'm sorry, I did not understand. Indicate your destination, for example Lille)*  
 U: Di- (user hesitation results in word fragment)  
**Constrained dialog**  
 S: *Donnez votre ville d'arrivée et épelez la si vous le souhaitez. Par exemple Lille L I 2 L E. (Give your arrival city and spell it if you want, for example Lille L I L L E)*  
 U: Dinan D I N A N  
 S: *Alors, de Paris à Dinan quel jour partez vous? (From Paris to Dinan, what day are you leaving?)*  
 ...

**Figure 4:** Excerpt illustrating a constrained dialog for the arrival city.

The dialog has been modified in several ways. The responses given by the system are much shorter, so as to keep the caller in tighter contact with the system, and to make for a more natural dialog. In the RAILTEL system information was given for up to 3 trains. Now the system proposes only one train, the one corresponding most closely to the caller's request. The user is given the opportunity to ask for another train (the preceding/following one, an earlier/later train, the first/last train). Constraint relaxation is used in retrieving timetable information in order to provide a more cooperative dialogue and response. For example, the system relaxes constraints on the departure time when no train corresponds to the user's request, allowing the system to return the closest train after or before the specified time. If the user has specified a time

range (e.g., early morning) only one train is proposed to the user, that closest to the middle of the specified time. Yes/no answers are handled by the system, which means that it can make use of explicit confirmation. Implicit confirmation is used by including new information supplied by the caller in the system response. This immediate feedback, as illustrated by the dialog in Figure 3 allows the user to know what the system has understood. An optional spell mode has been added for station names. The system lets the user know if it is unable to treat their request. An example response to an out-of-functionality request is: "Je suis désolée mais vous ne pouvez pas réserver pour plus d'une personne" (I am sorry, but you cannot reserve for more than one person).

Closely related to the changes in dialog management, the

generation component has been modified to produce texts which directly integrate information in the user's request. These responses aim to provide only the information that is new and/or of high relevance. In previous versions, the responses, while assessed favorably by subjects, tended to recapitulate too much information and were thus too long and repetitive. The synthesis dictionary unit was extended to the larger vocabulary and completed with additional formulations. New recordings were made in accordance with the changes to the generation component. Questionnaires completed by callers indicate that the new generation strategy is well accepted.

A two-level dialog strategy has been implemented, in which a mixed-initiative dialog where the user can provide any information at any point in time is combined with a system-directed dialog in case a problem is detected in obtaining departure and arrival station names and the date of travel. When the second level, or constrained dialog is active, the speech recognizer makes use of a dialog-state dependent language model. An example of a dialog which is largely user-initiated is shown in Figure 2. The final part of the dialog (concerning the reservation) is guided by the system.

Implicit confirmation plays an important role in keeping the user aware of what the system has understood. A constrained dialog can be initiated by the system if the user does not respond to the system prompt for one of the four basic items (timeout), or in cases where the information received by the system is contradictory with what was previously understood. Such constrained dialogs apply only to the departure and arrival cities, and the travel date. For example, if the system understands a change in the departure or arrival city, one of the following strategies is used depending upon the state of the dialog: the system may choose to ignore the information; it can ask for an explicit confirmation of the new city; or it can ask the user to repeat the information. If the caller changes one of these items during the confirmation request, implicit confirmation is used in the following system prompt. The directiveness of the prompt increases if the user does not supply the requested information. An example of a constrained dialog for the arrival city name is given in Figure 4.

The results of our assessment of the modified dialog strategy indicate that the overall dialog time is reduced, and the caller assessment is positive.

#### IV. USER TRIALS

We recruit callers on an ongoing basis, providing data for system development and evaluation. In addition to these user trials, the prototype system undergoes periodic evaluations carried by the SNCF (the French national railways, a partner in the ARISE project) in order to assess usability and performance.

LIMSI subjects are recruited via advertisements in local newspapers. Each subject receives a set of instructions, and calls the system 10 times. In 4 of the calls the user is given a predefined scenario to solve, and in the remaining 6 calls the subject is asked to obtain whatever information s/he would like to have. For consultation, subjects are given a randomly selected subset of the stations known to the system. At the start of each call, the subject enters a DTMF code to allow for easy tracking and processing. After each call, the caller is required to complete an evaluation sheet and return it LIMSI. In addition to a qualitative assessment, the caller provides the information obtained from the system, and is free to give any additional comments.

The periodic tests carried out by the SNCF involve subjects recruited by a hostess at a Parisian train station. The subjects were asked to test a new, but experimental service, and were given a gift certificate for their participation. Each subject called the system three times, carrying out an open scenario that s/he wrote down prior to each call. Subjects completed a short questionnaire after each call and a longer one after the final call. Despite the differences in recruitment, the general characteristics of these calls (in terms of dialog success, overall call duration, the number of exchanges, vocabulary, types of requests and typical problems) are essentially the same as those made by the subjects we recruit.

Here we report results on 50 dialogs recorded during a two-day period (May 27-28), just prior to the last SNCF user tests held in June98. The results are based on essentially all the calls recorded during this two-day period, removing only calls with no queries. There were an average of 19 exchanges per call, with an average total duration of 229 seconds. The dialog error rate in obtaining timetable information was 20% (in 26% of the dialogs some difficulty was experienced in obtaining supplementary information).

On a set of 58 calls from the June SNCF user trials<sup>1</sup> the dialog error in obtaining timetable information was 16%. The reservation failure rate was 11% (reservations require specifying the class of travel, seating preference and reduction). A higher error rate (30%) was obtained for diverse questions, due in part to functionality limitations. 12% of the dialogs ended without a closing formality (ie. the caller hung up) without saying goodbye.

These results are substantially better than the results of the Nov97 SNCF evaluation before most of the modifications presented in this paper were implemented. On 80 calls the timetable information failure rate was 47.5% and the reservation failure rate was 35.7%. In November, 52% of the calls were terminated without a closing formality.

<sup>1</sup> The calls from June 3rd are not used here, due to experimental problems, such as the subject speaking with the experimenter, or interference due to simultaneous recording. Results are given for calls from June 4 and 5, 1998.

The June calls are longer, averaging 15 exchanges (167 seconds), compared to 10 exchanges in November (114 seconds). Although more performant, the two-level dialog has increased the length of the dialog. The assessment by SNCF subjects has improved from a satisfaction level of 5.9 to 12.7 (out of 20).

We have carried out an analysis of the use of the barge-in capability on the 58 calls of the June98 SNCF test set. The callers were aware that they could interrupt the system if they so desired. Users interrupted the system in 72% (42) of the calls, speaking during 122 of 958 system responses (13%). When barge-in was observed during a call, it was used to interrupt, on average, 3 responses. Barge-in was observed in a variety of contexts, most frequently used to respond to questions before they were finished. For example, if the system is uncertain about a station name, the caller prompted to say and optionally spell the city name. Almost 40% of the interruptions followed this type of prompt. In almost 25% of the instances, the system responded after a long pause (the endpoint was assumed) or after a time-out, but the subject seemed to be engrossed in their thoughts, and unaware of what the system was saying. In contrast to our expectations, barge-in was only rarely used (6% of the cases) to correct the system, and usually to change the date of travel.

## V. DISCUSSION

Enabling efficient, yet user-friendly interaction for access to stored information by telephone is quite difficult. Most existing services are quite directive, restricting the caller to use limited formulations and to provide only the requested information. Some laboratory prototypes allow a more open, user-initiated dialog, but performance is generally lower than what can be obtained with more restricted dialog structures.

Our goal is to obtain high dialog success rates with a very open structure, where the user is free to ask any question or to provide any information at any point in time. In order to improve performance with such an open dialog strategy, we make use of implicit confirmation using the callers wording (when possible). Our preliminary observations of a barge-in capability, judged to be very important for usability, indicate that it is not heavily used, and is not used in the manner we had anticipated (to correct misrecognized items). This may be partially due to the experimental conditions, as callers do not really need the information they are asking for, and therefore may not notice (or care about) the errors.

An important issue that was highlighted during the SNCF user trials is that users do not distinguish the functionalities of the service from the system responses. Although the system was able to detect some out-of-functionality requests, and responded that it was unable to handle these, such responses are not satisfactory for users.

For example, a caller that wanted to reserve 3 seats was informed that he could only reserve one seat at a time. From the spoken language system developer's point of view, this response is correct, although the user may not agree. In such cases we may have a successful dialog, but an unhappy caller.

## ACKNOWLEDGMENT

We would like to thank the SNCF for providing the information database RIHO for use in the ARISE project.

## REFERENCES

- [1] S.K. Bennacef, L. Devillers, S. Rosset, L.F. Lamel, "Dialog in the RAILTEL Telephone-Based System," *ICSLP'96*, pp. 550-553, Oct. 1996.
- [2] S.K. Bennacef, H. Bonneau-Maynard, J.L. Gauvain, L.F. Lamel, W. Minker, "A Spoken Language System For Information Retrieval," *ICSLP'94*, Yokohama, Japan, Oct. 1994.
- [3] J.L. Gauvain, L.F. Lamel, G. Adda, M. Adda-Decker "Speaker-Independent Continuous Speech Dictation," *Speech Communication*, **15**, pp. 21-37, Sept. 1994.
- [4] J.L. Gauvain, L. Lamel, "Large Vocabulary Continuous Speech Recognition: from Laboratory Systems towards Real-World Applications," *Institute of Electronics, Information and Communication Engineers*, J79-D-II:2005-2021, December 1996.
- [5] S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. ASSP*, **35**(3), 1987.
- [6] L.F. Lamel, J.L. Gauvain, B. Prouts, C. Bouhier, R. Boesch, "Generation and Synthesis of Broadcast Messages," *Proc. ESCA-NATO Workshop on Applications of Speech Technology*, Lautrach, Germany, Sept. 1993.
- [7] L. Lamel, S.K. Bennacef, S. Rosset, L. Devillers, S. Foukia, J.J. Gangolf, J.L. Gauvain, "The LIMSI RailTel System: Field trials of a Telephone Service for Rail Travel Information," *Speech Communication* **23**, pp. 67-82, Oct. 1997.