A Telephone-Based Railway Information System for Spanish: Development of a Methodology for Spoken Dialogue Design

R. San-Segundo, J.M. Montero, J.M. Gutiérrez, A. Gallardo, J.D. Romeral and J.M. Pardo
Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. UPM
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040 Madrid, Spain
{lapiz|juancho|juana|gallardo|jdromeral|pardo}@die.upm.es

Abstract

In this paper, we describe the steps carried out for developing a Railway Information Service for Spanish. This work introduces a methodology for designing dialogue managers in spoken dialogue systems for restricted domains. In this methodology several sources of information are combined: intuition, observation and simulation for defining several dialogue strategies, evaluating them and choosing the best. Users in a final evaluation gave the system a 3.9 score in a 5-point scale with an average call duration of 204 s.

1 Introduction

The important improvements in Speech Technology have allowed developing automatic systems that can work under real conditions. At this point, telephone-based spoken dialogue systems have appeared as an important field for applying these technologies. Last decade, a great amount of restricted domain systems have been developed: JUPITER (Zue, 1997) that provides weather information; travel information and ticket reservation services like RailTel (Lamel, 1997) and ARISE European projects (Lamel, 2000; Baggia, 2000). Another important spoken dialogue project is the DARPA Communicator (http://fofoca.mitre.org). Some systems developed in this project are the CU (Pellom, 2000) and the CMU (Rudnicky, 2000) Communicator systems. These services enable natural conversational interaction with telephone callers that access information about airline flights, hotels and rental cars. Other services are the Directory-Assistance systems like (Schrâmm, 2000) or the automatic call redirection system as implemented in AT&T (Riccardi, 2000). In this paper, we present a methodology for designing the dialogue manager in spoken dialogue system for restricted domains. We use this methodology to develop a train timetable information system for the main Spanish intercity connections. In this paper, we also present an approach for designing the confirmation mechanisms and incorporating user-modelling techniques.

2 Methodology for Dialogue Design

The methodology proposed in this paper consists of 5 steps. The first step is the database analysis where the information contained in the knowledge database is described by an Entity-Relationship diagram. In the design by intuition, a "brain-storming" over the E-R is performed for proposing different dialogue alternatives. The third step is the design by observation, where we evaluate each proposal using user-operator dialogue transcriptions. The next step we simulate the system with a Wizard of Oz in order to learn the specific characteristics of human-system interactions. The fifth step is the design by iterative improvement where the confirmation strategies are designed in order to implement a fully automatic system. The railway information system developed in this work uses isolated speech recognition, but the methodology is general and it can be used to develop system with continuous speech reconigiton and understanding.

This methodology is similar to the Life-Cycle Model presented in (Bernsen, 1998) and (www.disc2.dk), but we incorporate the step "design by observation" where human-human interactions are analysed and we present measures to evaluate the different design alternatives at every step of the methodology.

3 Database Analysis and Design by Intuition

The database, as the initial point in our methodology contains the information or knowledge we want to provide in our service. For example in a railway information system, this database contains the time information for all the trains, their prices, their services, etc...The aim of database analysis is to describe this information, in order to offer the service. This description consists of an Entity-Relationship Diagram (E-R) that shows a semantic representation of the data. In this diagram, the main entity sets, their attributes and keys (attributes that uniquely define an element in an entity set), and entity set relationships must be defined. We must pay attention to the entity sets because they will be the possible goals that the system can provide (parts of service such as timetable information, reservations, fares, etc.). The keys are the main mandatory items to ask the user and they define the dialogue interactions. The E-R diagram is not unique and it depends strongly on the system designer and the service.

After the database analysis, we propose a "brain-storming" (design by intuition) over the E-R to propose *alternatives*. These proposals are concerned with the goals to provide, the sequence of offering them and the items that are needed to satisfy each goal (pieces of information that must be obtained from the user, such as departure and arrival cities, departing date, etc.).

The ways the user can specify each item are also an important issue. For example, for the departure place, the user can say the station name or the city name. In the second case, the system should provide the information for all the train stations in that city.

The result of this analysis is a table or work sheet that includes all the alternatives. This table will be used at next step of the methodology to compute the frequency of each alternative concerning with the goals, the information that the system must provide, the ways the user specifies an item, etc.

4 Design by Observation

It aims at analysing user-operator dialogues in a similar service and tracking off the observed events. This design phase evaluates and measures the impact of the alternatives proposed at previous step.

4.1 Goals analysis

a) *Which goals are most frequently required by the user, and which is the sequence for asking them.*

The result of this analysis is the number of times that a specific goal is required and its position. In Table 1, the goals analysis for 100 call transcriptions is presented. Several goals can appear in the same call. In our final system, we offer the goals that appear in more than 10% calls, except *Itinerary* (there is no information about that in the database) and *Others* (it contains very different goals but none reaches 10%).

Table 1: Goals analysis.

	%	Position				
		1st	2nd	3rd	4th	5th
Timetable	**64**	57	6	1	-	-
Round trip Timetable	**20**	-	14	5	1	-
Fares	**46**	6	30	10	-	-
Reservations	**26**	14	4	2	3	3
Train frequency	2	2	-	-	-	-
Itinerary	**14**	8	4	1	1	-
Bargains	5	1	2	1	1	-
Others	**12**	8	2	-	2	-

b) *The information given by the operator to satisfy each goal.*

We annotate these data (i.e. departure and arrival times, trip duration, train type, etc.) and their importance (number of times that the operator provides each item).

4.2 Items analysis

To satisfy a goal, the system must ask the user some items. These items have to be also analysed at this step:

a) *Which items are needed to satisfy each goal and the sequence for asking them.*

b) *To classify items as mandatory or optional:*
   - *Mandatory*: it is an item needed to satisfy the goal and must be asked to the user (e.g. "What date are you departing?"). If the user does not provide this information, it must be assigned a default value (e.g. "today") or the goals have to be satisfied depending on its value (e.g. for all days in current week).
   - *Optional*: the system does not ask it to the user. When the user specifies it, the provided information should match this value.

c) *To classify each item as simple or complex.*

A complex item can be divided into several simple ones (e.g. "departure date" can be divided into DAY, MONTH and YEAR). With isolated speech recognition, each simple item needs one interaction. A continuous speech recognizer could get several simple items simultaneously.

d) *To analyse the different ways a user can specify an item value and its importance.*

Table 2 shows the results for the departure date. A high percentage of people travel on current week and a relevant percentage do not specify any date, because they just want general travel information between two cities. 9.4% of people travelled in the same month and specified the day with the day number (e.g. the 20th). 4.7% of people travelled on another month and specified the date by the name of the month and the day number.

Table 2:Analysis for the Departure Date.

Current week		**This month**	**Other month**	**Any**
today	25.0%	9.4%	4.7%	28.1%
tomorrow	15.6%			
weekday	17.2%			

e) *Item ordering and grouping.*

One step in the dialogue is a group of items that can be asked to the user without confirmation. Our recommendations are:

- To group items that are contiguous in the item sequence and have a semantic relationship between them.
- Not to group more than 4 consecutive items (without leaving simple items alone).

Just one confirmation can validate a group of items. We can also incorporate rest zones to provide help or a summary of the interaction. The steps of the dialogue impose a certain rhythm to the user-system interaction.

4.3 Negotiation analysis

There is a "negotiation" when the user has to choose one of the possible travel options:

a) *What information helps the user to take a decision and its importance.*

Table 3 shows the analysis for negotiation criteria. The number of connections is an important factor, but it does not appear frequently in user-operator dialogues because the operator only offers direct trips without connections. Train Type provides a hint about fares and the services offered in the train, generally known by the user.

Table 3: Criteria analysis for negotiation.

Criteria	%	**Criteria**	%
Depart. time	41.0	**# Connections**	2.6
Arrival time	15.4	**Connection**	2.6
Depart. station	2.6	**Class**	10.3
Fares	7.7	**Duration**	5.0
Train Type	12.8		

b) *To Choose the negotiation strategy.*

You can present the best option and let the user ask for the previous/next one, or you can present several alternatives at the same time and ask the user to choose one of them. In this case, it is important to analyse the *number of alternatives* the operator gives simultaneously (and the user can manage), during the conversation.

In the observation analysis, only human-human interactions are considered, and they can be severely different from human-system ones. Because of this, high level dialogue characteristics can be learnt, but specific behaviours when interacting with an automatic system are not detected (e.g. changes in the user speaking rate, usage of formal vs. colloquial phrases, etc.). On the other hand, this analysis permits to evaluate several alternatives without having implemented any system.

5 Design by Simulation

Now, the specific characteristics of human-system interactions are analysed by simulating the system with a Wizard of Oz approach. We must focus on the design of the dialogue flow, the questions to ask for obtaining the needed item and the information the system should provide to satisfy each goal. Dialogue alternatives are implemented and evaluated. In this evaluation, 15 users called the system, completed 6 scenarios and fill a questionnaire. The evaluating measures come from system annotations (referred as System) and from user answers in the questionnaire (referred as Questionnaire), where the user answers about subjective aspects difficult to be measured by the system. The user answers are recorded in audio files. The evaluating measures were:

5.1 Goal evaluation

a) *The measures used to validate the goal sequence and the goal coverage were:*

- System: number of times a goal is required by the user, time and number of questions (interactions) to satisfy a goal.
- Questionnaire: new goals suggested by the user.

5.2 Item evaluation

a) *To design the questions and recognition vocabulary.*

- System: number of times the user keeps in silence (does not answer the system question) and, recognition rate for each question (if a recognizer is already available).
- Questionnaire: how easy is it to specify an item value.

b) *For the item sequence evaluation.*

In some parts of the dialogue, several sequences of questions for asking the items are proposed and randomly selected in each call. The evaluation measures are:

- System: when it is possible, the Sequence Recognition Rate (SRR) computed as the product of all independent item recognition rates.

Table 4 shows the sequence analysis for the 9departure and arrival cities. The SRR difference is not significant, but we observed the following effect during user calls: when the system asks the Arrival City first, the user assumes that the system knows the Departure City and he/she gets confused when the system asks about it. This explains the great difference between the item recognition rates for the Arrival-Departure City sequence.

Table 4: Sequence evaluation for Departure and Arrival cities (item recognition rate and Sequence Recognition Rate).

Recognition Rates (%)			
Sequence	**1st item**	**2nd item**	**SRR**
Depart.-Arrival City	75.6	80.0	60.5
Arrival-Depart. City	94.3	60.0	56.6

- Questionnaire: asking the user for his/her preference.

c) *Analysis of different ways to split complex items into simple ones.*

- System: number of interactions and time to get a complex item, and the Sequence Recognition Rate of the simple items.
- Questionnaire: how easy is it to specify a complex item with the proposed sequence of interactions.

d) *How can we manage the mandatory items when the user does not specify any value for them*

We should parameterise the information by this item values or fix a default value.

- Questionnaire: to evaluate the best option, we ask the user in the questionnaire if the information given by the system exceeds his/her expectations (and it is necessary to fix a default value) or it is adequate.

5.3 Negotiation

In our case, we have decided to present several travel options at the same time and ask the user to choose one of them. Randomly, the system presents the options one by one, two by two or three by three. For the information provided per option, several patterns are designed and randomly selected for each group of options. The measures considered for evaluation were:

- System: number of questions and time for negotiation.
- Questionnaire: What information helps the user to choose and number of options he/she can manage at the same time.

Users preferred to manage 3 options at the same time (less interactions) but negotiation takes more time. We decided to keep the negotiation in a 3 by 3 basis reducing the information provided per option to Train Type plus Departure and Arrival Times.

Table 5: Negotiation analysis in the simulation step.

Negotiation analysis (User Questionnaire)		
1 by 1	**2 by 2**	**3 by 3**
21.4%	21.4%	57.2%

Negotiation Criteria				
Train Type	Depart. Time	Arrival Time	Fares	Duration
15.6%	37.5%	25.0%	15.6%	6.2%

For global dialogue evaluation, we propose the following measures:

- System: average number of interactions and time per call, and recognition rate for all the vocabularies.
- Questionnaire: the user has to evaluate in a scale how fast did the user obtain the information, how easy was to learn, which part would you change, and how do you compare the system to other ways of obtaining the service: web, going to the station, etc.

Whenever it is possible, it is better to use only the measures computed by the system because they are objective. The questionnaire can not be longer than one sheet and should only address issues that cannot be resolved by automatic system annotations, such as subjective evaluations or default values for mandatory items. If we want to include many questions, it is better to design several questionnaires that evaluate different aspects. Every user should fill only one of the questionnaires proposed. About the scenarios to complete, it is important that the users can define their own scenarios for testing a greater diversity of situations. Some scenarios should be imposed, in order to have enough data to evaluate a specific situation.

The Wizard of Oz permits us to analyse different dialogue designs without any system implemented. Moreover, the system does not need to confirm the item values, so we can make independent designs for the dialogue and for the confirmation mechanisms.

On the other hand, the main problem in this step is the difficulty for simulating an automatic system, especially the answer time. A person spends more time when selecting the item values than an automatic system. It is necessary to develop tools that help the Wizard to simulate the system in a realistic way.

6 Design by Iterative Improvement

At this step, we implement the first version of a fully automatic system in an iterative modify-and-test process. With the Wizard of Oz, a first version of the dialogue flow was defined and it was necessary to design confirmation mechanisms that take into account the recognition confidence: an analysis of confidence measures is required.

6.1 Confirmation Strategies

Depending on the number of items to confirm:

- One item . ("Have you said Madrid?")
- Several items. ("Do you want to go from Madrid to Sevilla?")

Depending on the possibility to correct (Lavelle, 1999):

- *Explicit confirmation*: the system confirms one or several item values through a direct question. ("I understood you want to depart from Madrid. Is that correct?")
- *Implicit confirmation*: the system does not permit the user a correction, it only reports about the recognition result. ("You want to leave from Madrid. Where are you arriving at?")
- *Semi-implicit confirmation*: it is similar to the implicit confirmation, but the user can correct ("You want to leave from Madrid. In case of error, say correct, otherwise, indicate your arrival city").
- *No confirmation*: the system does not provides feedback about the recognized value (i.e. in yes/no questions).
- *Item value rejection* and repeat question: when the confidence is low, the system does not present the value to the user and repeats the question ("Sorry, I can not understand. Where are you departing from?").

6.2 Confidence Measures in recognition

Confidence measures in recognition are vey useful for designing the confirmations (Sturm, 1999). The recognition module used in our system is a large vocabulary telephone speech recognizer, that can recognize isolate words and simple expressions such as "On monday", "Next week" or "In the morning". The recognizer (Macías, 2000a) is based on a hypothesis-verification approach. The best features for confidence annotation are concerned with the verification step and are based on (Macías, 2000b):

- *First Candidate Score:* acoustic score of the best verification candidate.
- *Candidate Score Difference:* difference of acoustic score between the 1st and 2nd verification candidates.
- *Candidate Score Mean and Variance:* average score and variance over the 10 best candidate names.
- *Score Ratio:* difference between the score of the phone sequence (hypothesis stage) and the score of the best candidate word (verification stage).

All the features are divided by the number of frames. We have considered a Multi-Layer Perceptron (MLP) to combine the features in order to obtain a unique confidence value. In this case we use the features directly as inputs to the MLP. With this solution, a preprocessing is required to limit the dynamic range of each measure to the (0,1) interval. Here, the normalization scales the features using the minimum and maximum value obtained for each measure in the training set. The hidden layer contains of 10 units and one output node was used to model the word confidence. During weight estimation, a target value of 1 is assigned when the decoder correctly recognizes the name and a value of 0, when incorrect recognition occurs. The database used in the confidence experiments is built with the results obtained in the recognizer evaluation for a 1,100 name dictionary. In this case we have 2,204 examples, considering 1,450 for training the MLP, 370 as the evaluation set and 370 for testing. We have repeated it six times providing a 6-Round Robin training to verify the results. In this paper we present the average results of these experiments. A 39.1% of wrong recognized words are detected at 5% false rejection rate (Figure 1), reducing the minimum classification error from 15.8% (recognition error rate) to 14.0%.

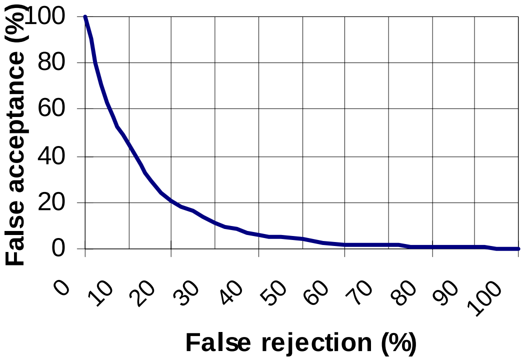

Figure 1. False acceptance vs False Rejection.

6.3 Confirmation Mechanism design

For designing the confirmation mechanism, it is necessary to plot the correct words and errors distributions as a function of the confidence value and to define different confidence thresholds.

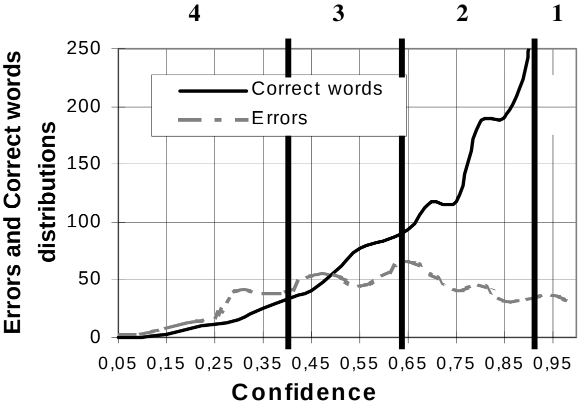

Figure 2. Bottom detail for Errors and Correct words distributions vs. confidence.

We have defined 4 levels (3 thresholds) of confidence (Figure 1):

1. *Very High Confidence:* the number of correctly recognized words is much higher than the number of errors.
2. *High Confidence:* the number of correctly recognized words is higher than the number of errors.
3. *Low Confidence:* both distributions are similar. The system is not sure about the correctness of the recognized word.
4. *Very Low Confidence:* in this case, there are much more errors than correctly recognized words, so we reject the recognized word and the system must ask again.

For the departure and arrival cities step, we define CL(D) and CL(A) as departure/arrival city Confidence Levels (CL). Depending on the CL, we implemented the following confirmation strategies:

- CL(D)=1 and CL(A)=1: *implicit confirmation of both items.* "You want to travel from Madrid to Sevilla, When do you want to leave?"
- CL(D)=2 or CL(A)=2: *explicit confirmation of both items.* "Do you want a Madrid-Sevilla trip?"
- CL(D)=3 or CL(A)=3: *explicit confirmation of each item.* "Do you want to travel from Madrid?"
- CL(D)=4 or CL(A)=4: *rejection of each item.* "Sorry, I do not understand. Where are you departing from?"

Under hard conditions, the system asks the user to spell the city name (San-Segundo, 2000). In this design, we have not considered the semi-implicit confirmation because it is not very friendly for the user. In Section 5.5, we describe the CORRECT command that permits the same functionality without increasing the prompt length.

6.4 Confirmation Mechanism Evaluation

For the iterative improvement, it is necessary to evaluate the confirmation mechanisms in order to adjust them when we changes the dialogue. The proposed measures are:

- *For Implicit confirmation*: the recognition rate obtained for each item when comparing the recorded audio files and the word that was recognized as 1st candidate.
- *For Semi-implicit confirmation*: number of times the user wants to correct. If this

number is high, we should change to an explicit confirmation; otherwise, we should use an implicit confirmation strategy.

- *For Explicit confirmation*: number of times the user denies the system proposal. If this number is low we could think on relaxing the confirmation strategy to an implicit one.

For all confirmations, we can compute the number of questions and time needed to confirm the item values and the number of times the user deny the system proposal. To analyze the impact of the confirmation mechanisms in the dialogue speed, the system can compute the percentage of implicit vs. explicit confirmations (see Table 6).

6.5 Error recovering mechanisms

When an automatic system uses implicit confirmations, it is necessary to define some mechanisms to permit the user to recover from system errors:

START OVER: this command permits the user to start from scratch. Instead of resetting all the items, our system begins confirming groups of items explicitly (dialogue steps). When one group is not confirmed, the system starts from that point:
S: "The selected option is an Intercity train... "
U: "start over"
S: "Let us start the call. Do you want to go from Madrid to Barcelona?"
U: "yes"
S: "Do you want to travel on 19th of July?"
U: "No"
S: "Do you want to leave this week, next week or later?"

CORRECT: When the system makes a mistake and takes a wrong item value as right (in an implicit confirmation), the user can correct the system by saying this command at any point of the dialogue. In this case, the system asks again the last introduced item.
When considering these commands, some aspects must be kept in mind:

- The selected commands must be very different from the words contained in the recognition vocabularies, but it must be very friendly to the user.
- The sentence used by the system to report to the user the recognition of these commands has to be carefully designed to permit a smooth transition in the dialogue.
- It is necessary to specify the dialogue points where the system should report to the user about these possibilities: e.g. in the initial help and in the help provided when the system detects problems in the interaction.

When one of these commands is recognized, the confirmation strategies have to be more pessimist about the confidence because one error, recognizing these commands, can produces delays in the interaction.

6.6 User-Modelling

The user modelling technique applied is based on (Veldhuijzen van Zanten, 1999) using 4 user skill levels. Depending on current level, the prompt sentences are clearer (they contain more information about how the user should answer, possible values, etc...) or the system provides more or less information per time unit. The levels considered are:

- **1st level.** The prompts explain *how to interact* with the system, the *asked item*, the possible *accepted values* and *how to specify* one of the them (for the period of the day: "Please *speak after the tone*. Say the *period of the day* you want to travel in; in the morning, in the afternoon or in the evening.").
- **2nd level.** The prompts include the *item* needed, the *accepted values* for it and the *way to specify them*. ("Say the *period of the day* you want to travel in; *in the morning, in the afternoon or in the evening*.")
- **3rd level.** Only the required item is included in the prompt. ("Say the *period of the day* you want to travel in.")
- **4th level.** The user knows everything (the accepted values and the way to specify one of them) and we can relax the question ("When do you want to leave?").

Current level depends on the initial state, the number of errors and positive confirmations along the interaction. In our case, the system starts at the 2nd level (after providing a general explanation about how to interact with the system). When several errors (or positive confirmations) occur, the system decreases (or increases) the level. The number of errors or positive confirmations that forces a change depends on current level. Thus, the system adapts dynamically the interaction to the user skill, making more explicit questions when recognition errors occur.

Example:
[The system is in the level 3]

S: "Say the period of the day you want to travel in."
U: "After lunch"
[The system recognizes "in the evening"]
S: "Have you said in the evening?"
U: "No"
[The system decreases the level from 3 to 2]
S: "Say the period of the day you want to travel in; in the morning, in the afternoon or in the evening."
U: "In the afternoon"

When the user makes several scenarios in the same call, the system could start the new scenario one level higher than the level at the end of the previous one. This is possible because when the user has done a complete scenario it means that the user has answered sucessfully the system questions once.

For simplicity, we have not made the confirmation strategies to depend on the user ability level but it could be possible redefining the confirmation mechanism in each level.

Considering this user-modeling technique, one interesting measure to evaluate the adaptability of the system are the average ability level during the interaction (see Table 6).

7 Field Evaluation

In this evaluation, 30 users called the system for completing 4 scenarios (120 calls). The evaluating measures come from the system annotations (table 6) and from user answers in a questionnaire (table 7).

Table 6. Measures calculated by the system.

Measure	Value
Call duration (seconds)	204
Number of questions per call	21.2
% of implicit confirmations	61.3
Number of START OVER commands	0.08
Number of CORRECT commands	0.43
Average User-Modeling level along the calls	1.95
Duration of Negotiation(seconds)	58

35.4% of the calls asked information of the two legs in a round-trip and 31.5% completed the reservation for single or round-trips. As we can see, average call duration is 204 seconds, higher than in the operator based service (152 seconds), but similar to other automatic services (Baggia, 2000). About the recognition rates, we got more than 95% for small vocabularies (less than 50 words/expressions: weekdays, period of the day...). For departure and arrival cities (770-words vocabulary), we obtained 90.1% recognition rate for in-vocabulary cities and not rejected answers. 25% of the wrong cases were solved with the second candidate and 36% with the spelled name recogniser. For the remaining cases, more interactions were necessary.

In these experiments, we got 4.1% out-of-vocabulary cities, detecting 32% of the cases with the spelled name recognizer. For the remaining 68% cases, the user hung up after several trials.

We asked the user for his/her preference when obtaining train information: 75.4% of the people preferred the system, 24.6% preferred web access and nobody preferred to go to the ticket office.

Table 7. Measures (out of 5) obtained from the questionnaire.

Measure	Score
User experience in these kind of systems.	1.8
The system understands what I say.	3.6
I understand what the systems says.	4.5
I get train information fast.	4.0
The system is easy to learn.	3.9
In case of error the correction was easy.	3.1
The system asks me in a logical order.	4.6
Generally, it is a good system.	3.9

The dialogue point with more problems was the departing date specification, because we used isolated speech recognition and it needs several interactions to get a date. System intelligibility obtained a score of 4.5 (out of 5) due to our restricted-domain female-voice synthesis (Montero, 2000). The dialogue flow obtained the best score (4.6) because of the detailed analysis performed.

8 Conclusions and Future Work

In this work, we propose a new methodology for designing dialogue managers in automatic telephone-based spoken services. This methodology has been successfully applied to a train information system for Spanish. A combination of several sources of information is proposed: intuition, observation and simulation for defining and evaluating several dialogue strategies, and choosing the best one. The first steps are database analysis (E-R diagram) and design by intuition, where a "brain-storming" over the E-R is performed for proposing different dialogue alternatives and for defining an evaluation table. In design by observation, we evaluate each proposal using user-operator dialogue transcriptions, without having any system implemented. The limitation of the observation step is that human-human interactions are different from human-system ones. This problem is solved by the Wizard of Oz, that simulates the human-system interaction.

In design by iterative improvement, we describe an approach to incorporate recognition confidence measures for defining and managing the confirmation mechanisms. For all vocabularies, this approach obtained a recognition rate higher than 90%. Two mechanisms for error recovering are described: Start-Over and Correct. User-modelling techniques are incorporated for adapting the system dynamically to the user ability.

With this proposed methodology, we have implemented a fully automatic system with a good user acceptability: mean call duration was 204 s, similar to (Baggia, 2000). The users validated the applicability and usability of the system giving a general score of 3.9 (out of 5).

In a future work, we will apply this methodology to develop automatic systems with continuous speech recognition and understanding modules.

9 Acknowledgements

This work has been partially supported by the grant 2FD1997-1062-C02 (EU and Spanish CICYT). Authors want to thank the contributions of colleagues at GTH and RENFE railway company.

References

Baggia, P., Castagneri, G., Danieli, M., 2000. "Field trails of the italian ARISE train timetable system". Speech Communication. Vol. 31, pp. 356-368.

Bernsen, N.O. Dykjaer, H. and Daykjaer, L. 1998 "Designing interactive speech systems. From first ideas to user testing" Springer Verlag.

Lamel, L.F., Bennacef, S.K., Rosset, S., Devillers, L., Foukia, S., Gangolf, J.J., Gauvain, J.L., 1997. "The LIMSI RailTel system: Field trial of a telephone service for rail travel information". Speech Communication. Vol. 23, pp. 67-82.

Lamel, L., Rosset, S., Gauvain, J.L., Bennacef, S., Garnier-Rizet, H., Prouts, B., 2000. "The LIMSI ARISE system". Speech Communication. Vol. 31, pp 339-355.

Lavelle, A.C., Calmes, H., Pérennov, G., 1999."Confirmation strategies to improve correction rates in a telephonic inquiry dialogue system". Proc. of EUROSPEECH, Budapest, Hungary. Vol. 3, pp. 1399-1402.

Macías-Guarasa, J., Ferreiros, J. Colás, J., Gallardo, A., and Pardo. JM., 2000a."Improved Variable List Preselection List Length Estimation Using NNs in a Large Vocabulary Telephone Speech Recognition System". Proc. of ICSLP, Beijing, China. Vol. II, pp. 823-826.

Macías-Guarasa, J., Ferreiros, J., San-Segundo, R., Montero, JM., and Pardo, JM., 2000b. "Acoustical and Lexical Based Confidence Measures for a Very Large Vocabulary Telephone Speech Hypothesis-Verification System". Proc. of ICSLP. Beijing, China. Vol. IV, pp. 446-449.

Montero, J.M., Córdoba, R., Vallejo, J.A., Gutiérrez-Arriola, J., Enríquez, E., Pardo, J.M., 2000."Restricted-domain female-voice synthesis in Spanish: from database design to a prosodic modeling". Proc. of ICSLP. Beijing, China.

Pellom, B., Ward, W., Pradhan, S., 2000. "The CU Communicator: An Architecture for Dialogue Systems". Proc. of ICSLP, Beijing, China.

Riccardi, G., Gorin, A., 2000. "Stochatic Language Adaptation over time and state in natural spoken dialog systems". IEEE Trans. on Speech and Audio Processing, Vol 8, pp. 3-10.

Rudnicky, A., Bennet, C., Black, A., Chotomongcol, A., Lenzo, K., Oh, A., Singh, R. 2000. "Task and domain specific modelling in the Carnegie Mellon Communicator system". Proc. of ICSLP, Beijing, China.

San-Segundo, R., Colás, J., Ferreiros, J., Macías-Guarasa, J., Pardo, J.M., 2000. "Spanish Recognizer of continuously spelled names over the telephone". Proc. of ICSLP, Beijing, China.

Schrâmm, H., Rueber, B., and Kellner, A., 2000. "Strategies for name recognition in automatic directory assistance systems". Speech Communication. Vol 31, No 4 pp. 329-338.

Sturm, J., den Os, E., and Boves, L., 1999 “Issues in Spoken Dialogue Systems: Experiences with the Dutch ARISE System”. Proceedings of ESCA Workshop on Interactive Dialogue in MultiModal Systems. Kloter Irsee, Germany, 1-4.

Veldhuijzen van Zanten, G., "User modelling in adaptive dialogue management". 1999. Proc. of EUROSPEECH, Budapest, Hungary. Vol. 3, pp. 1183-1186.

Zue, V., Seneff, S., Glass, J., Hetherington, L., Hurley, E., Meng, H., Pao, C., Polifroni, J., Schloming, R., Schmid, P., 1997. "From interface to content: transclingual access and delivery of on-line information". Proc. of EUROSPEECH, Athenas, Greece.