Assessment of Spoken Dialogue Systems by Simulating Different Levels of User Cooperativeness

Ramon López-Cózar[1], Gonzalo Espejo[1], Zoraida Callejas[1], Ana Gutiérrez[1], David Griol[2]

[1] Dept. of Languages and Computer Systems, University of Granada, Spain
{rlopezc, zoraida}@ugr.es, gonzaep@correo.ugr,es
gutcasan@gmail.com

[2] Dept. of Computer Science, University Carlos III de Madrid, Spain
dgriol@inf.uc3m.es

Abstract. This paper proposes a user simulation technique to test the performance of spoken dialogue systems. The technique is based on a novel approach to simulating different levels of user cooperativeness, which allows to carry out a more detailed system assessment. In the experiments we have evaluated a spoken dialogue system designed for the fast food domain. The evaluation has focused on the performance of the speech recogniser, semantic analyser and dialogue manager of this system. The results show that the technique provides relevant information to obtain a solid evaluation of the system, enabling us to find problems in these modules which cannot be observed taking into account an optimal cooperativeness level.

1 Introduction

Continuous advances in the field of spoken dialogue systems make the processes of design, implementation and evaluation of these systems more and more complex. To solve this problem, a technique that has attracted increasing interest in the last decade is based on the automatic generation of dialogues between the dialogue system and an additional module, called *user simulator*, which represents user interactions with the dialogue system. The user simulator makes it possible to generate a high number of dialogues in a very simple way. Therefore, it reduces the time and effort required for the evaluation of a dialogue system each time the system is modified.

Research in techniques for user modelling has a long history within the fields of language processing and spoken dialogue systems. The main objective in the latter is to construct a representative user model that describes his state during the interaction with the system. According to the level of abstraction at which the dialogue is modelled, it is possible to find in the literature user simulation techniques working at the word level [1, 2, 3] or the intention level [4, 5]. The implementation of these techniques is typically carried out using either statistical approaches [4, 6] or rule-based methods [1, 8, 9].

The main purpose of the user model is not to capture the individual characteristics of a specific user, but to form the basis of a user simulation tool which is representative of diverse user behaviours. With this goal, in this paper we present a technique to enhance a rule-based user simulator previously developed by including different levels of user cooperativeness. Providing a fine-grained scale of user cooperativeness makes it possible to carry out a detailed evaluation of the performance of a spoken dialogue system, assessing its ability to deal with responses that do not necessary match every system prompt.

Our study has been evaluated by means of a Spanish dialogue system called Saplen [10], designed to answer customers queries and register product orders in a fast-food restaurant. Our user simulator has been used to improve the system by identifying problems in the performance of the speech recogniser, semantic analyser and dialogue manager. Moreover, the evaluation results provide valuable information about how to best tune the dialogue management strategies and language models for speech recognition to meet the needs of real users

The remainder of the paper is organised as follows. Section 2 introduces the initial version of our user simulation technique and its application to the Saplen dialogue system. Section 3 presents the changes made in this simulator to create an enhanced version of it which implements the proposed technique. Sections 4 and 5 summarize the results of its evaluation and the analysis of the dialogues generated by means of its interaction in the Saplen domain. Finally, section 6 presents the conclusions and section 7 outlines possibilities for future work.

2 The initial user simulator

In a previous study [1] we developed a user simulator which is the basis for the simulator proposed in this paper. The purpose of the initial simulator was to interact automatically with a spoken dialogue system in order to create a corpus of dialogues that could be used for testing the performance of the system. We carried out experiments with this simulator employing the Saplen dialogue system, previously designed in our lab to provide fast food information and register product orders [10]. As can be observed in Figure 1, the simulator receives the current prompt generated by the dialogue system as well as the frame(s) obtained from the analysis of the previous response of the simulator. Each response of the simulator is an utterance (voice samples file) recorded by a client of the fast food restaurant, which is taken from a speech database.

The interaction between the simulator and the dialogue system is carried out by means of a set of scenarios that indicate the goals the simulator must try to achieve during the interaction. For example, a scenario may specify that the simulator must order one ham sandwich, one large beer and one chocolate milkshake, and provide the telephone number, postal code and address of a user. After the simulator has selected the appropriate scenario goal, e.g., <POSTAL_CODE> = “18001”, it retrieves an utterance from the speech database for which the associated frame matches the selected goal, e.g., “my postal code is 18001”. This utterance is the response of the simulator.

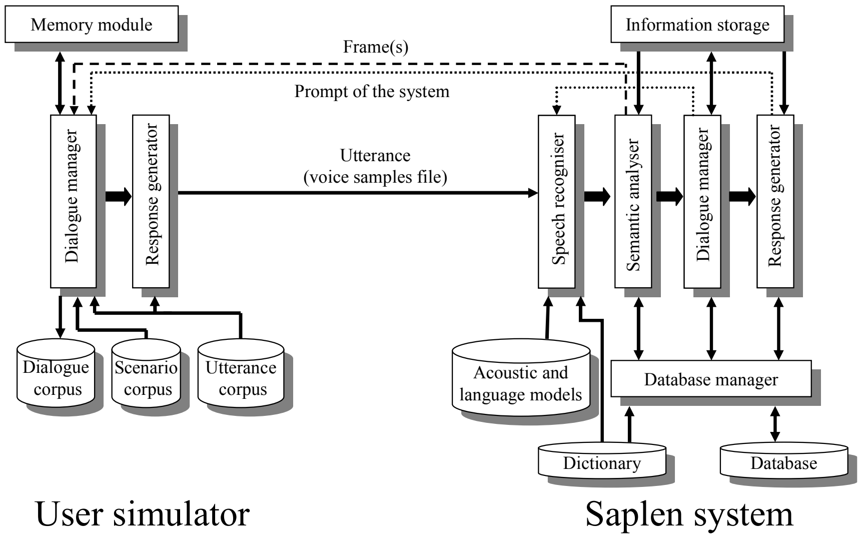

Fig. 1. Interaction between the initial version of the user simulator and the Saplen system.

In this version of the simulator, the generated responses always fulfilled the system expectations, because the level of user cooperativeness was the highest in all cases. For example, if the system prompted for the user's address, the simulator provided an address; if the system prompted for the user's telephone number, the simulator provided a telephone number and so on, as can be observed in the following sample dialogue:

(1) Saplen: What would you like to have?
(2) User simulator : Three ham sandwiches
(3) Saplen: I have to confirm your order. Did you say six ham sandwiches? Please say yes or no.
(4) User simulator : No
(5) Saplen: OK. How many sandwiches did you say?
(6) User simulator : Three
(7) Saplen: OK. So that's three ham sandwiches. Please say your telephone number.
(8) User simulator : 9 5 8 2 7 5 3 6 0
(9) Saplen: I need to confirm your telephone number. Did you say 9, 5, 8, 2, 7, 5, 3, 6, 0? Please answer yes or no.
(10) User simulator : Yes
(11) Saplen: OK. Telephone number 9, 5, 8, 2, 7, 5, 3, 6, 0. Please say your postal code.
(12) User simulator: 1 8 0 0 1
(13) Saplen: OK. Postal code 1, 8, 0, 0, 1. Please say your address, for example Elm street, number 13, second floor, apartment letter e.

(14) User simulator : Andalucia street number 58 first floor letter e
(15) Saplen: I need to confirm your address. Did you say Andalucia street, number 58, first floor letter e? Please answer yes or no.
(16) User simulator : Yes
(17) Saplen: OK. Address Andalucia street, number 58, first floor letter e. Would you like to drink something?
(18) User simulator : No
(19) Saplen: OK. I need to confirm your order. I think you ordered three ham sandwiches. Is this correct? Please answer yes or no.
(20) User simulator : Yes
(21) Saplen: OK. The price is 12 euros. Is this OK for you? Please answer yes or no.
(22) User simulator : Yes
(23) Saplen: OK. The delivery service will take 30 minutes approximately to get to your address. Is this OK for you? Please answer yes or no.
(24) User simulator : Yes
(25) Saplen: OK. Thank you very much for using our automatic telephone service. Good night.

An advantage of this kind of interaction between the simulator and the dialogue system is that it makes it possible to estimate the system performance interacting with users who behave "optimally" as they provide exactly the data requested by the system in each interaction. However, its main drawback is that this behaviour may not be very realistic.

3 The enhanced user simulator

In order to address this drawback of the initial simulator, we have developed an *enhanced user simulator* which shows a more realistic behaviour. To do so, this simulator implements three cooperativeness levels: *low*, *medium* and *high*.

The *high* level corresponds to the behaviour of the initial user simulator. To implement the *medium* cooperativeness the simulator does not always respond with the kind of utterance requested by the dialogue system, but always generates responses that are appropriate for the system prompts. For example, the simulator may repeat a piece of data when the system prompts to confirm the data using a *yes*/*no* response. This behaviour can be observed in the following sample dialogue:

(1) Saplen: What would you like to have?
(2) User simulator: Three ham sandwiches
(3) Saplen: I have to confirm your order. Did you say six ham sandwiches? Please say yes or no.
(4) User simulator: Three
(5) Saplen: OK. So that's three ham sandwiches. Please say your telephone number.

This behaviour of the simulator can be useful to model experienced users who know how to make the interaction go faster by providing in advance the data that the system will require in subsequent prompts, e.g. uttering directly 'Three' instead of 'No' in turn (4). Simulating this type of user makes it possible to evaluate the ability and "intelligence" of the dialogue system to deal appropriately with responses that do not match every system prompt exactly.

To implement the *low* cooperativeness the simulator generates appropriate responses for some system prompts, but for others it generates responses that are completely inappropriate. This behaviour can be observed in the following sample dialogue:

(1) Saplen: What would you like to have?
(2) User simulator: Andalucia street number 58 first floor letter e
(3) Saplen: I need to confirm your address. Did you say Andalucia street, number 58, first floor letter e? Please answer yes or no.
(4) User simulator: Yes
(5) Saplen: OK. Address Andalucia street, number 58, first floor letter e. Would you like anything to eat?

The main advantage of the enhanced user simulator is that it makes it possible to model real users who because of inexperience or being nervous may answer some prompts with unexpected utterances. Therefore, this type of simulation allows us to test the robustness of the speech recogniser and semantic analyser to deal with these utterances. In addition, it can be useful to test the ability of the dialogue manager to appropriately handle data items that are not expected. For example, if the address data were provided unexpectedly in advance, the dialogue manager should not prompt for these data later.

4 Experiments

The goal of the experiments is to study whether the dialogues generated employing the user simulator provide a better basis for the system assessment than the initial simulator.

The evaluation is carried out in terms of word accuracy (WA), sentence understanding (SU) and task completion (TC). SU is the proportion of correctly understood sentences regardless of the possible speech recognition errors. In other words, we say that there is sentence understanding if the semantic representation obtained by the semantic analyser is correct even though some words might have been incorrectly recognised.

TC is the proportion of successful dialogues, i.e., the percentage of dialogues in which the simulator achieves all the scenario goals. In order to avoid excessively long dialogues between the system and the simulator, which would not be accepted by real users, the simulator cancels the interaction with the system if the total number of turns (i.e. of system plus user simulator turns) exceeds a threshold set to 30 turns. Cancelled dialogues are not considered successful and thus decrease the TC rate.

4.1 Speech database and scenario corpus

To carry out the experiments we have employed a speech database that we collected in a fast food restaurant and contains around 800 dialogues between clients and restaurant assistants. The database is comprised of 18 sentence types: product orders, telephone numbers, postal codes, addresses, queries, confirmations, amounts, food names, ingredients, drink names, sizes, flavours, temperatures, street names, building numbers, building floors, apartment letters, and error indications.

Selecting at random 5,500 client utterances among the 18 sentence types in the database, we have created two utterance corpora, one for training the language models and the other for testing them, ensuring that no training utterances were included in the testing corpus. Both corpora include the orthographic transcriptions of the utterances as well as their corresponding semantic representations (frames). One half of the utterances that the simulator employs to correct system errors and to confirm data are used for training and the other half are used for testing. These utterances are not used as scenario goals given that they are scenario-independent.

To automatically generate dialogues between the Saplen system and the proposed user simulator we have designed 50 scenarios. The scenario goals are selected by choosing frames at random in the test utterance corpus corresponding to product orders, telephone numbers, postal codes and addresses.

Employing the simulator we have generated 20 dialogues per each scenario, cooperativeness level and type of language model for speech recognition, which makes a total of 20 x 50 x 3 x 2 = 6,000 dialogues. These dialogues have been saved in log files for further evaluation.

4.2 Language models for speech recognition

The Saplen system was configured to use two different kinds of language model for speech recognition: one based on 17 prompt-dependent language models (PDLMs), in the form of word bigrams [11], whilst the other was based on one prompt-independent language model (PILM), also a word bigram. Both kinds of language model have been used in previous studies [1, 12].

The problem with the PDLMs is that they provide very poor recognition results if the users respond to system prompts with a type of utterance that does not match the active grammar (e.g. an address when the system prompted for a telephone number). This happens because the utterances are analysed employing a grammar compiled from utterances of a different type. Therefore, this language model is not appropriate if we want to provide users with a natural interaction that enables them to answer system prompts with utterances that do not strictly match what the system requires, which is something that they would probably do when interacting with a human operator.

Contrary to what happens with the PDLMs, the PILM permits the recognition of any kind of utterance within the domain, which helps to provide users with more flexible interaction. However, the accuracy is in general lower than with the PDLMs given that the vocabulary is much larger and there are many more types of utterance to be considered.

4.3 Results

Table 1 shows the average results obtained for the three levels of user cooperativeness and the two language models for speech recognition. As can be observed, better performance is achieved for higher cooperativeness levels regardless of the language model employed. The differences in the performance are more clearly observed when the PDLMs are employed. When the cooperativeness is *high* the simulator always provides responses that match the current system prompt. Therefore using the PDLMs each utterance is analysed employing the appropriate recognition grammar.

The scores decrease when the level of cooperativeness is *medium* or *low* given that for these conditions the simulator sometimes provides utterances that do not match the current system prompt. According to the results set out in the table, it can be said that the system should only employ the PDLMs if the cooperativeness of real users were *high*, thus achieving TC = 70.56%. For the other cooperativeness levels the performance would be very poor.

Table 1. Evaluation results (in %) for the three levels of user cooperativeness employing PDLMs and PILM (1 = High cooperativeness, 2 = Medium cooperativeness, 3 = Low cooperativeness).

	PDMLs			PILM		
Cooperat. level	WA	SU	TC	WA	SU	TC
1	90.05	85.18	70.56	75.40	57.86	11.67
2	70.56	70.76	21.67	76.60	55.71	5.56
3	43.69	56.82	11.13	77.87	53.28	4.47

When the PILM is employed the values for WA and SU are similar for the three level of cooperativeness. The reason is that regardless of the level, the simulator responses always match the recognition grammar, as it is compiled from training utterances permitted for all the system prompts. Taking into account the results set out in the table, it can be said that the system performance is totally unacceptable for the PILM regardless of the cooperativeness level, since TC is 11.67% as the greatest.

5 Discussion of results

The main objective of developing the user simulation is to carry out an evaluation of the Saplen system in order to identify problems with the speech recogniser, semantic analyser or dialogue manager, to fix those and thus increase the system's robustness to deal with a variety of users. To do this we have considered the log files created during the dialogue between the simulator and the Saplen system, have focused on the dialogues with very low values for the evaluation measures, and have analysed these to find the reasons for the unacceptable system performance.

5.1 Findings for the *high* cooperativeness

When the PDLMs are employed, the utterances that the simulator generates as responses always match the active speech recognition grammars, which cause WA to be quite high (90.05%). The 10% word error rate is caused by three factors. One is that some 'Yes/No' answers to confirmation prompts are misrecognised, e.g. the word 'sí' (yes) is sometimes substituted by the word 'te'. Another reason is that there are problems recognising some addresses for which not all data items are recognised. The third reason is that there are many recognition errors if the speakers have strong southern Spanish accents, as they usually do not pronounce the final 's' of plural words, wich causes the recognition of the singular form of substantives and adjectives instead of plurals. Given that these errors in the number correspondence do not affect the semantics of the utterances, most of the product orders are correctly understood even though some words are incorrectly recognised. The average TC employing the PDLMs is 70.56% which suggests that the system performance can be considered more or less acceptable for real users with a *high* level of cooperativeness.

When the PILM is employed, there are also many speech recognition errors in the responses to system confirmation prompts, especially if these are uttered by speakers with strong southern Spanish accents. Given that these users omit the final 's' of plural words, as discussed above, because of acoustic similarity the word 'no' is often substituted by the word 'dos' (two), 'uno' (one) or 'error', whereas the word 'sí' (yes) is often substituted by the word 'seis' (six). Moreover in many cases the words 'sí' and 'no' are discarded by the semantic analyser of the system as their confidence scores are smaller than the lower confidence threshold employed (set to 0.3). Because of these problems there are many repetitive confirmation turns to get data confirmed, which lengthens the dialogues and causes some of these to be cancelled as the interaction limit (30 turns) is reached before all the scenario goals are achieved.

5.2 Findings for the *medium* cooperativeness

When the PDLMs are employed the WA for the *medium* cooperativeness is lower than for the *high* cooperativeness (70.56% vs. 90.05%). The reason is that in addition to facing the problems discussed in the previous section, in this case the Saplen system has to overcome the problem that the sentences uttered to answer confirmation prompts are not permitted by the active grammars.

If the cooperativeness is *medium*, the simulator answers confirmation prompts by repeating the data that the dialogue system is trying to confirm, although it always prompts for a 'Yes/No' response. The problem identified in the analysis is that the grammar employed to recognise responses to confirmation prompts was initially created considering only users who would utter either a confirmation, a negation or an error indication (i.e. high cooperativeness). Because of this, the system confirmation strategy employing the PDLMs fails for the medium cooperativeness as the responses of the simulator are not permitted by the grammar. On the contrary, product orders, telephone numbers, postal codes and addresses are more or less well understood, although the errors in gender/number correspondences and those for some addresses discussed above also occur in these dialogues. As a consequence of all the problems the average TC employing the PDLMs is 21.67%, which is obviously too low to consider the system performance acceptable for real users behaving with *medium* cooperativeness.

When the PILM is employed WA is 76.6%, which is very similar to that obtained for the *low* cooperativeness (77.87%) employing the same language model. The reason for this low rate is the large amount of errors in the recognition of responses to confirmation prompts (e.g. 'no' substituted by 'dos', and 'sí' substituted by 'seis'), and also in gender/number correspondences (e.g. 'verdes' (green) substituted by 'verde'). As commented above, these errors happen especially when the words are uttered by speakers with strong southern Spanish accents. Analysing the dialogues we observe that the affirmative and negative responses are sometimes discarded by the semantic analyser of the system as their confidence scores are smaller than the lower confidence threshold (set to 0.3), provoking the repetition of confirmation turns. Also the same problems detected for the *high* cooperativeness with the recognition of some addresses are found for the *medium* cooperativeness, which means that the system needs to employ extra turns to get and confirm all the data items in the addresses.

In addition we observe a problem in the confirmation strategy that is not observed for the *high* cooperativeness and that is particularly noticeable in the confirmation of telephone numbers. If the cooperativeness is *medium* the simulator confirms this data by repeating the telephone number instead of generating the 'Yes/No' response requested by the system. To have a telephone number correctly understood, the Saplen system requires on the one hand that all its digits are recognised with confidence scores greater than the higher confidence threshold (set to 0.5). On the other hand, the system requires an implicit confirmation from the user when it includes the recognised number if the prompt is to get the postal code.

According to the method employed to assign confidence scores to frame slots, the confidence score of a slot that contains a telephone number is the lowest confidence score of the digits. For example, the confidence score for the recognition hypothesis "nine (0.5684) five (0.9652) eight (0.5647) one (0.5894) two (0.6954) three (0.9654) three (0.4362) four (0.6584) five (0.5898)" would be 0.4362.

Because of all these factors, the system has problems confirming some telephone numbers, especially when these are uttered by speakers with strong southern Spanish accents. The reason is that employing a telephone number to confirm a telephone number tends to require another confirmation, given that it is likely that at least one digit is misrecognised or recognised with low confidence. This problem provokes repetitive dialogue turns that lengthens the dialogues, causing some of these to be cancelled as the interaction limit is reached.

5.3 Findings for the *low* cooperativeness

When the PDLMs are employed, the WA is very low (43.69%). One reason for this is the high number of errors in the confirmation turns given that these users do not answer confirmation prompts with 'Yes/No' responses but they repeat the data the system is trying to confirm, as it happened with the *medium* cooperativeness. Another reason is that for these users the simulator selects at random the kind of utterance to answer system prompts to enter product orders, telephone numbers, postal codes or addresses. This decreases the average TC employing the PDLMs to 11.13%, which is obviously too low to consider the system performance acceptable for real users with low cooperativeness.

When the PILM is employed, the value of WA (77.87%) is very similar to that obtained for the other cooperativeness levels (75.4% and 76.6%), given that in the three cases the same kind of language model is employed throughout the whole dialogue regardless of the system prompt. Consequently, the SU rate (53.28%) is also similar to that for the other cooperativeness levels (57.86% and 55.71%).

As discussed before, the behaviour of the simulator for the *low* cooperativeness is very similar to that for the *medium* cooperativeness, with the difference that the former features a random selection of utterances to answer system prompts to enter product orders, telephone numbers, postal codes and addresses. Because of this difference, the interaction for the *low* cooperativeness reveals a problem in the semantic analyser of the Saplen system which is not observed for the other two levels: in some cases telephone numbers are correctly recognised but are understood as postal codes, while postal codes are correctly recognised but understood as telephone numbers.

The reason is that the system employs its current prompt to differentiate between both kinds of utterance. Therefore, when it prompts to get a telephone number, it considers that the recognised sequence of digits is a telephone number. Similarly, when it prompts for a postal code, it assumes that the sequence is a postal code, and when it prompts for a building number, it considers that the digit sequence is a building number.

This simple understanding method works well if the cooperativeness level is *high* or *medium* and the simulator produces the expected kind of utterance. However, if the cooperativeness level is *low*, the simulator may answer prompts to enter a telephone number with a product order, telephone number, postal code or address, which causes the possible confusion if the postal code is randomly selected.

6 Conclusions

In this paper we have presented a user simulation technique that enables assessing spoken dialogue systems by employing different levels of user cooperativeness. The technique has been employed to test the performance of the Saplen system using two front-ends for speech recognition: one based on 17 prompt-dependent language models (PDLMs) and the other based on one prompt-independent language model (PILM).

The results obtained in terms of word accuracy (WA), sentence understanding (SU) and task completion (TC) show that the system should only employ the PDLMs if the cooperativeness of the real users were *high* since the system performance can be considered more or less acceptable (TC = 70,56%). The results also show that when the PILM is used the system performance is too poor for an interaction with real users regardless of the type of language model employed, since TC is too low (only 11.67% in the best case).

Analysis of the log files created during the interaction between the simulator and the system let us detect problems with the speech recogniser, semantic analyser and dialogue manager of the system. The main problem with the speech recogniser is the difficulty in correctly recognising some words uttered by speakers with strong southern Spanish accents, as was observed in the experiments that we made with the initial version of the simulator.

As an attempt to solve this problem we would try to replace some incorrect recognition hypotheses taking into account contextual information. This might be interesting as we observe that when the PILM is employed and the simulator answers confirmation prompts, the word ‘sí’ (yes) is often substituted by the word ‘seis’ (six), while the word ‘no’ is often substituted by the word ‘dos’. Therefore, one possible solution to this problem is to employ the PDLMs for this prompt type, and force the user say ‘Yes’ or ‘No’ as no other words would be recognised.

The main problem with the semantic analyser is the incorrect understanding of telephone numbers and postal codes when the cooperativeness is *low*, given that in some cases recognised telephone numbers are understood as postal codes and vice versa. This problem was not observed in the experiments that we carried out with the initial version of the simulator, which shows the advantage of using the proposed enhanced user simulator.

To solve this problem it could be possible to include knowledge in the semantic rules about the different format of telephone numbers (nine digits) and postal codes (five digits). This way, the semantic analyser could guess whether the utterance is a telephone number or a postal code regardless of the prompt, and the system could ask the user to confirm the guess if the utterance type does not match the current prompt.

The main problem with the dialogue manager is the repetitive confirmation turns due to an inefficient system confirmation strategy, which happens especially when the user cooperativeness is *medium* or *low*. Again, this problem was not observed in the experiments that we carried out with the initial version of the simulator. These repetitive confirmation turns would not be accepted by real users as they might think the system has many problems understanding them and is very inefficient because it requires a lot of attempts to get data confirmed. To solve this problem it could be possible to implement an improved confirmation strategy that changes the prompt automatically if the system needs to repeat the confirmation turns.

7 Future work

Future work to improve the proposed technique includes studying alternative methods to simulate more precisely the behaviour of real users. One possibility would be to set the level of user cooperativeness dynamically as the dialogue evolves. In the current set up this selection is made beforehand and the setting remains fixed throughout all the dialogue. A different strategy would be to consider that a real user may change his cooperativeness depending on the success of the interaction. For example, the cooperativeness of the simulator could be set to *low* at the beginning of the dialogue and it could be changed to *medium* or *high* dynamically as long as the system restricts the interaction freedom as an attempt to recover from understanding problems.

We also plan to enable the simulator's ability to model changes of mind by the user. In our application domain these changes may be related to modifications in the ordered products, which will be useful to test the system functionality that handles the product orders.

References

1. López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A. J., Sánchez, V.: Assessment of dialogue systems by means of a new simulation technique. Speech Communication, 40(3) (2003) 387–407
2. Araki, M., Watanabe, T., Doshita, S.: Evaluating dialogue strategies for recovering from misunderstandings. Proc. of IJCAI Workshop on Collaboration Cooperation and Conflict in Dialogue Systems (1997) 13–18
3. Pietquin, O., Beaufort, R.: Comparing ASR modeling methods for spoken dialogue simulation and optimal strategy learning. Proc. of Interspeech (2005) 861–864
4. Griol, D., Hurtado, L. F., Segarra, E. Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50 (2008) 666-682
5. Jung, S., Lee, C., Kim, K., Jeong, M., Geunbae, L.: Data-driven user simulation for automated evaluation of spoken dialogue systems. Computer Speech and Language 23 (2009) 479-509
6. Schatzmann, J., Weilhammer, K., Stuttle, M.,Young. S.: A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. Knowledge Engineering Review 21 (2006) 97-126
7. Chung, G. Developing a flexible spoken dialog system using simulation. Proc. of 42nd Annual Meeting of the ACL (2004) 63–70
8. Lin, B. S., Lee, L. S.: Computer aided analysis and design for spoken dialogue systems based on quantitative simulations. IEEE Trans Speech Audio Process 9(5) (2001) 534–548
9. Filisko, E., Seneff, S.: Learning Decision Models in Spoken Dialogue Systems Via User Simulation. Proc. of AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems (2006)
10. López-Cózar, R., García, P., Díaz, J., Rubio, A. J.: A voice activated dialogue system for fast-food restaurant applications. Proc. of Eurospeech (1997) 1783–1786
11. Rabiner, L., Juang, B. H.: Fundamentals of speech recognition, Prentice-Hall (1993)
12. López-Cózar, R., Callejas, Z.: Combining language models in the input interface of a spoken dialogue system. Computer Speech and Language 20 (2005) 420–440