Chapter 18
Development and Integration of Speech Technology into COurseware for Language Learning: The DISCO Project

Helmer Strik, Joost van Doremalen, Jozef Colpaert, and Catia Cucchiarini

18.1 Introduction

Language learners seem to learn best in one-on-one interactive learning situations in which they receive optimal corrective feedback. The two sigma benefit demonstrated by Bloom [1] has provided further support for the advantages of one-on-one tutoring relative to classroom instruction. However, one-on-one tutoring by trained language instructors is costly and therefore not feasible for the majority of language learners. In the classroom, providing individual corrective feedback is not always possible, mainly due to lack of time. This particularly applies to oral proficiency, where corrective feedback has to be provided immediately after the utterance has been spoken, thus making it even more difficult to provide sufficient practice in the classroom.

The emergence of Computer Assisted Language Learning (CALL) systems that make use of Automatic Speech Recognition (ASR) seems to offer new perspectives for training oral proficiency. These systems can potentially offer extra learning time and material, specific feedback on individual errors and the possibility to simulate realistic interaction in a private and stress-free environment. For pronunciation training, systems have been developed that either provide overall scores of pronunciation performance or try to diagnose specific pronunciation errors

H. Strik (✉) · J. van Doremalen · C. Cucchiarini
CLST, Radboud University, Erasmusplein 1, 6525 HT, Nijmegen, The Netherlands
e-mail: W.Strik@let.ru.nl; j.vandoremalen@let.ru.nl; c.cucchiarini@let.ru.nl

J. Colpaert
Linguapolis, University of Antwerpen, Antwerp, Belgium
e-mail: jozef.colpaert@ua.ac.be

P. Spyns and J. Odijk (eds.), *Essential Speech and Language Technology for Dutch*,
Theory and Applications of Natural Language Processing,
DOI 10.1007/978-3-642-30910-6_18, © The Author(s) 2013

[12, 14–16, 18, 22]; commercial systems are e.g., marketed by Digital Publishing,[1] Auralog,[2] and Rosetta Stone.[3] However, the level of accuracy achieved in signaling pronunciation errors to the learners is not always satisfactory [16].

Research at the Radboud University of Nijmegen has shown that a properly designed ASR-based CALL system is capable of detecting pronunciation errors and of providing comprehensible corrective feedback on pronunciation with satisfactory levels of accuracy [3]. This system, called Dutch-CAPT (Computer Assisted Pronunciation Training), was designed to provide corrective feedback on a selected number of speech sounds that had appeared to be problematic for learners of Dutch from various L1 backgrounds [17]. The results showed that for the experimental group that had been using Dutch-CAPT for 4 weeks the reduction in the pronunciation errors addressed in the training system was significantly larger than in the control group [3]. These results are promising and show that it is possible to use speech technology in CALL applications to improve pronunciation.

We therefore decided to extend this approach to other aspects of speaking proficiency like morphology and syntax. So far there are no systems that are capable of automatically detecting morphology and syntax errors in speaking performance and provide feedback on them. A project proposal which aimed to achieve this was funded by the STEVIN programme: the DISCO project. At the moment of writing the DISCO project has not been completed yet. Therefore, in this chapter we report on the research that has been carried out so far.

In the remainder of this chapter we first describe the aim of the DISCO project. We then go on to briefly deal with materials and method with respect to system design and speech technology components. Subsequently, we present the results of the DISCO project that are currently available. We then discuss the DISCO results, we consider how DISCO has contributed to the state of the art and present some future perspectives.

18.2 DISCO: Aim of the Project

The aim of the DISCO project was to develop a prototype of an ASR-based CALL application for Dutch as a second language (DL2). The application aims at optimising learning through interaction in realistic communication situations and providing intelligent feedback on important aspects of L2 speaking, viz. pronunciation, morphology, and syntax. The application should be able to detect and give feedback on errors that are made by DL2 learners.

L2 learners tend to make different morphologic and syntactic errors when they speak than when they write. It is generally acknowledged in the L2 literature

[1] http://www.digitalpublishing.de

[2] http://www.tellmemore.com/

[3] http://www.rosettastone.com/

that the fact that L2 learners are aware of certain grammatical rules (i.e. those concerning subject-verb concord of number, tenses for strong and weak verbs, and plural formation) does not automatically entail that they also manage to marshal this knowledge on line while speaking. In other words, in order to learn to speak properly, L2 learners need to practice speaking and to receive corrective feedback on their performance on line, both on pronunciation and on morphology and syntax. The ASR-based CALL system to be developed in the DISCO project was conceived to make this possible.

With respect to pronunciation, we aimed at the achievement of intelligibility, rather than accent-free pronunciation. As a consequence, the system was intended to target primarily those aspects that appear to be most problematic. In previous research [17] we gathered relevant information in this respect. In the DISCO project we wanted to extend the pronunciation component by providing feedback on more sounds and by improving the pronunciation error detection algorithms.

It is well-known that recognition of non-native speech is problematic. In the Dutch-CAPT system recognition of the utterances was successful because we severely restricted the exercises and thus the possible answers by the learners. Since DISCO also addresses morphology and syntax, the exercises have to be designed in such a way that L2 learners have some freedom in formulating their answers in order to show whether they are able to produce correct forms. So, the challenge in developing an ASR-based system for practicing oral proficiency consists in designing exercises that allow some freedom to the learners in producing answers, but that are predictable enough to be handled automatically by the speech technology modules.

In morphology and syntax we wanted to address errors that are known to cause problems in communication and that are known to be made at the low proficiency level (the so called A1/A2 proficiency level of the Common European Framework) that is required in national language citizenship examinations in the Netherlands ('inburgeringsexamen'). For morphology this concerns (irregular) verb forms, noun plural formation; and for syntax it concerns word order, finite verb position, pronominal subject omission, and verb number and tense agreement.

The DISCO project is being carried out by a Dutch-Flemish team consisting of two academic partners, the Radboud University in Nijmegen (CLST and Radboud in'to Languages) and the University of Antwerp (Linguapolis), and the company Knowledge Concepts.

18.3 Material and Methods: Design

In this section we first describe the user interaction design and secondly the design of the speech technology modules utilised in the system.

18.3.1 User Interaction Design

The design model for the project was based on the engineering approach described in [2]. The design concepts for the application to be developed were derived from a thorough analysis of pedagogical and personal goals. While the pedagogical goals of this project were clearly formulated, for the elicitation of personal goals we needed to conduct a number of specific focus groups and in-depth interviews.

18.3.1.1 Interviews with DL2 Teachers and Experts

Exploratory in-depth interviews with DL2 teachers and experts were conducted. The results presented in this sub-section concern their opinions about DL2 learners.

Two types of DL2 learners were identified: those who want immediate corrective feedback on errors, and those who want to proceed with conversation training even if they make errors. Teachers also believed that our target group (highly-educated DL2 learners) would probably prefer immediate corrective feedback. To cater for both types of learners, the system could provide two types of feedback strategies and have the learners choose the one that suits them better through parameter setting.

The interviews also revealed that DL2 learners often want more opportunities to practice. A CALL system can provide these opportunities. DL2 learners feel uneasy at speaking Dutch because they are not completely familiar with the target language and culture. Therefore, it might be a good idea to provide some information about the target culture(s), so that learners can try to achieve intercultural competence.

18.3.1.2 Focus Group with DL2 Students

A focus group with nine DL2 learners revealed that DL2 learners preferred conversation simulation for building self-confidence over another traditional school-like approach. They also clearly preferred respect for their identity over explicit focus on integration.

DL2 learners often feel discouraged if they don’t have sufficient knowledge of the topic of the conversation, for example politics, habits, etc. Furthermore, they want to feel respected for their courage to integrate in the target culture(s). The conversations may thus certainly deal with habits and practices of the target culture(s).

Also, learners feel frustrated because they cannot keep up with the pace of conversations in the target language. DL2 teachers and experts mentioned lack of exposure to L2 culture, but the participants did not complain about this lack, even if we explicitly asked them.

18.3.1.3 Conceptualisation

After an initial design based on a concept where the user was expected to make choices (communicative situation, pronunciation/morphology/syntax), we eventually decided to limit our general design space to closed response conversation simulation courseware and interactive participatory drama, a genre in which learners play an active role in a pre-programmed scenario by interacting with computerised characters or "agents".

The simulation of real-world conversation is closed and receptive in nature: students read prompts from the screen. However, at every turn, students pick the prompt of their choice, which grants them some amount of conversational freedom. The use of drama is beneficial for various reasons, (a) it "reduces inhibition, increases spontaneity, and enhances motivation, self-esteem and empathy" [13], (b) it casts language in a social context and (c) its notion implies a form of planning, scenario-writing and fixed roles, which is consistent with the limitations we set for the role of speech technology in DISCO [21].

This framework allows us to create an engaging and communicative CALL application that stimulates Dutch L2 (DL2) learners to produce speech and experience the social context of DL2. On the other hand, these choices are safe from a development perspective, and are appropriate for successfully deploying ASR while taking into account its limitations [10]. In order to make optimal choices with respect to important features of the system design, a number of preparatory studies was carried out in order to gain more insight into important features of system design such as feedback strategies, pedagogical and personal goals.

18.3.1.4 Prototyping

Pilot Study with DL2 Teachers

The current and the following pilot study were carried out by means of partial systems with limited functionality (e.g. no speech technology). The functions of the system that were not implemented such as playing prompts and giving feedback were simulated. For this pilot study, an internet application was used to present one conversation tree including graphics.

In general, DL2 teachers were positive about the possibilities offered by such a CALL system to practice pronunciation, morphology and syntax. Most of the comments dealt with how the exercises on morphology and syntax should be designed. The main conclusions were that different types of exercises probably require different approaches.

Pronunciation Exercises For pronunciation exercises, we decided that simply reading aloud sentences is a good modality for reliably detecting and correcting errors in pronunciation.

Morphology Exercises Regarding morphology, a multiple choice approach was recommended. For example, for personal and possessive pronouns: “Hoe gaat het met (jij/jou/jouw)?” (“How are (you/you/your)?”) and for verb inflections: “Hoe (ga/gaat/gaan) het met jou?” (“How (are/is/to be) you?”).

Syntax Exercises For syntax exercises, constituents can be presented in separate blocks in a randomised order. There shouldn’t be too many of them (e.g. max. four) and some of these blocks could be fixed, such as the beginning and the end of the sentence. This can be made clear by using differently colored blocks.

Pilot Study with DL2 Students

A web-based prototype of the application was developed. A pronunciation teacher simulated the functions that were not yet implemented, e.g. by reading lines from the screen and providing feedback. The speech of the students was recorded, video recordings were made, and subsequently analyzed.

The pilot was carried out in Antwerp (five participants) and Nijmegen (four participants). The first research question concerned the feedback students prefer. Five out of nine respondents indicated a preference for immediate feedback, and four out of nine students responded that they did not know which feedback they preferred. The fact that no student wanted communicative (delayed) feedback confirms the hypothesis that highly-educated learners want to receive overt feedback with high frequency.

In exercises on morphology and syntax students first have to construct the grammatical form they want to utter. As a result, the cognitive load produced by these exercises is probably higher, which in turn may lead to a higher number of disfluencies and to speech recognition and error detection problems. A possible solution might be to ask students to first construct their answer on the screen by means of keyboard and mouse (textual interaction), and then utter these answers.

The average number of disfluencies per turn were measured by hand and we found that it was significantly lower in the cases with textual interactions. This shows that this procedure is useful to substantially reduce the number of disfluencies. However, CALL research does suggest that it is beneficial to maintain modalities, and not to use keyboard and mouse interaction in courseware that is essentially conversational in nature [13].

Furthermore, for some students it may not be necessary, or students may have a preference for not using it. Based on these results textual interaction could be included as an option and the output could be used to improve speech recognition and error detection.

Another important result from this pilot study is that the order of events was not always clear to students. Although the teacher that guided the experiment provided instructions that would normally be shown by the computer, students did things in the wrong order, acted ahead of time, spoke while carrying out the textual interaction, only uttered part of the prompts, or proceeded to the next item without speaking the utterance. The consequences for the design are that the interaction

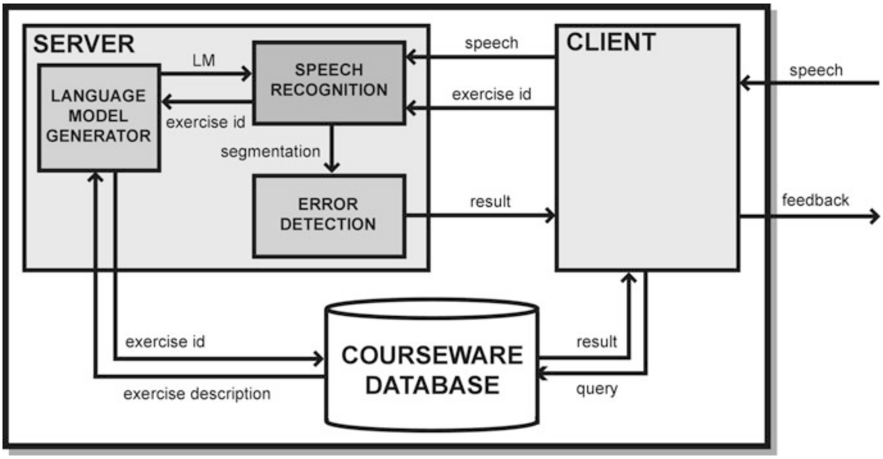


Fig. 18.1 Architecture of the DISCO system. More information is given in Sect. 18.3.2.1

sequence should be clearly structured and scaffolded, that instructions should be clear and concise, that a push-to-talk button should be used, and that students should be allowed to proceed to the next item if they have finished their task.

Finally, we also noticed that teachers, both in Nijmegen and in Antwerp, spontaneously provided non-verbal feedback during the conversation, and that students clearly responded to this kind of feedback. As CALL research also suggests [11], non-verbal feedback may be used complementarily to the verbal (overt or covert) feedback, and may be beneficial to student motivation and the learning effect. The virtual agents can provide this kind of feedback, e.g. by nodding or shaking their heads, smiling, frowning, etc.. However, we will need to be careful with showing this kind of feedback at all times, since it may become tiresome after a while. A random or intelligent random control for the non-verbal feedback may need to be implemented.

18.3.2 Speech Technology Design

18.3.2.1 System Architecture

Based on the exercises described in the previous section, we designed a system architecture which in principle is able to fulfill all the requirements stated during the courseware design phase (Fig. 18.1).

The system consists of three main components: the client, the server and the courseware database. The client will handle all the interaction with the user, such as recording the audio, showing the current exercise and appropriate feedback, as well as keeping track of the user's progress. The content of the courseware is stored in the courseware database. The server is the component which processes the spoken utterances and detects errors.

In the DISCO application, the students' utterances have to be handled by the speech technology. For this purpose we employ a two-step procedure which is performed by the server: first it is determined what was said (content), and second how it was said (form). On the basis of the current exercise, the server generates a language model (language model generator) which is used by the speech recognition module to determine the sequence of words uttered by the student. If the speech recognition manages to do this, possible errors in the utterance are then detected by the error detection module. Finally, a representation of the spoken utterance, together with detected errors, is sent back to the client. The client then provides feedback to the learner.

The details of the design of the speech recognition and error detection modules are presented below.

18.3.2.2 Speech Recognition

For developing the speech recognition module the DISCO project has been able to profit from a previous STEVIN project, the SPRAAK project Chap. 6, which provided the speech recognition engine employed in DISCO.

During speech recognition, which is necessary to establish whether the learner produced an appropriate answer, the system should tolerate deviations in the way utterances are spoken. We call this step utterance selection. Exercises are designed such as to elicit constrained responses from the learner. For each exercise there is a specific list of predicted, correct and incorrect, responses. Incorrect responses are automatically generated using language technology tools based on the correct target responses.

Syntax Exercises In syntax exercises, three or four groups of words are presented on the screen. The task of the student is to speak these word groups in a syntactically correct order. For these exercises, language models are automatically generated by including all permutations of the word groups as paths in a finite state grammar (FSG). The task of the speech recogniser is to determine which of these paths in the FSG is the most likely one given the speech input from the student.

Morphology Exercises In morphology exercises, a whole sentence is presented on the screen, but for one word a multiple choice list containing alternatives for that word, typically around two to four, is presented. Here, the language models are generated in a similar fashion as in the syntax exercises. For the word that has to be chosen by the student, alternative paths are included in the FSG.

Pronunciation Exercises In pronunciation exercises, language models contain only one path: the target utterance. The reason for doing this recognition is explained below.

The sequence of words that is now selected does not always correspond exactly to what was actually spoken: the spoken utterance might not be present in the FSG, or even if it is present it might not be the one that is actually recognised. Since providing feedback on the wrong utterance is confusing, we try to avoid this as much as possible. To this end we automatically verify whether the recognised utterance was spoken using a so called confidence measure, which indicates how well the recognised word sequence reflects the spoken utterance. The confidence measure is compared to a predefined threshold to determine whether the utterance has to be accepted (confidence measure above the threshold) or rejected (below the threshold). This step is called utterance verification. When the utterance is accepted the learner gets feedback on the utterance, if it is rejected the learner might be asked to try again.

We conducted several experiments for optimising both utterance selection and utterance verification steps within the speech recognition module. These are described in Sect. 18.4.2.1.

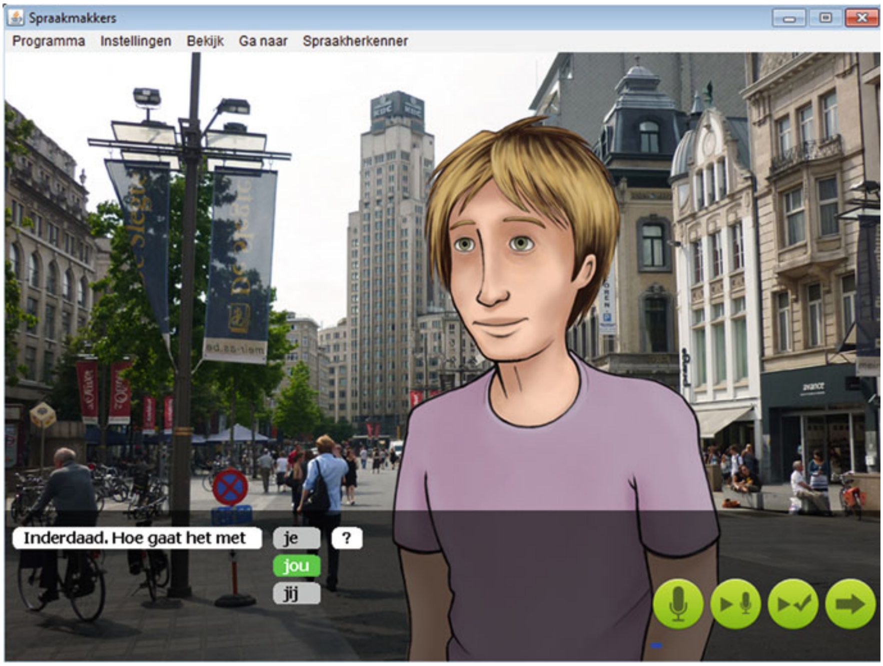
18.3.2.3 Error Detection

After the speech recognition module has calculated the segmentation of the speech signal into words and phones, the error detection module detects errors on the levels of pronunciation, morphology and syntax. These types of error detection are explained below.

Pronunciation Exercises In previous studies [17] we investigated which pronunciation errors are made by learners of Dutch, and how these errors can be detected automatically. On the basis of three different databases, we drew up an inventory of frequent errors made by DL2 students [17]. Since Dutch has a rich vowel system, it is not surprising that many of the errors concern vowels. The distinction between tense and lax vowels, and the diphthongs appear to be problematic. Among the consonants the velar fricative /x/, a well-known shibboleth sound, and the glottal fricative /h/ seem to pose problems. For this reason we focused on detecting errors in the following phonemes: /i/, /ɪ/, /eː/, /ɛ/, /aː/, /ɑ/, /oː/, /ɔ/, /u/, /y/, /ʏ/, /ɛi/, /ʌu/, /øː/, /œy/, /x/, /ɦ/ and /ŋ/.

For pronunciation error detection, it has to be tested whether segments are realised correctly. We carried out multiple experiments to evaluate existing automatic methods for detecting these kinds of errors.

Syntax and Morphology Exercises While pronunciation error detection concerns detecting whether segments are correctly realised or not, syntactic and morphological error detection generally concerns detecting which words are correctly realised and whether they are in the right order. Because syntactically and morphologically incorrect responses are included in the list of predicted (correct and incorrect) responses, the output of the speech recognition module can thus be an incorrect utterance present in the predicted list and in this way errors can be detected.



Inderdaad. Hoe gaat het met

je

jou

jij

?

Fig. 18.2 Screenshot of a morphology exercise in the DISCO system. The student gave the correct answer which is indicated by the *green block*. The functions of the four buttons on the right of the screen are (from left to right): start and stop recording speech input, listen to your own answer, listen to the prerecorded correct answer and proceed to the next prompt

18.4 Results

18.4.1 Design of the DISCO System

The results of the preparatory studies were taken into account in finalising the design of the DISCO system. The practice session starts with a relatively free conversation simulation, taking well into account what is (not) possible with speech technology: learners are given the opportunity to choose from a number of prompts at every turn (branching, decision tree, as shown in Fig. 18.2). Based on the errors they make in this conversation they are offered remedial exercises, which are very specific exercises with little freedom.

Feedback depends on individual learning preferences: the default feedback strategy is immediate corrective feedback, which is visually implemented through highlighting, and from an interaction perspective by putting the conversation on hold and focusing on the errors. Learners that wish to have more conversational freedom can choose to receive communicative recasts as feedback, which let the conversation go on while highlighting errors for a short period of time.

18.4.2 Speech Technology

18.4.2.1 Speech Recognition

For the purpose of developing the speech recognition module we used the JASMIN-CGN corpus (cf. Chap. 3, p. 43 to train and test experimental implementations. In a study in which we tested an experimental implementation of the speech recognition module, we showed that significant improvements relative to a baseline recognition system can be attained in several ways. The details of this experiment are described in [9].

The baseline system was a standard HMM-based speech recogniser with acoustic models trained on native speech. The language models were FSGs with about 30 to 40 parallel paths containing answers from non-native speakers to questions (from the JASMIN-CGN speech corpus). This baseline system had an utterance error rate UER of 28.9 %. The UER could be decreased to 22.4 % by retraining the acoustic phone models with non-native speech.

Furthermore, we found that filled pauses, which are very frequent in non-native speech [4], can be handled properly by including 'filled pause'-loops in the language model. Filled pauses are common in everyday spontaneous speech and generally do not hamper communication. Students are therefore allowed to produce (a limited number of) filled pauses. By using phone models trained on non-native speech and language models with filled pause loops, the UER of the speech recognition module in this task was reduced to 9.4 %.

As explained in Sect. 18.3.2.3, after the selection of the best matching utterance, the utterance verification step is needed to verify whether the selected response was indeed the utterance that was actually spoken by the learner. In [9] we presented and evaluated different methods for calculating confidence measures that are employed for this verification step.

The best results were obtained through a combination of acoustic likelihood ratios and phone duration features using a logistic regression model. The acoustic likelihood ratio indicates how well the acoustic features calculated from the speech match with the recognised utterance. Using only this feature the system has an equal error rate (EER) of 14.4 %. The phone duration features measure the number of extremely short (lower than the 5th percentile duration measured in a native speech database) and long (higher than the 95th percential duration) phones. By adding these features to the regression model the EER is decreased to 10 %.

18.4.2.2 Error Detection

In the current system design syntactical and morphological errors can already be detected after speech recognition, so no additional analysis is needed for these kinds of errors. However, for pronunciation errors such an analysis is required because these errors often concern substitutions of acoustically similar sounds. Therefore, considerable research efforts were made to improve the detection of pronunciation errors.

First, we conducted an experiment with artificial pronunciation errors in native speech [5]. We introduced substitutions of tense with lax vowels and vice versa, which is an error pattern frequently found in non-native speech. The results of this experiment show that discriminative training using Support Vector Machines (SVM's) based on acoustic features results in better pronunciation error classifiers than traditional acoustic likelihood ratios (LLR) (EER's of 13.9 % for SVM classifiers versus 18.9 % for LLR-based scores).

After having invested in improving the annotation of non-native read and spontaneous speech material in the JASMIN-CGN speech corpus, we first studied whether and how the error patterns of these two types of speech material differ in terms of phoneme errors [6]. We concluded that these two types of material indeed contained different phonemic error patterns, which partly depend on the influence of Dutch orthography [7].

Furthermore, we observed specific vocalic errors related to properties of the Dutch vowel system and orthography. We used this knowledge to develop a new type of pronunciation error classifier, which is designed to automatically capture specific error patterns using logistic regression models [7] and [8]. These classifiers performed better than acoustic LLR-based scores with average EERs of 28.8 % for the LLR-based scores and 22.1 % for the regression models).

18.4.3 System Implementation

We implemented the system architecture as depicted in Fig. 18.1. As stated in Sect. 18.3.2.1. the system has three main components: the client, the courseware database and the speech processing server. One of the advantages of separating client and server is that these components can be developed relatively independently, as long as the communication protocol is clearly defined. In most cases this might be the optimal set-up because different components will typically be developed by different experts, for example interaction designers, language teachers and speech technologists. The protocol was devised before developing the client and the server and it caters for both the transmission of audio and status messages (speech recogniser ready to receive speech, recognition started, recognition finished etc.). We chose to use one central server that can handle multiple clients because this is easy to maintain and update.

The client is implemented in Java using the AWT toolkit. The user-system interactions, the learners results, and the courseware, are stored in the relational MySQL courseware database. The speech processing server, which is the component which processes the spoken utterances and detects possible errors, is implemented in Python. The SPRAAK speech recogniser, implemented in C with an API in Python, is used in the speech recognition module. To handle multiple recognition requests a queueing system was implemented in which a constant number of recognisers is initialised. If all the recognisers in the queue recognise when a new recognition request from a client comes in, this request is processed only after one of the recognisers has finished. This queueing method makes the system easily scalable.

Due to practical constraints, the speech recogniser's phone models are trained on native speech, the utterance verification is performed by only using an acoustic LLR measure and for pronunciation error detection we have also used acoustic LLR measures.

18.4.4 Evaluation

As mentioned above, various components of the system were evaluated at different stages in the project: the exercises, the speech recognition module, the error detection module, and finally the whole system as a preparation of the final evaluation. For the final evaluation of the whole system we chose an experimental design in which different groups of DL2 students at UA and Radboud into Languages use the system and fill in a questionnaire with which we can measure the students' satisfaction in working with the system. The student-system interactions are recorded. Experts then assess these recordings (the system prompts, student responses, system feedback, etc.) to study the interaction and especially the quality of the feedback on the level of pronunciation, morphology and syntax. At the moment of writing this evaluation is being conducted.

Given the evaluation design sketched above, we consider the project successful from a scientific point of view if the DL2 teachers agree that the system behaves in a way that makes it useful for the students, and if the students rate the system positively on its most important aspects.

18.5 Related Work and Contribution to the State of the Art

Within the framework of the DISCO project various resources have been developed. First of all a blue-print of the design and the speech technology modules for recognition (i.e. for selecting an utterance from the predicted list, and verifying the selected utterance) and for error detection (errors in pronunciation, morphology, and syntax). In addition: an inventory of errors at all these three levels, a prototype of the DISCO system with content, specifications for exercises and feedback strategies, and a list of predicted correct and incorrect utterances.

The fact that DISCO is being carried out within the STEVIN programme implies that its results, all the resources mentioned above, will become available for research and development through the Dutch Flemish Human Language Technology (HLT) Agency (TST-Centrale[4]).This makes it possible to reuse these resources

[4] www.tst-centrale.org

for conducting research and for developing specific applications for ASR-based language learning.

In addition, within DISCO research was conducted to optimise different aspects of the system. For instance, [9] presented research aimed at optimising automatic speech recognition for low-proficient non-native speakers, which is an essential element in DISCO. [5] addressed the automatic detection of pronunciation errors, while in [7] and [8] we described research on alternative automatic measures of pronunciation quality.

In [6] we studied possible differences in pronunciation error incidence in read and spontaneous non-native speech. Finally, research on automatic detection of syntactical errors in non-native utterances was reported on in [19] and [20].

Apart from the resources that become available during development of the system, additional resources can be generated by using the CALL system after it has been developed. Language learners can use it to practice oral skills and since the system has been designed and developed so as to log user-system interactions, these can be employed for research. The logbook can contain various information: what appeared on the screen, how the user responded, how long the user waited, what was done (speak an utterance, move the mouse and click on an item, use the keyboard, etc.), the feedback provided by the system, how the user reacted on this feedback (listen to example (or not), try again, ask for additional, e.g. meta-linguistic, feedback, etc.).

Finally, all the utterances spoken by the users can be recorded in such a way that it is possible to know exactly in which context the utterance was spoken, i.e. it can be related to all the information in the logbook mentioned above. An ASR-based CALL system like DISCO, can thus be used for acquiring additional non-native speech data, for extending already existing corpora like JASMIN-CGN, or for creating new ones. This could be done within the framework of already ongoing research without necessarily having to start corpus collection projects.

Such a corpus and the log-files can be useful for various purposes: for research on language acquisition and second language learning, studying the effect of various types of feedback, research on various aspects of man-machine interaction, and of course for developing new, improved CALL systems. Such a CALL system will also make it possible to create research conditions that were hitherto impossible, thus opening up possibilities for new lines of research.

For instance, at the moment a project is being carried out at the Radboud University of Nijmegen, which is aimed at studying the impact of corrective feedback on the acquisition of syntax in oral proficiency.[5] Within this project the availability of an ASR-based CALL system makes it possible to study how corrective feedback on oral skills is processed on-line, whether it leads to uptake in the short term and to actual acquisition in the long term. This has several advantages compared to other studies that were necessarily limited to investigating interaction in the written modality: the learner's oral production can be assessed on line,

[5] http://lands.let.kun.nl/~strik/research/FASOP.html

corrective feedback can be provided immediately under near-optimal conditions, all interactions between learner and system can be logged so that data on input, output and feedback are readily available for research.

18.6 Discussion and Conclusions

In the previous sections we have presented the various components of the DISCO system, how they have been developed, the results that have been obtained so far, and the resources that have been produced. The methodological design of the system has led to a software architecture that is sustainable and scalable, a straightforward interface that appeals to – and is accepted by – the users (by responding to their subconscious personal goals), a sophisticated linguistic-didactic functionality in terms of interaction sequences, feedback and monitoring, and an open database for further development of conversation trees. However, for a more complete and detailed appreciation of the whole system we will have to await the results of the final evaluation which is now being conducted.

In this paper we have also seen how important language resources are for developing CALL applications and how fortunate it was for DISCO to be able to use the JASMIN-CGN speech corpus (cf Chap. 3, p. 43) and the SPRAAK toolkit (cf Chap. 6, p. 95). In addition, we have underlined the potential of such applications for producing new valuable language resources which can in turn be used to develop new, improved CALL systems.

References

1. Bloom, B.: The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educ. Res. **13**(6), 4–16 (1984)
2. Colpaert, J.: Elicitation of language learners' personal goals as design concepts. Innov. Lang. Learn. Teach. **4**(3), 259–274 (2010)
3. Cucchiarini, C., Neri, A., Strik, H.: Oral proficiency training in dutch L2: the contribution of ASR-based corrective feedback. Speech Commun. **51**(10), 853–863 (2009)
4. Cucchiarini, C., van Doremalen, J., Strik, H.: Fluency in non-native read and spontaneous speech. In: Proceedings of DiSS-LPSS Joint Workshop 2010, Tokyo (2010)
5. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic detection of vowel pronunciation errors using multiple information sources. In: Proceedings of ASRU 2009, Merano, pp. 580–585 (2009)
6. van Doremalen, J., Cucchiarini, C., Strik, H.: Phoneme errors in read and spontaneous non-native speech: relevance for CAPT system development. In: Proceedings of the SLaTE-2010 workshop, Tokyo (2010a)

7. van Doremalen, J., Cucchiarini, C., Strik, H.: Using non-native error patterns to improve pronunciation verification. In: Proceedings of Interspeech, Tokyo, pp.590–593 (2010b)
8. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic pronunciation error detection in non-native speech. submitted to J. Acoust. Soc. Am.
9. van Doremalen, J., Cucchiarini, C., Strik, H.: Optimizing automatic speech recognition for low-proficient non-native speakers. EURASIP J. Audio Speech Music Process. (2010d), http://asmp.eurasipjournals.com/content/2010/1/973954
10. van Doremalen, J., Cucchiarini, C., Strik, H.: Automatic speech recognition in CALL systems: the essential role of adaptation. Commun. Comput. Inf. Science, **126**, 56–69 (2011). Springer
11. Engwall, O., Bälter, O.: Pronunciation feedback from real and virtual language teachers. J. Comput. Assist. Lang. Learn. **20**(3), 235–262
12. Eskenazi, M.: Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype. Lang. Learn. Technol. **2**, 62–76 (1999)
13. Hubbard, P.: Interactive participatory dramas for language learning. Simul. Gaming **33**, 210–216 (2002)
14. Kim, Y., Franco, H., Neumeyer, L.: Automatic pronunciation scoring of specific phone segments for language instruction. In: Proceedings of Eurospeech, Rhodes, pp. 645–648 (1997)
15. Mak, B., Siu, M., Ng, M., Tam, Y.-C., Chan, Y.-C., Chan, K.-W.: PLASER: pronunciation learning via automatic speech recognition., In: Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications using Natural Language Processing, Edmonton, pp. 23–29 (2003)
16. Menzel, W., Herron, D., Bonaventura, P., Morton, R.: Automatic detection and correction of non-native English pronunciations. In: Proceedings of InSTILL, Dundee, pp. 49–56 (2000)
17. Neri, A., Cucchiarini, C., Strik, H.: Selecting segmental errors in L2 Dutch for optimal pronunciation training. Int. Rev. Appl. Linguist. **44**, 357–404 (2006)
18. Precoda, K., Halverson, C.A., Franco, H.: Effects of speech recognition-based pronunciation feedback on second-language pronunciation ability. In: Proceedings of InSTILL, Dundee, pp. 102–105 (2000)
19. Strik, H., van de Loo, J., van Doremalen, J., Cucchiarini, C.: Practicing Syntax in spoken interaction: automatic detection of syntactic errors in non-native utterances. In: Proceedings of the SLaTE-2010 workshop, Tokyo (2010)
20. Strik, H., van Doremalen, J., van de Loo, J., Cucchiarini, C.: Improving ASR processing of ungrammatical utterances through grammatical error modeling. In: Proceedings of the SLaTE-2010 workshop, Venice (2011)
21. Strik, H., Cornillie, F., Colpaert, J., van Doremalen, J., Cucchiarini, C. (2009) Developing a CALL system for practicing oral proficiency: how to design for speech technology, pedagogy and learners. In: Proceedings of the SLaTE-2009 workshop, Warwickshire (2011)
22. Witt, S.M.: Use of speech recognition in computer-assisted language learning. Doctoral Dissertation, Department of Engineering, University of Cambridge, Cambridge (1999)