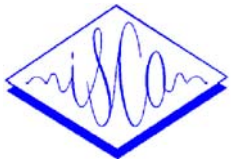

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

SOFTWARE TO SUPPORT RESEARCH AND DEVELOPMENT OF SPOKEN DIALOGUE SYSTEMS

Michael F. McTear
School of Information and Software Engineering
University of Ulster
Newtownabbey BT37 0QB
Northern Ireland
mf.mctear@ulst.ac.uk

ABSTRACT

The development of a spoken dialogue system requires the integration of the various components of spoken language technology, such as speech recognition, natural language processing, dialogue modelling, and speech synthesis. Recently several toolkits have been developed that provide support for this process, enabling developers who have no specialist knowledge of the component technologies to produce working spoken dialogue systems with relative ease. This paper reports on the use of CSLU's RAD (Rapid Application Developer) to provide practical experience for undergraduate students taking courses in spoken dialogue systems. Two groups of students were involved - students of linguistics, speech and language therapy, and communication, on the one hand, and students of computational linguistics and computing science. The paper describes the use of the toolkit for students with these different degrees of competence in computing and reports on plans for future work with the toolkit.

1. INTRODUCTION

Developing a spoken dialogue system is a complex process involving the integration of the various components of spoken language technology. It would be a formidable task to build and integrate these components from scratch. Fortunately a number of toolkits and authoring environments have become available that support the construction of spoken dialogue systems, even for those who have no specialist knowledge of the component technologies such as speech recognition and natural language processing. The present paper concentrates on courseware and toolkits that have been developed for educational purposes and that are available in the public domain, focussing in particular on CSLU's RAD (Rapid Application Developer).

2. COURSEWARE FOR SPOKEN DIALOGUE SYSTEMS DEVELOPMENT

The Danish Dialogue Project, which began in 1991, has been a focal point for extensive research in interactive speech systems, with a particular emphasis on dialogue development methodologies and platforms [1]. Recent EU-funded projects include DISC, a project that examines best practice in dialogue development and evaluation, and REWARD, a project concerned with development platforms for real world applications of spoken dialogue systems. One of the partners in these projects, the Centre for PersonKommunikation at the University of Aalborg, has produced a Web-based course that has the goal of giving 'an understanding of the design and implementation of speech-based interfaces through hands-on experience in building a spoken dialogue system' [2]. Students taking the course are required to design, implement and test a spoken dialogue system for a telephone based service. The dialogues, which are developed in a textual script-language, are tested on the REWARD dialogue platform, which is connected to the telephone network. The course assumes that the students have taken prerequisite modules in 'Spoken Language Processing' and 'Design of Multi Modal HCI Systems'. The literature supporting the course includes reports from the DISC and REWARD projects. This material, which includes details of the development platform to be used for implementation, is not publicly available.

GULAN: A System for Teaching Spoken Dialogue Systems Technology, is under development at KTH (Stockholm) and at Linköping University and Uppsala University [3]. The system, which is currently in Swedish but due to be ported to English, is presently only runnable locally.

The CSLU toolkit RAD is available free-of-charge under a license agreement for educational, research, personal, or evaluation purposes. The toolkit has been developed at the Center for Spoken Language Understanding (CSLU) at the Oregon Graduate Institute of Science and Technology to support speech-related research and development activities. The toolkit includes core technologies for speech recognition and text-to-speech synthesis, as well as a graphically-based authoring environment (RAD) for designing and implementing spoken dialogue systems. Essentially this process involves selecting and linking graphical dialogue objects into a finite-state dialogue model. Each object can be used for functions such as generating prompts, recording and recognizing speech, as well as performing actions that are programmed in TCL. Speech recognition is automated for the developer who simply lists the words to be recognised at each state. Prompts, which are output using the Festival TTS (text-to-speech) system, are specified in textual form, or they can be pre-recorded, and, with some additional effort, spliced together at run-time. Other facilities include sub-dialogues for dialogue sub-tasks, a default repair dialogue if the recognition score for the user's input falls below a given threshold, special recognisers for digits and spelt words, and functions to support voice-based Web access. This paper describes courses taught using a beta version of the toolkit. Since then the toolkit has been thoroughly overhauled and extended to include a wide range of new features [4,5].

3. RAD IN COURSES AT THE UNIVERSITY OF ULSTER

RAD has been used to support the teaching of spoken dialogue technology to two different groups of undergraduate students at the University of Ulster. The first group consisted of students of Linguistics, Speech and Language Therapy, and Communication Studies, who had taken several courses in linguistics, including pragmatics and discourse analysis. These students had a good understanding of the nature of human language and of human-human dialogue, but relatively little computer literacy and no programming experience. Some of the students had a sound command of phonetics, particularly those studying the degree in Speech and Language Therapy. None of the students in this group had any understanding of the issues involved in speech technology. The second group consisted of students with a computing background – some of these were taking a degree in Computational Linguistics, while others were taking a degree in Computing Science. The former had a good understanding of issues in linguistics and computational linguistics and had some programming experience, although not in TCL, the language required by the RAD toolkit. The Computing Science students had specialist knowledge of topics such as software engineering, database technology, systems analysis and design, but had no background in linguistics. None of the students in this second group had any previous experience of speech technology. The courses given to both sets of students consisted of lectures, in which the theoretical background was presented, and laboratory classes, in which relevant techniques involving the toolkit were taught and practised. The students were then required to specify and develop a system of their choice. Assessment included both the program developed using the toolkit as well as a paper integrating the theoretical contents of the lectures with what was achieved and achievable using the toolkit. The following subsections outline the main differences in the orientation of the courses as determined mainly by the skills and prior knowledge of the students.

3.1 Using RAD with non-computing students

Following a brief introduction to spoken language technology, including an overview of the main issues involved in speech recognition, students were introduced to the basics of the software development lifecycle for spoken dialogue systems, including requirements analysis, specification, design, implementation, and testing. At the same time the students were taught the essential elements of RAD, including the construction of simple state-based dialogues, the use of subdialogues, how to record and splice together prompts, and how to use grammar-based recognisers for word spotting. In this way the students were made familiar with the types of dialogue that could be implemented using RAD, so as to inform their subsequent choice of a domain for the system they were to develop. The systems selected by the students involved transactions such as ordering meals from a student meal service, booking in animals for a cattle market, and acting as an auto-attendant in a furniture store. These applications had in common a well-structured task that could be decomposed into orderly sub-tasks. The students carried out a detailed task

analysis, on the basis of which they were able to specify the dialogue flow using flowcharts, which in turn could be easily implemented in RAD using a finite-state dialogue. Figure 1 presents a screenshot from the furniture store auto-attendant.

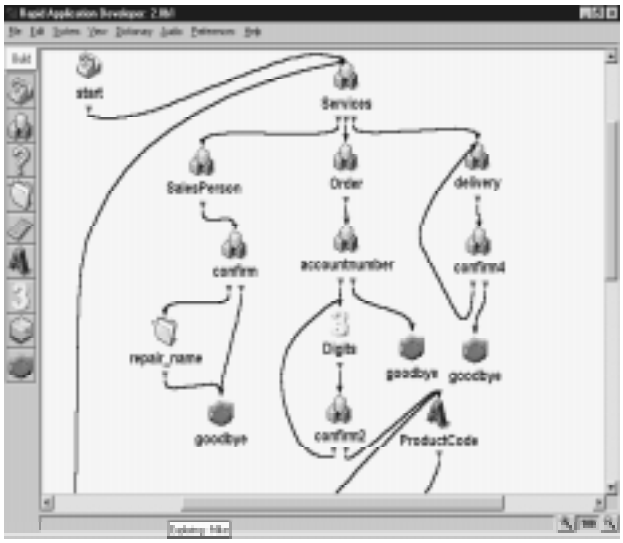

Figure 1. Using RAD to simulate an auto attendant at a furniture store

In this example there are three services – talk to a salesperson, place an order, and arrange delivery. The system is implemented using confirmation states, digit and alpha-digit recognisers for account numbers and product names, and a repair sub-dialogue for spelling misrecognised names.

Designing dialogues such as these enabled the students to appreciate the need for a clear analysis and specification of the system and to understand the advantages as well as the limitations of a state-based dialogue control. For example: providing confirmations and permitting corrections of misrecognised items required careful design to take account of the limitations of the dialogue control while at the same time ensuring that information had been accurately elicited and allowing for an acceptable dialogue flow. Additional tasks involved a limited amount of coding, for example, to return variables holding the words recognised at a particular dialogue state, to provide spliced prompts, or to specify grammar-based recognition. Students from the Speech and Language Therapy degree also used the facility for hand-crafted pronunciations to develop an interface that would recognise the speech of a user with a speech disability.

3.2 Using RAD with computing students

One of the main limitations of the applications developed by the non-computing students was that, although their systems elicited a wide range of information through the dialogues, there was no facility to communicate with an external information source such as a database, spreadsheet, or Web page. The second group of students was in a position to develop applications with such a facility.

The following example of a banking application, presented in Figure 2, is superficially similar to the systems developed by the non-computing students in that it consists of a state-based dialogue with sub-dialogues for the three sub-tasks, and within these sub-dialogues there are states for confirmations, and digit and alpha-digit recognisers. However behind these states there is more complex functionality. For example: the state 'TimeOfDay_Greeting' uses a TCL function to establish the time from the system clock, then determines which greeting to output based on the time of day. The system provides the

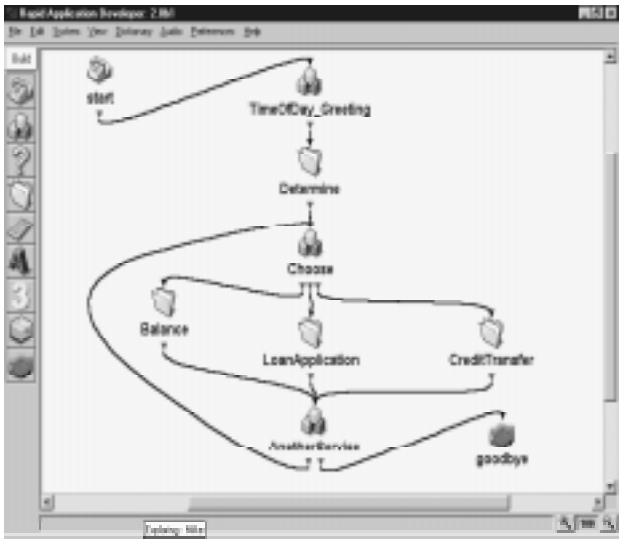

Figure 2. Using RAD for a banking application

user with two methods of system entry. If the user knows their Account number and Pin number, the system asks for these and verifies with the database before granting entry. Otherwise the system queries the user's name, address, phone number and date of birth, and then verifies these against the database before going on to offer a Pin reissue. Similar functionalities are provided for the main sub-tasks. For 'Balance Inquiry' the user's current balance is checked against the database; for 'Loan Application' the user's credit rating is determined by calculating from the current balance; and for 'Credit Transfer' there is a check for sufficient balance. In

addition, a log file is created doucmenting the transactions performed during the dialogue. In each case, these functions involved programming in TCL. Furthermore, as file access to external information sources such as databases and spreadsheets was not supported in this version of the toolkit, these functions had to be simulated using a suitably structured text file that could be accessed using various functions specified in TCL to read in the file and find relevant records (in the form of separate lines in the file) and fields (in the form of TCL lists) that retrieved the requested information by the query as elicited in the user-system dialogue.

4. FUTURE DEVELOPMENTS

There are several ways in which the ideas presented could be further developed. For example: in the current version of the toolkit there is a default repair sub-dialogue that is activated if the recognition score for the user's input falls below a given threshold. There is scope for a range of additional repair strategies, some of which can be used to handle generic repair situations such as low confidence recognition, while others apply to particular dialogue functions, such as eliciting a name or account number. There is also a need for specialised sub-dialogues for functions such as eliciting a series of digits, as in an account number, or eliciting expressions of time and dates. Problems can occur in the elicitiation of strings of digits. For example: the system may recognise more or fewer digits than were actually spoken by the user. There is a need for clearly specified sub-dialogues that can successfully elicit the correct string of digits. Similarly, sub-dialogues for time and date expressions, making use of grammar-based recognition specified with finite-state grammars, would provide a set of reusable components for a wide range of applications. In each case the subdialogues would require robust error handling functions.

A clear requirement for dialogue systems is the ability to communicate with external information sources. An interface to the MS Access database system has been developed in a recent student project at the University of Ulster, with support from CSLU. The robustness and usefulness of this interface will be evaluated in future projects.

Finally, there are plans to investigate the use of recent additions to the toolkit such as facial animation, synchronised lip movement, media objects, natural language understanding, and enhanced text to speech synthesis. These tools will be used for applications for language learning and for users with speech and language disabilities.

5. CONCLUDING REMARKS

Although no formal evaluation has been carried out as yet on the use of the toolkit to support the practical sessions as described here, it is useful to report on some informal observations. Both groups of students experienced teething problems with the system hanging up, returning incomprehensible error messages, or generally performing at a sub-optimal level. Some of these problems were due to the inexperience of the students who made predictable errors that led to system failure. In other cases the hardware was simply not powerful and fast enough for the software. Despite these problems, the students reported a great sense of achievement and excitement at being able to specify and implement a working spoken dialogue system. These encouraging results provide a good foundation for the extension of the course to students in other courses and, in the longer term, to schoolchildren and adolescents.

REFERENCES

[1] Bernsen, N.O., H. Dybkjaer and L. Dybkjaer (1998), *Designing Interactive Speech Systems: From First Ideas to User Testing*. New York: Springer Verlag.

[2] IMM course in Spoken Dialogue Systems. http://www.kom.auc.dk/~lbl/IMM/S9_98/SDS_course_overview.html

[3] Sjölander, K., J. Beskow, J. Gustafson, E. Lewin, R. Carlson and B. Granström (1998), Web-based educational tools for speech technology. *Proc 5th International Conference on Spoken Language Processing*, Dec 1998, Sydney, Australia, 3217-3220.

[4] Sutton, S. et al. (1998), Universal Speech Tools: The CSLU Toolkit, *Proc 5th International Conference on Spoken Language Processing*, Dec 1998, Sydney, Australia, 3221-3224.

[5] Cole, R. et al. (1999), New tools for interactive speech and language training: Using animated conversational agents in the classrooms of profoundly deaf children. *Proceedings of ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education*, London, UK, Apr 1999, 45-52.