Associating the Dialogue Move Engine with Speech Output

Jim Hieronymus Stina Ericsson Staffan Larsson

Distribution: Public

Specification, Interaction and Reconfiguration in
Dialogue Understanding Systems: IST-1999-10516

Deliverable D2.2

December 2000

Specification, Interaction and Reconfiguration in Dialogue Understanding Systems:
IST-1999-10516

Göteborg University
Department of Linguistics

SRI Cambridge
Natural Language Processing Group

Telefónica Investigación y Desarrollo SA Unipersonal
Speech Technology Division

Universität des Saarlandes
Department of Computational Linguistics

Universidad de Sevilla
Julietta Research Group in Natural Language Processing

For copies of reports, updates on project activities and other SIRIDUS-related information, contact:

The SIRIDUS Project Administrator
SRI International
23 Millers Yard,
Mill Lane,
Cambridge, United Kingdom
CB2 1RQ
milward@cam.sri.com

See also our internet homepage http://www.cam.sri.com/siridus

Contents

Chapter 1

Introduction

The purpose of this deliverable is to describe the connection of system speech output to information states in the baseline SIRIDUS architecture. This important step will allow us to explore ways in which information states can help guide and refine the speech synthesis, particularly in the area of prosody.

We first describe some of the potential benefits of using the information state to control speech output. We then compare and evaluate some available synthesizers for use in dialogue systems, and describe a baseline implementation of speech output for the SIRIDUS architecture. Finally, we discuss some advanced features of speech synthesizers which are of interest in the context of spoken dialogue systems.

Chapter 2

Using Information States to Control Speech Synthesis

2.1 Intonation and focus

Information states contain logical forms of the utterance which reveal the semantics of messages. Using the logical form and the dialogue move it should be possible to determine the focus words and the intonation contour which represents this meaning best. Many speech synthesis systems have the capability to alter the prosody by putting special commands in the text string or by using one of the markup languages for speech synthesis like Sable or VXTML. Many misunderstandings between dialogue systems and their users can be attributed to improperly placed focus or incorrect prosody. The default setting for speech synthesis is to be intelligible and to have neutral intonation. It is not reasonable to expect that this prosody will be correct for a turn in a dialogue.

2.2 Feedback and barge-in

Having a system which reports to the Dialogue Move Engine (DME) what it has just uttered, will allow us to try belief revision asynchronously, while the system is still speaking. Humans utter confirmation words (e.g. OK, Yes, Uh-huh) called backchannels while the other person is talking to confirm that they heard and understand what the person said. This builds a shared belief structure in which both parties know what has been agreed on in the present conversation. When the user “barges in” while the system is speaking to provide the information requested or to ask a clarification question, it is also important to know what the system has just said, in order to know what question or information the user heard before responding.

2.3 Multilingual output

Doing generation from logical forms has the promise of being able to produce generators from logical forms to speech for other languages than English. This is of great interest in the European Union, since there are a large number of official and unofficial languages.

Chapter 3

Choosing the speech synthesizer

There are a number of speech synthesizers which could be used for the Siridus system, each with particular advantages and disadvantages. The synthesizers we considered seriously are AT&T, Bell Labs, Eloquent (IBM), Festival, Telia Infovox, Lernout and Hauspie Real Speak, and Microsoft.

So far the possible systems offer a matrix of capabilities and shortcomings.

Synthesizer	Linux	Solaris	Windows/NT
AT&T	X	X	X
Bell Labs	X	X	X
Eloquent(IBM)	X	-	X
Festival	X	X	X
Infovox	-	-	X
L & H	-	-	X
Microsoft	-	-	X

Table 3.1: Platform availability

Synthesizer	Spanish	English	German	Swedish
AT&T		X		
Bell Labs	X	X	X	
Eloquent(IBM)	X	X	X	
Festival	X	X	X	
Infovox	X	X	X	X
L & H		X	X	X
Microsoft		X		

Table 3.2: Languages covered

The AT&T synthesis system comes from many years of research at Bell Labs [7] and the large unit synthesis work at ATR in Japan by Hunt and Black [4]. It has undergone many years of development on the text to phoneme system, and rarely mispronounces a word. It is a large unit concatenative synthesis system. It uses a large database to find the largest pieces of speech which fits with what is to be synthesized. This results in vary natural sounding speech, with strange prosodic shifts due to the concatenation. Words which cannot be found in the database are synthesized using diphones. So far the only language available is English. Other languages have been planned but as yet none are available.

The Bell Labs Text to Speech system has been developed over the past 30 years [6]. [7] Many advances in part of speech tagging, intonation modeling, pronunciation modeling and designing, cutting and refining diphones and larger units have gone into making the system highly intelligible. Mispronounced words are rare in this system and it is very good at pronouncing abbreviations correctly. The speech it produces still sounds buzzy, and somewhat unnatural. The system produces speech in English, German, French, Italian, Japanese, Spanish, Mandarin Chinese, and Romanian. This multilanguage capability is very interesting for European projects, since it includes many European languages. The Bell Labs system is available commercially for Windows for a modest fee.

Eloquent has been developed over the past 30 years [3], and is presently being distributed by IBM in its Via Voice system. The system has rather good pronunciation rules, so the words are generally correctly pronounced. The speech is less natural than the Bell Labs system, and the intonation is less natural. Eloquent was recently bought by SpeechWorks for multilingual speech output. Eloquent has text to speech in several languages including German, French, Spanish, and Italian. The IBM version runs on Windows NT and Linux, while SpeechWorks runs on Windows platforms.

The Festival Speech Synthesis system was developed at the Centre for Speech Technology Research at Edinburgh University over the past 7 years. It is the result of several years of research by Alan Black and Paul Taylor[8]. They were also responsible for developing the ATR CHATR Speech Synthesis system for Japanese and English. [1] Festival is a different design than CHATR, and is made available open source, so that researchers around the world can participate in its development and enhancement. Presently American and British English, Spanish, French and German is available. Newer developments in Festival include large unit synthesis and synthesis customized to a particular task domain. Festival allows the modification of the prosody for sentence focus, duration and pitch contour, which will allow the DME to generate more accurate prosody from the logical forms. Festival runs on Linux, Solaris, and Windows and is open source.

The Infovox Speech Synthesis System the result of many years of work at the Royal Institute of Technology in Stockholm, Sweden. The system has changed over the years from a Dennis Klatt like formant synthesizer [2], to a diphone synthesizer, and now to a large unit synthesizer. It can generate speech in Swedish, German, French, Russian, and Spanish. The best language by far is

Swedish, but their large unit system is not working yet, so the speech sounds buzzy. Hopefully they will soon produce a large unit system which will sound as natural as the AT&T system. The system runs on Windows.

Lernout and Hauspie has produced a speech synthesis system called Real Speak which is based on large unit synthesis [5]. The system is available for a few languages, like English, Dutch, Swedish, Norwegian, Spanish. The system for generating pronunciations is not as good at the one at Bell Labs, so in some of the online demonstrations, like Ananova, which read news, the mispronounciations are very noticeable. The Lernout and Hauspie system runs on Windows. Recently L&H has declared Bankruptcy, which clouds the future use of their systems.

Microsoft has developed a large unit synthesis system, called Whistler, which is eventually supposed to allow you to construct a speech synthesis system with your own voice automatically. Speech recognition is used to cut di-triphone units automatically from a short corpus. The make-your-own-voice component provides the units out of which the speech is synthesized, by automatically labeling your speech and automatically cutting diphones. The system uses the windows operating system. This system will not run on Linux.

Chapter 4

Baseline speech output module

We have chosen the Festival system for the baseline system because it provides source code and runs on a good range of platforms. Since Festival has a Sable interface for marking focus words and changing the intonation, we should be able to generate speech with the focus and intonation we need in the Siridus dialogue systems. We have also produced some demo systems which use the IBM Via Voice synthesis system.

There are two basic ways to interface the synthesizer to the SIRIDUS baseline architecture. Either one builds an OAA wrapper and runs the synthesizer as an OAA agent, or one builds a TrindiKit wrapper and runs the synthesizer as a TrindiKit module. For example, the Trindikit wrapper for ViaVoice consists of a module (`output_viavoice`) which, when called, will take the text string in the OUTPUT field of the total information state and pass it to the ViaVoice TTS engine using a system command-line call (`cmdlinespeak`). The text will also be printed on the screen.

Chapter 5

Advanced features for speech synthesis

A feature which we have developed using the Festival System is the ability to tell what the synthesizer has just said. This is useful for knowing what the system said just before a barge-in or backchannel. Knowing what was said, will eventually allow the system to know what the barge-in is about or is likely to answer. Since humans backchannel or say agreement words during conversations, it will be possible to mark statements which the system says and the person agrees with as grounded or a part of shared beliefs. The shared beliefs would be added to the Information State as the conversation proceeds and theoretically let the system do less confirmation. This is a new capability for dialogue systems, and will allow the systems to behave more like humans in a dialogue. Difficulties with this include deciding whether or not the person is barging in. Since a barge-in requires the system to stop talking, making an error in this decision will interrupt the flow of conversation. In order to reliably make this decision is will be necessary to parse the user utterance to find out it the statement is relevant to the present dialogue (is it an answer to a question, or is it a backchannel or another conversation). Answering these questions will be an important part of the research.

Chapter 6

Conclusion

This deliverable has shown some of the considerations which were involved in deciding which speech synthesis system to use and what advanced features are necessary for dialogue system research. We hope to provide OAA wrappers for several speech synthesis systems, especially the more natural sounding large unit synthesis systems, later in the project.

Bibliography

[1] A. Black and P. Taylor. Chatr: a generic speech synthsisi system. In *Proc. COLING94*, pages 983–986, 1994.

[2] R. Carlson, B. Granstrom, and S. Hunnicott. Multilanguage text-to-speech development and applications. In William Ainsworth, editor, *Perception and Comprehension*. JAI Press, 1989.

[3] S. R. Hertz and M. K. Huffman. A nucleus based timing applied to multi-dialect speech synthesis by rule. In *Proceedings of ICSLP92*, volume 3, pages 1171–1174, Banff, Canada, 1992.

[4] A. Hunt and A. Black. Unit selection in concatinative speech synthesis system using a large speech database. In *Proc. ICASSP96*, volume 1, pages 373–376, 1998.

[5] Learnout and Hauspie (www.lhs.com). Website. Technical report, Learnout and Hauspie, Ieper, Belgium, 2001.

[6] J. Olive. Rule synthesis of speech from dyadic units. In *Proc. of ICASSP77*, pages 568–570, 1977.

[7] R. Sproat, editor. *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Kluwer Academic, Dordrecht, Boston. London, 1998.

[8] P. Taylor, A. Black, and R. Caley. The architecture of the festival speech synthesis system. In *Proc. of the 3rd ESCA Workshop on Speech Synthesis*, pages 305–310, 1998.