# Incremental speech synthesis

**Jens Edlund**
KTH Speech Music and Hearing
Stockholm, Sweden
edlund@speech.kth.se

## Abstract

This demo paper describes a proof-of-concept demonstrator highlighting some possibilities and advantages of using incrementality in speech synthesis for spoken dialogue systems. A first version of the application was developed within the European project CHIL and displayed publically on several occasions. The current version focuses on different aspects, but uses similar technology.

## 1 Introduction

Human interaction with spoken dialogue systems differ in many ways from their interactions with each other. One notable example is that spoken dialogue systems tend to have a strict concept of *turns* which makes the dialogue more similar to a ping-pong game than to humans conversing. Given that we aim at creating spoken dialogue systems that can engage in human-like conversation (note that although this is the case for most dialogue work at KTH, it is not true by necessity; for a discussion see Edlund et al., 2008), this rigid dependence on turns needs a solution. The present paper discusses a step in that direction: the use of incremental varieties of speech synthesis. A brief background and discussion on incrementality in spoken dialogue systems is given, followed by a discussion of the specific requirements an incremental speech synthesis should meet, and a presentation of a prototype system meeting some of these requirements.

## 2 Background

If a spoken dialogue system is to achieve the responsiveness and flexibility shown by human interlocutors, it is essential that they process information incrementally and continuously rather than in large (utterance, or turn sized) chunks (e.g. Allen et al., 2001). In our work with the Higgins spoken dialogue platform (Skantze et al., 2006) we have investigated incremental input in the incremental robust interpreter Pickering (Skantze & Edlund, 2004) and online analysis of prosody in **`/nailon/`** (Edlund & Heldner, 2006). We have investigated methods and ideas that require a spoken dialogue system to be incremental in order to be fully exploited, such as the use of brief single word feedback utterances and brief feedback grunts (Edlund et al., 2005, Wallers et al., 2006). An incremental system is of little use, however, unless it is incremental throughout, and we now turn to speech synthesis.

Incremental speech synthesis makes possible a wide range of behaviours that are common amongst humans in conversation, but that are currently unavailable to spoken dialogue systems. Barge-ins – human users barging in to the computer's speech – for example, are currently handled in one of two ways: the system either assumes that the ongoing utterance is completed, which causes mismatch between what the system believes it has said and what the user has actually heard, or it assumes that the ongoing utterance has not been spoken at all, which causes unnecessary repetition. A system that produces its speech incrementally could handle barge-ins in a more flexible manner. Another example is self-barge-ins. People regularly change their minds about what they are saying, and often do so seamlessly and without restarts. Incremental speech synthesis makes it possible to mimic this behaviour as well.

## 3 Requirements

Examples of requirements for incremental speech synthesis:

- *Must know what has been said.* In order for the dialogue system to make informed decisions on what to do when a

user barge-in has occurred or when it changes its mind about what to say, it needs to be informed of what has currently already been said. An incremental speech synthesis system should continuously provide feedback on its progress to the dialogue system.

- *Must be able to halt, then continue/break as well as stop*. Many user barge-ins are caused by events that are unimportant for the interaction, such as a door slamming or someone coughing. These events are often brief, and a system would be better off simply halting briefly when a noise is heard, and continuing at where it left off of the noise rapidly dissipates. Similarly, if a sound that causes the system to halt is subsequently interpreted as feedback, the system could again simply continue what it was saying, rather than restarting the entire utterance or skipping the remainder of it.
- *Must be real-time and online*. Although slightly circular (as incrementality is often used as a way of achieving onlineness), an incremental speech synthesis system must be able to act in real-time and with a small and constant latency.

## 4 Prototype

The prototype presented here takes text and mark-up as its input and produces speech as its output. It is based on the standard diphone synthesis used and developed at KTH, with additions to meet some of the requirements listed in section 2.

The synthesis uses the timing information on word level (start-of-word, end-of-word) produced by the standard system to keep track of what words have been said. Together with mark-up mapping word sequences to specific semantics, this meets the requirement that the system know what it has said. Note that in the type of complex research system we are chiefly addressing here, it is common practice to modularise the system to the greatest extent possible, and the speech synthesis module in such a system would often be designed to work with little or no knowledge of the semantics of what it produces – these are the responsibility of higher-level modules. Incremental synthesis presents no pressing reasons to step away from this principle. The present solution is to have the module producing the surface representation of the utterance (which is what the speech synthesis takes as input) also generate mark-up tying words to abstract labels referring to meaning. The synthesis, then, only know how to read these labels.

In the current implementation, utterances are pre-synthesised and then played back. To halt, in the simplest case, is merely a matter of temporarily pausing the playback. There are complications, however. If the stopped mid-utterance, chances are that the stop will occur mid-word, which makes it difficult for the system to decide if the words should be deemed to have been said or not, bringing us back to the original problem of knowing what has been said. This can be solved by allowing the system to continue to the next semantic constituent break as specified by the input – normally a word – before stopping. This and several other solutions are being investigated in the prototype.

## References


Allen, J. F., Ferguson, G., & Stent, A. (2001). An architecture for more realistic conversational systems. In Proceedings of the 6th international conference on Intelligent user interfaces (pp. 1-8).

Edlund, J., & Heldner, M. (2006). /nailon/ - software for online analysis of prosody. In Proc of Interspeech 2006 ICSLP. Pittsburgh PA, USA.

Edlund, J., Gustafson, J., Heldner, M., & Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. Speech Communication, 50(8-9), 630-645.

Edlund, J., House, D., & Skantze, G. (2005). The effects of prosodic features on the interpretation of clarification ellipses. In Proceedings of Interspeech 2005 (pp. 2389-2392). Lisbon, Portugal.

Skantze, G., & Edlund, J. (2004). Robust interpretation in the Higgins spoken dialogue system. In Proceedings of ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction. Norwich, UK.

Skantze, G., Edlund, J., & Carlson, R. (2006). Talking with Higgins: Research challenges in a spoken dialogue system. In André, E., Dybkjaer, L., Minker, W., Neumann, H., & Weber, M. (Eds.), Perception and Interactive Technologies (pp. 193-196). Berlin/Heidelberg: Springer.

Wallers, Å., Edlund, J., & Skantze, G. (2006). The effects of prosodic features on the interpretation of synthesised backchannels. In André, E., Dybkjaer, L., Minker, W., Neumann, H., & Weber, M. (Eds.), Proceedings of Perception and Interactive Technologies (pp. 183-187). Springer.