An Engine-independent Text-to-Speech Workplace

Margot Mieskes

European Media Laboratory GmbH, Heidelberg, Germany

margot.mieskes@eml-development.de

Abstract

We present a workbench for access to Text-to-Speech engines. The workplace, a web-based graphical user interface, is intended to be engine-independent, allowing the user to not worry about the interaction with the specific engine, but to focus on his/her task and create a good synthesis result. Additionally, the workplace offers support for non-expert users in specific tuning and interaction tasks, such as phonetic transcriptions or creating a lexicon for usage during synthesis. We also present two application scenarios which were the basis for creating this workplace and the current status of the workplace.

Index Terms: Text to Speech, web-based application, engine independence

1. Introduction

Text-to-Speech (TTS) has achieved a maturity, which allows it to be used in real-life applications that are not restricted to very narrow domains. But the tools offering access to TTS are (at least partially) still far from intuitive to use for non-experts. Some require good knowledge about the specific phonetic alphabets used, some require the use of external tools to get additional information (for example on the length of a sound file) and still others require knowledge about internal workings of TTS to be able to make full use of a specific feature. Therefore, they are still hard to use for non-expert users, although the TTS engines themselves could support them in their tasks. On the other hand tools available as web application rarely offer many options or possibilities to improve the synthesized speech or they are limited in the amount of data they can process, as they only serve for demo purposes. Of course there are a number of tools available listed for example here [2].But if demos are available they are often restricted as is detailed here [1].

1.1. Application Scenarios

The workplace in its current state was created with two applications in mind, namely audio descriptions for the blind, and tourist guides for museums and towns. In both cases the tools are rarely used by TTS experts. Nevertheless, the interfaces normally offered by the TTS providers overwhelm a non-expert user. To reduce the chasm between the potential users and the TTS engines, we decided to provide a tool which is easy to use and offers help in common tasks, *e.g.* when creating the phonetic transcription of a word. The target applications can moreover be multi-lingual, but the quality of voices across the different languages, as well as the combination of available languages, varies for most TTS engines. So if the user who wants to create a tour guide or an audio description needs a specific language combination they have to accept the quality changes or deal with several engines. The quality differences are unacceptable for most people hearing the final result. On the other hand, simultaneously dealing with several engines is cumbersome for the user, as various engines offer different interactions, lexicon formats, phonetic alphabets and sometimes even differ in offered features – especially commercial tools, which want to differentiate themselves from their competitors. This nonuniformity, which makes it infeasible to simply switch TTS engines, also extends to open source tools, such as MARY [3] or Festival [4]. Therefore, an engine independent workplace is a highly desirable feature for multi-language applications.

We present a graphical user interface which is aimed at bridging this gap between the non-expert users and the capabilities of today's TTS engines.

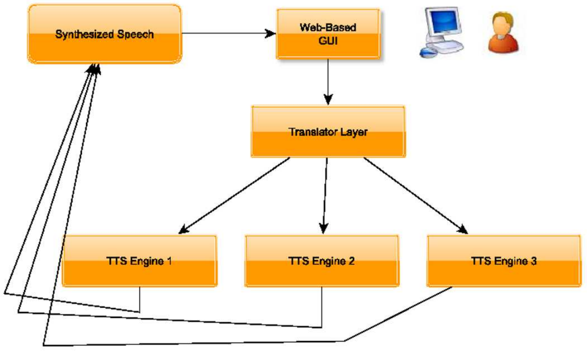


Figure 1: *The schematic Architecture of the TTS Editor.*

2. Architecture of the Workplace

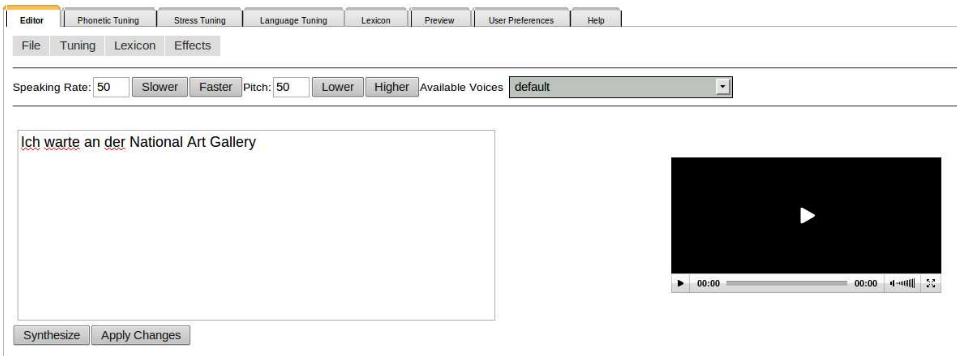
We created a web-based graphical user interface (wGUI) to available TTS engines. The basic architecture is schematically depicted in Figure 1. The wGUI offers a text editor as well as the ability to listen to the synthesis result and to access various tuning tools. We describe these capabilities in more detail in Section 3. We introduced a *translation-layer* between the wGUI and the engines working in the background. This layer takes care of translating the input from the wGUI to a representation suitable as input for the specific engine. The internal representation at the backend of the wGUI uses XML format, which can also easily be translated into other formats such as SSML. This translation layer allows for the engine independence of the workplace, as this translation layer can be extended for various engines. The actual engine then produces the synthesized speech, which can be listened to.

3. Workplace Features

Here, we present the features that have already been implemented and are available to use. In Section 4.1 we describe some of the planned features.

3.1. The Main Editor

As mentioned in Section 2, the workplace offers an editor for entering text. This is shown in Figure 2. The editor offers the capability of importing text, which is convenient especially for longer documents. Alongside the editor, the workplace offers a player for listening to the resulting synthesis, with switches for increasing or decreasing speaking rate and pitch and also for switching the voice. Besides these controls, Figure 2 also shows the main menu, through which the user can create new projects



Speaking Rate: 50

Pitch: 50

Available Voices default

Ich warte an der National Art Gallery

Figure 2: *The main Editor of the Workplace.*

or export the resulting synthesis to their local file system for further usage either as .wav or as .mp3 file.

3.2. Tuning Methods

The workplace offers various tuning methods to improve the quality of the synthesis result.

The *stress tuning* allows the user to change stress on words and/or syllables. This control over the word stress in a sentence allows the user to change a sentence's prosody profile. The meaning of a sentence may change as the prosody changes, *e.g.* the different meanings of: Watch the *blue* car. vs *Watch* the blue car. Additionally, words can change their meaning based on stress like *Balsamico will record a record.* Figure 2 shows the tab for the interaction with the stress tuning method. Here, the user can assign stress to whole words, but also to single syllables.

The *phonetic tuning* allows the user to change the phonetic transcription of specific words, that are not correctly synthesized. The phonetic tuning interaction is shown as a tab in Figure 2. The phonetic tuning can be done on the whole text or on parts of the text. The phonetic transcription is done automatically, using a grapheme-to-phoneme (g2p) [5] conversion method. In German, compounds are very productive and can become very long, so it is likely their g2p conversion might be poor. Therefore, the user may have to correct the suggested conversion. If the same word occurs several times in the text and is not correctly pronounced in all cases, this word can be transferred to a user lexicon.

3.3. Lexicon Interface

The user lexicon is another important feature of the workplace. The user can add words and abbreviations, but also transfer words from the phonetic tuning directly. The lexicon interface tab is also shown in Figure 2. A corrected word can be copied from the phonetic tuning to the lexicon together with its transcription. The lexicon can store abbreviations with their full spelling, as well as a "soundslike" description of how the word is pronounced without knowing the proper phonetic transcription. The users can also add whole word lists, if the text they want to synthesize has an unusual or strange vocabulary. These lists can be automatically converted into their respective phonemic representation and corrected by the user, if necessary. Furthermore, the user can add information about a word's language. Imported words are a common phenomenon in any language. If an English or French word appears in a German text, the imported word should be handled appropriately. The information in the lexicon is then applied to the respective words during synthesis.

4. Summary

We presented a workplace which offers access to Text-to-Speech engines through a web-based graphical user interface. This interface supports non-expert as well as expert users in creating good quality synthesis with little effort and without the need to adapt to a new engine when it needs to be switched. The features presented here are fully implemented and functional. The interface itself was designed with two specific scenarios in mind, but can easily be extended to other applications.

4.1. Future Work

At a first stage, we focused on German. Future tasks include adding voices in other languages, such as English, making the system truly multi-lingual. As the quality of voices sometimes varies acrross engines, this could also mean extending the supported engines from Mary and Loquendo to others. We also plan on supporting audio editing via the waveform display, which currently only allows for zooming into the waveform. We observed that few TTS tools offer this feature, but we feel it is quite important. For the audio description scenario, we intend to add the capability of watching the movie together with the synthesized speech in a preview mode directly within the workplace. Finally, we want to offer the addition of new voices based on recordings and their respective transcriptions.

In terms of evaluating the tool as a whole acceptance by users will have to be tested and assessed in their every day life.

5. Acknowledgements

We thank other staff members for the fruitful discussions and their willingness to help testing. We are grateful to the Klaus Tschira Foundation for the financial support.

6. References

[1] http://www.laits.utexas.edu/hebrew/personal/tts/table.html

[2] http://www.filetransit.com/category.php?id=246

[3] M. Schröder and J. Trouvain, "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching", *Int. J. of Speech Tech.*, vol 6, pp. 365-377, 2003.

[4] P. Taylor, A. Black and R. Caley, "The architecture of the Festival Speech Synthesis System,", *3rd ESCA Workshop on Speech Synthesis*, pp. 147–151, Jenolan Caves, Australia, 1998

[5] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, 2008. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html