Development Challenges of Multilingual Text-to-Speech Systems

Kimmo Pärssinen & Marko Moberg & Mikko Harju & Olli Viikki

Nokia Technology Platforms
Tampere, Finland

{kimmo.parssinen; marko.moberg; mikko.a.harju; olli.viikki}@nokia.com

Abstract

In this paper the importance of wide language support in a Text-to-Speech (TTS) system is discussed. From a typical user's point of view it is important that the application is available in user's mother tongue. This applies also for TTS. However, localization of a TTS system requires a lot of work due to demanding and often expensive language development process. Some ideas how a common framework in language development would benefit the industry are presented. Also the possibility to use Speech Synthesis Markup Language (SSML) to provide information about language mappings is discussed briefly.

1. Introduction

One of the main problems in the current Text-to-Speech systems is the time consuming internationalization and/or localization process of the synthesis technology. Development requires knowledge about human speech production and about languages being developed. The actual implementation of a fully functional system, on the other hand, requires good software skills. It is often difficult to find a single person to master all the areas of TTS development. Instead, the experts of various fields must work together. Especially the development of multiple languages tends to require linguistic knowledge that can only be acquired by consulting the experts who are familiar with the given language. Therefore it is necessary to be able to separate the language creation process from the actual TTS engine development. This becomes even more evident when the TTS engine needs to support multiple languages at the same time. However, even if the language development would be fully independent of the actual TTS engine development, the development of a new language and voice is a very time consuming and expensive process. Also the problem is that once a language is developed for one TTS system it is not, at least directly, usable for another TTS system from a different vendor. Therefore re-using the available language resources becomes unnecessarily difficult and inefficient.

Currently Series 60 phones support over 40 languages and already have some voice user interface (UI) features built-in. An example is the speaker independent name dialling (SIND) system available in Nokia Series 60 phones. In SIND user says a voice command and hears the feedback using a TTS system. One of the main advantages of the SIND is that it has been internationalized for all the Series 60 languages so the user is able to use one's own mother tongue. The importance of the wide language coverage applies also for other voice UI features and technologies. If the language support is not wide enough, the technology easily remains a niche feature that is not widely used. It is also important to be able to provide all customers the same features regardless of their native tongue.

2. Language Development Framework

The SSML is designed to provide an XML-based markup language for assisting the generation of synthetic speech in Web and other applications. The role of the markup language is to provide authors of synthesizable content a standard way to control aspects of speech such as pronunciation and volume etc. across different TTS platforms. According to the SSML specification the intended use of SSML is to improve the quality of synthesized content, either using automatic or manually inserted markups. The original purpose of SSML is therefore not to tackle the issues of TTS systems language coverage or development. However, from the language development point of view it would be beneficial, if there would be a standard framework to describe the language development process and data formats used in the development. The framework should include information how to describe the speech database, units, parameters, extracted features etc. It should also include tools to describe natural language related usually domain specific transformations (conversion of abbreviations, cardinal and ordinal numbers etc.). The framework should also provide means to describe how the automatic segmentation of a text is done, e.g. word boundaries in Mandarin and Cantonese etc. This role does not suit for SSML as such - at least the SSML would have to be extended to include a new description language or a set of languages. However, a standardized language development or description framework would have clear benefits. The collected speech and language data would then have a common representation and could be more easily compiled to the final representation for the TTS system in hand. This would ease the language development process greatly since the same data could be more easily shared and re-used between different TTS systems. It would also make the language development more systematic and standardized as the interface to different TTS systems would be known. It is of course not easy if even possible to provide a description language sufficient to describe the natural language processing rules i.e. how to normalize the numbers, abbreviations and so on. However, even some standard, high level description language and framework would be useful and a good starting point.

3. Use of SSML in Language Mapping

SSML has a *voice* element that has an optional attribute *xml:lang* to control the desired synthesis language. A possible extension to this would be a list of languages i.e. fallback languages in case the first language is not supported by the TTS system. This optional list of languages would provide language mapping information to the TTS system. In addition to the language mapping element another element containing information how the original language should be mapped to the new target language would be useful. The idea would be to create an approximation of the new, unsupported language by using a language that the TTS system already supports. SSML could also contain some parameters to describe the typical intonation of the new language to be modeled using an existing language.

Another extension proposal is to include a separate element *read* that could be used to control the selection of the pre-processor in a TTS system. The idea is that for example a Finnish word in the middle of English text would be processed using Finnish pre-processor but the actual TTS voice would remain the same. The correct pronunciation could of course be achieved by using already available SSML elements but there the user of the SSML would have to know how to pronounce the foreign words. If the TTS system supports multiple languages it could choose the correct pre-processor without changing the voice. In a way element *read* separates the spoken language and read language. The benefit is that the voice remains the same and the user of this new tag needs no information about how the word should be pronounced.

Conclusions

The need for a standardized framework for language development has been presented. Whether the framework can be applicable in practice should be investigated. However, there’s a clear need to extend the language coverage of the existing TTS systems and for now language development is too expensive and complex to make the feature widely available. Another proposal is about language mapping rules using SSML. A new element was presented. It provides information how to use an existing synthesis language to create a close approximation of a new language. The last proposal introduces a new element to select the correct pre-processor i.e. reading language in a multilingual TTS system. The benefit is that the voice remains the same and the user of this new tag needs no information about how the word should be pronounced.