International Journal of Software Engineering and Technology (IJSET)
Vol. 1, No. 1, April 2016, pp. 29~34
ISSN: 2032-4038

Using FOSS TTS Developer Tool to build Malay TTS System

Sabrina Tiun*, Ahmad Lufti Salikin*, Siti Khaotijah Muhammad**
* Faculty of Technology and Information Sceince, Universiti Kebangsaan Malaysia
** School of Computer Sciences, Universiti Sains Malaysia

Article Info

Article history:

Received Jun 12th, 2015
Revised Aug 20th, 2015
Accepted Aug 26th, 2015

Keyword:

User Experience
Text-to-Speech Software Tool
Malay Text-to-Speech
Free/Open Source Software
Software Tool

ABSTRACT

In order to develop a complete and usable Text-to-Speech (TTS) system requires years of time, hours of human workloads and tons of knowledge from various field of subjects. However, with a simple and easy software tools to use and understand, the burden of developing a complete TTS for a specific language can be overcome. Thus, such existing software is able to cater for dynamic society that demands for less and less appliance to communicate with machines and prefers natural user interface, like voices and visual to interact with machine. In this paper, we will present our work and experience as a user in using a software developer tool called eSpeakedit (a free/open source software (FOSS)) of eSpeak TTS system to build a Malay TTS system within a short time. In this paper, not only our experience as a user will make other TTS researcher in Malay or non-Malay language be awared of the eSpeak developer tools and architecture, but other FOSS and non-FOSS researcher might be enlighten of such well-designed FOSS architecture as well.

Corresponding Author:

Sabrina Tiun,
Faculty of Technology and Information Sceince,
Universiti Kebangsaan Malaysia,
43600 Bangi, Selangor Darul Ehsan, Malaysia.
Email: sabrinatiun@ftsm.ukm.my

1. INTRODUCTION

Taking users' involvement in the shaping the new product and refine an already-popular product is a great concern in software usability engineering [1]. The recent term called the UX or user experience has broadening the term in the aspects of what software usability engineer investigating between a user and software. Mentioned by [2], the UX is a concept concerning more aspects of a user in more personal values like the feeling of joy, pleasure, fun and pride. Thus, we can conclude user's experience is one of the important element in software engineering.

In this paper, we will not taking an account the standard designed UX in reporting our experience as a user but rather like presenting our experience, a common user experience, in using a software tool for developing a software system. The software tool which we used was the eSpeakedit/eSpeak, a tool for a TTS engine named eSpeak [3]. What we aim to contribute from this paper is to give insights to the software FOSS software developer of the other fields on how eSpeak, in our opinion, is cleverly designed, that it be able to help in satisfying the demand of dynamic society in term of communication across multi-language.

In this paper, we also will give details on how we build our Malay TTS using the FOSS developer up to the analysis of the output generated by the built TTS system. Thus, we can say, this paper serves objectives: (1) to enlighten our reader about our experience as a user of using FOSS software for developing a complex system, and (2) detail description of the Malay TTS development. For that, we will give a glance on eSpeakedit tool in section 1.1 and the overview of Malay TTS in section 1.2. Afterwards, we will present how we build a Malay TTS engine in section 2. Section 3 will be result evaluation on our Malay TTS and also a brief discussion and summary of our experience as a user in using eSpeakedit/eSpeak software.

Finally, in section 4, we will conclude our paper and discuss briefly on the future work of Malay eSpeak TTS engine.

1.1 Espeak:eSpeakedit

ESpeak is a multi-lingual software speech synthesizer which has been used in various speech enabled application. Most well-known application that uses eSpeak is Google Speech-to-Speech machine translation, the translation.google.com. The successfulness of eSpeak being widely and popularly used is because; it is a free open-source code speech synthesizer and new language is able to be built based on its engine in a very short time period with limited knowledge in speech processing and linguistics. Thus, with such design, is able to provide synthetic voices for any language compared to other FOSS speech synthesizer like Festival [4] or Mary TTS [5]. The user interface tool which is used by developer to develop a specific language of speech synthesizer based on eSpeak engine is called eSpeakedit. ESpeakedit can be downloaded freely at http:eSpeak.sourceforge.net/editor. Figure 1 shows the screenshot of the eSpeakedit main interface.

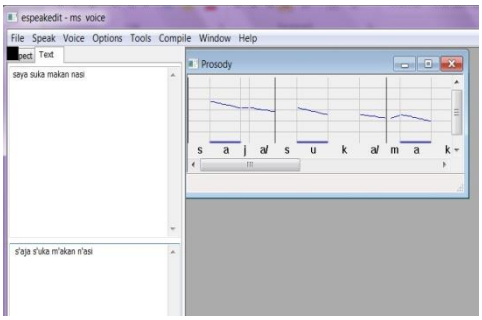

Figure 1. Part of eSpeakedit main screen

1.2 Overview of Malay TTS

Roughly, the research work of Malay TTS can be distinguished into three categories based on the speech synthesis techniques they are using:

a. Syllable as synthesis unit: Basically, this approach uses syllables as synthesis unit and only one instance existed for each of synthesis unit. During the synthesis process, none or less signal processing is applied between the concatenation points. Since the speech database is a fixed inventory, the same synthesis unit is used in any location of the word in a sentence and thus, high occurrence happen to both prosodic and segmental mismatch. SM TTS system that was built based on this approach is the Malay Speech Synthesizer (MSS) system developed by Samsudin et al. (2004). Later, Samsudin (2007) proposed to adapt corpus-based approach to improve the MSS to be more natural-sounding and re-named the TTS as MSS ver2. The more sophisticated system that uses the approach of using single instance of syllable as speech unit was proposed by El-Imam and Don (2000). 2. Synthesis with fixed inventory. In this approach, diphone is used as synthesis unit in a fixed inventory database and signal processing is used to modify the prosodic values and smooth concatenation points. Fasihtm, which was developed by MIMOS, uses this approach. The system uses rule-based approach that predicts the pitch values at phrasal and word levels. The prediction depends on the location of breaks within the target sentence, and then locates the breaks based on Part-of-Speech (POS). The predicted pitch at certain segment of all the target synthesis units will be used as an input to Mbrola engine (Kow, 2005).

b. Corpus-based synthesis approach: Two known research works on SM TTS based on unit selection synthesis technique are the SM TTS system based on Festival framework by Loo et al. (2007) and another one is a TTS system called Malay Text-to-Speech (MTTS) developed by Tan and Shaikh-Salleh (2008) from Universiti Teknologi Malaysia (UTM). Both of these systems are not using prosodic features in selecting speech unit but claimed that the naturalness of speech output generated by unit selection outperformed the diphone-based synthesis technique. Another attempt of creating Malay TTS based on corpus-based approach is by Tiun et al (2011). The author believes the naturalness degradation scenario is caused by the objective of those SM TTS systems on prioritising flexibility rather than naturalness at the first place. Therefore, Tiun et al (2011) worked on prioriting naturalness before

flexibility, where the author adopting an Example-based Machine Translation architure to be Malay TTS system and uses parser as the tool to select the most appropriate speech units, in which, the speech units consists of sentence, phrase, word and sub-syllables.

Based on the above overview on Malay TTS, we can presume that most of the prominent works on Malay TTS are using concatenative approach and with or without corpus-based technique for selecting units. This trend is very much influence by the current state-of-arts of the speech synthesis. Some works on formant-based synthesis for Malay does exist, such as the work of [11]. However, since none of this Malay TTS unable to be available freely to the other speech-based application, thus this discourages the speech-based application or potential speech-based application to be developed for Malay language. A very good example will be the translate.google.com, where the voice for Malay is unavailable. Therefore, we would like to fill in the void by creating a Malay TTS system which is flexible but with low quality of generated voice. Since we found out the FOSS of eSpeak be able to provide us facility to achieve our goal, we also need to highlight to the other developer of FOSS of how eSpeak system has satisfied us as a user.

2. MALAY ESPEAK

The goal of our Malay TTS project is to build a flexible Malay synthesizer system within a short time. Based on FOSS speech synthesizer software architecture called eSpeak and used its developer tool called eSpeakedit, we intent to build a flexible but less natural-sounding Malay speech synthesizer engine. In order to build a very basic TTS engine for a specific language, eSpeak requires these data: (1) A set of new language phonemes file, (2) Grapheme-to-Phoneme (G2P) rules files and (3) Pronunciation dictionary file. Before we describe in detail on the required files that need to be generated for Malay Espeak, we will explain briefly about Malay phonemes and G2P rules.

2.1. Malay Grapheme and G2P rules

The Standard Malay phonemes consist of 27 consonants and 6 vowels [7. The two tables below present the Malay vowel inventory and consonant inventory, respectively. In [12], several rules of Malay G2P are discussed. Some of these rules are: (1) the glide formation, where

2.2. Malay eSpeak file

In previous work on Malay eSpeak [8], in order to create a phoneme file for Malay voice, the English set of phoneme was used as a phoneme base file. File created was named ‘ms’, thus the name of Malay voice in eSpeak is also ‘ms’. The name is created based on the Malay ISO language code required by eSpeak. The performed process generated two files named *ms* (phoneme base file) and *ph_Malay* (Malay phoneme file).

For the creation of G2P rule file, an Indonesian voice file was used as a guide. By removing irrelevant Indonesian G2P rule and adding Malay G2P rule to the file, a Malay G2P for eSpeak was created. This was done by checking the Indonesian file, and unsuitable rule removed, changed and for Malay language. The checking of the pronunciation can be checked using the eSpeakedit by pressing the button ‘translate’ and ‘speak’. The button ‘translate’ will display the phonemic of the given orthographic word, and the button ‘Speak’ will be used to hear how the given word will be pronounced. This is how G2P can be manually checked in G2P rules file. In [8], the obvious change was to add the Malay G2P rule of ‘e’ pepet and ‘e’ taling. This is one of the obvious different in Malay and Indonesian language. At this the stage, a file named *ms_rule* was created. The creation of Malay G2P file in [8] was done by referring to [6] and [7]. In order to create the Malay voice pronunciation dictionary file, [8] modifed the Indonesian pronunciation dictionary to become a Malay pronunciation. The modification was done by adding and changing the Indonesian pronunciation into Malay pronunciation. Example of modified pronunciation lexicon was the number ‘8’ synthesized as ‘delapan’ in Indonesia, but was modified it into ‘lapan’. File created from this process is the *ms_list* file. The process of checking the pronunciation was again by using the ‘translate’ and ‘speak’ buttons in the eSpeakedit tool. Finally, after the creation of all those files, a voice for Malay was ready to be setup in eSpeak Engine. In this paper, the voice created from work of [8], we named it as MS1.

2.2.1 Addition of Malay G2P rule in Malay eSpeak

As being mention in [x], there still a huge space for improvement to make Malay eSpeak sounded totally in Malay accent. One of them is to add the G2P rules. Some of the recent modifications to the Malay G2P rules for Malay eSpeak are shown the Table 1.

Table 1. List of modification to Malay G2P rules

group a	.group e	.group j
a) a _|a:	CAC) e (C E	A)j _|dZ
C) a (A aa:	_) e E	o)j (o _|dZ
C) a (w aa:	_) e (ma E2:	
A) a _|a:	_) e (mC E2:	
C) a (y aa	_t) e (mCu E2:	
C) a (_ @	_t) e (mCo E	
y) a (_@@	_C) e (C E2	
C) a (ys E		
.group k	.group t	.group u
_ke) k (_ k	A) t (_ t2	C)u (C_ o
	CA) t (C t2	

The group of /a/, /e/, /j/, /k/, /t/ and /u/ on the Table 1 referring to the types of phonemes where recent modification takes affect. The voice generated with the added modification is named as MS voice. The reference of the addition rules was look upon on the list of Malay G2P in [6]. The script in Table 1 is understoond by referring to the user manual of eSpeak in [3].

3. RESULT AND DISCUSSION

The first attempt of creating Malay Espeak [8], there was no formal evaluation was carried out. However, with the addition of more G2P rules, a small perceptual evaluation was carried out to see whether there is an improvement or not for the Malay eSpeak. We will describe in detail on the small perceptual evaluation and our experience as a user in developing Malay eSpeak in this section.

3.1. Perceptual evaluation on Malay eSpeak

The perceptual evaluation procedure conducted for Malay eSpeak was similar to the smoothnes test by [8] where the participants were asked to choose which words perceived as not smooth. Similar to our perceptual evaluation, except that the participants were asked to choose which words sounded much better, in a sense, the articulation of the perceived words are clearer. The dataset for the perceptual evaluation consists of voice previous Malay eSpeak [8] and current voice of Malay eSpeak based on the new addition G2P rules in Table 1. The voice of Malay Espeak by [8] was labeled as MS1 and the new voice of Malay Espeak was labeled as MS1. Twelve volunteered participants of native Malay at the range of aged 20-50, with 42% male and 58 % female, took part in the subjective evaluation.

Based on the added and modified rules of G2P in Table 1, for the perceptual evaluation, we generated voices of MS1 and MS for this list of words: *bakul (*basket*), basuh (*wash*), batuk (*cough*), betul (*correct*), catur (*chess*), dosa (*sin*), emas (*gold*), kek (*cake*), kita (*we*), lapuk (*vintage*), lebih (*extra*), mentimun (*cucumber*), mereka (*they*), pucuk (*bud*), pukul (*hit*), tembus(*penetrate*), tempuh (*run over*), tidur (*sleep*), tikus (*rat*), tubuh (*body*).* Using a specific program (Figure 2), each participant need to listen into both of MS1 and MS voices and click on the on the respective buttons to which voice sounded smoother, or better.

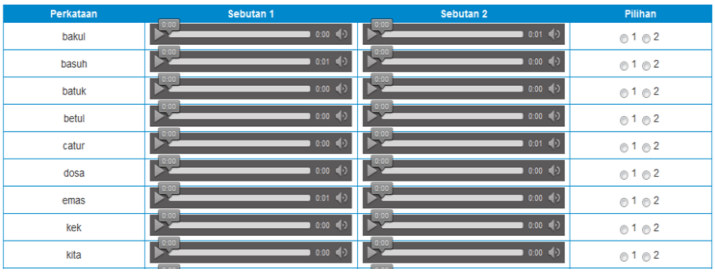

Figure 2. The screenshot of the perceptual evaluation program

In overall, based on the collected data from perceptual evaluation, most of the words synthesised by MS sounded much better than MS1. In Figure 3 below, averagely, more than 80 % participants perceived words generated from MS are clearer, and the highest percentage for word MS1 perceived as clearer than word MS is above 20%. Seeing the result of some from words of MS1 perceived clearer than MS, could

mean that probably some of the added rules degraded the sounding the generated voices. However, since the gap difference between MS1 and MS are quite huge, we can be optimistic that the new added rules are suitable to be added to the Malay eSpeak.

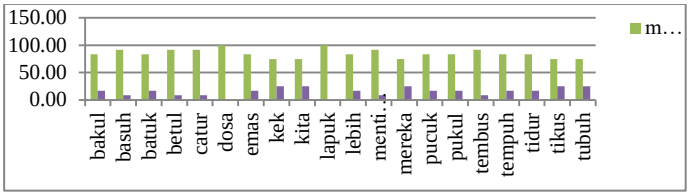

Figure 3. The percentage of clear articulated words of MS1 vs MS

3.2. User Experience Result

As mentioned in [8], the number of developers required for a TTS based on eSpeak engine maybe around one to two persons, depending on the individual expertise. If such person has both low to medium skills in linguistics and computer programming, maybe only one number developer is sufficient. However, a combination of a linguist and a computer programmer is needed if both of them only knowledgeable in their own fields. The time taken to complete a basic TTS system is also depending of the available resources. If one does not have any data for the G2P rules, non-existing pronunciation dictionary and unknown list of phonemes of a specific language, the duration may require months to get the project completed. As for Malay language, a list phoneme and G2P rules are available, plus Malay is a phonemic language, thus [8] only took about two weeks with five days a week, which each day was spent about only six to eight hours, to get the project completed. The most difficult part in the project was to understand the syntax of the files used in eSpeak engine. Fortunately, a basic Indonesian eSpeak TTS engine is available, therefore the available Indonesian files speed-up our work on preparing the data files for the Malay TTS engine.

The Malay eSpeak voice, named MS, the user has no background on linguistics or any Natural Language Processing. He is a native Malay speaker that has only programming skill and background, and the improvement work was done only when he has a free time. It took him more than one month to understand the eSpeak architecture and add the new improvement on the Malay eSpeak done by [8].

Table 2. The Summary of User Experience [8]

Items	*Value*
Required Time	Low to medium (*1 week to months*)
No. of Labors	1 to 2 persons (*depends on individual expertise*)
Required Equipment	Low (*basic desktop window with speakers*)
Required Expertise	Linguistics (*low- medium*); Computer skill (*low to medium*)

ESpeak can be developed in many platforms and required a very little resource. The voices of Malay eSpeak MS and MS1 developed our Malay eSpeak on a window platform. For the voice of MS1, the voice was created using Intel Pentium Processor with XP window. Since both Ms and MS2 do not require recording activity, sound proof recording room and high-quality microphone are not needed. Table 1 summarizes the experience in developing the Malay eSpeak TTS engine for both voice MS and MS1

Based on these experiences of creating Malay eSepak for voice MS1, [8] suggested that in order for any language to be developed in a faster rate, find a similar language that is already available in eSpeak as a guide and reference. However, if one cannot proceed with this process, the developer of the eSpeak engine is willing to be consulted.

4. CONCLUSION

The good concept of eSpeak software architecture is that, it only requires a developer to know about the knowledge on the data (language and linguistics) which are required to build a TTS system, without making it a burden for the users to understand other concepts of TTS software; i.e signal processing, machine learning and other things. In such way, the eSpeak will tremendously beneficial for people or researchers who only have the knowledge of a linguistic of a specific language but wanted to build a TTS system. Thus, this speech developer tool, eSpeakedit/eSpeak, is able to feed to one of the demands in dynamic society, that

is to enable society to communicate among each other across language barrier. In the aspect of our work project, our Malay TTS project obviously has a huge space for improvement: First, is to get rid of the English and Indonesian accent, and second, to make the Malay eSpeak sounded more natural for Malay voice. However, with a very short time duration, we achieved our aim of building a flexible with acceptable quality of Malay TTS voice for certain kind of speech enabled applications. In the future, we hope to see speech synthesis can be advanced into not only naturally sound and contains emotions, but a synchronization with facial expression like other languages such as in [11]. Furthermore, Malaysia consists of many races and ethnicities with different dialects and languages of communication. Thus, the fast development TTS system with very few resources need is quite attractive. A good speech aqusition tool like in [12] maybe can use as a start to develop a multi-language TTS for Malaysian.

ACKNOWLEDGEMENTS

Part of this project work was supported by Universiti Sains Malaysia under grant *USM Incentive Grant* and Universiti Kebangsaan Malaysia under grant *UKM GUP Grant (GUP-2012-007).*

REFERENCES

[1] M. Duechting , D. Zimmermann and K. Nebe, "*Incorporating User Centered Requirement Engineering into Agile Software Development*," 12th International Conference on HCI International. July 2007, pp. 58-67.
[2] D. J. Mayhew, "User Experience Design: The Evolution of a Multi-Disciplinary Approach", in *Journal of Usability Studies*, Vol.3 Issue 3, pp 99 -102, 2008.
[3] "eSpeak Text To Speech", [Online]. Available at: http://eSpeak.sourceforge.net/ , [Accessed:September, 2012]
[4] "The Festival Speech Synthesis System", [Online] . Available at : http://www.cstr.ed.ac.uk/projects/festival/", [Accessed:September, 2012]
[5] "Mary text-to-Speech",[Online]. Available at:http://mary.dfki.de/, [Accessed: September, 2012]
[6] Tan T. P., "Grapheme to Phoneme System". Unit Terjemahan Melalui Komputer (UTMK), Penang: Universiti Sains Malaysia, 2008.
[7] Y. Maris, The Malay Sound System. Fajar Bakti, Kuala Lumpur, 1980.
[8] S. Tiun, S. K. Mohammed. "*Experience in using TTS developer tool to build Malay TTS System*", 6th Malaysian Software Engineering Conference (MySec12). 2012.
[9] S. Tiun. Natural Sounding of Malay Speech Synthesis Based on UTMK EBMT Architecture System. PhD Thesis, USM. 2011.
[10] "Adding and Improving language", [Online]. Available at:http://espeak.sourceforge.net/add_language.html, [Accessed: Septemebr, 2012].
[11] Y. Wang, X.Yang and J. Zou , "Research of Emotion Recognition Based on Speech and Facial Expression", in *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol.11 Issue 1, pp 83-90, 2013.
[12] H. Chen and H. Mao, "Application of Computer Software in Analyzing Sound Acquisition in Modern Standard Chinese ", in *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Vol.11 Issue 8, pp 4824~4831, 2013.

BIOGRAPHIES OF AUTHORS

Sabrina Tiun is currently a senior lecturer from Universiti Kebangsaan Malaysia. Her research work and interests range from Speech Processing, Natural Language Processing and Information Retrieval. She obtained her PhD in Natural Language Processing (Speech Processing) from Universiti Sains Malaysia, Penang, Malaysia.

Ahmad Lufti is a Master student of Universiti Kebangsaan Malaysia, and his research area currently in Speech Processing. Currently he is working at MIMOS (Malaysia) Sdn Bhd.

Siti Khaotijah Mohammad is a senior lecturer at Universiti Sains Malaysia, Penang, Malaysia. Her research niches are covers from Natural Language Processing, Computational Linguistics, Lexicography and Data Mining. She obtained her PhD from Universiti Sains Malaysia in the field of Computational Linguistics.