mENUNCIATE: DEVELOPMENT OF A COMPUTER-AIDED PRONUNCIATION TRAINING SYSTEM ON A CROSS-PLATFORM FRAMEWORK FOR MOBILE, SPEECH-ENABLED APPLICATION DEVELOPMENT

Pengfei Liu, Ka-Wa Yuen, Wai-Kim Leung and Helen Meng

Department of Systems Engineering and Engineering Management,
Shun Hing Institute of Advanced Engineering,
The Chinese University of Hong Kong, Hong Kong SAR, China
{pfliu,kwyuen,wkleung,hmmeng}@se.cuhk.edu.hk

ABSTRACT

This paper presents our ongoing research in the field of speech-enabled multimodal, mobile application development. We have developed a multimodal framework that enables cross-platform development using open standards-based HTML, CSS and JavaScript. This framework brings high extendibility through plugin-based architecture and provides scalable REST-based speech services in the cloud to support large amounts of requests from mobile devices. This paper describes the architecture and implementation of the framework, and the development of a mobile computer-aided pronunciation training application for Chinese learners of English, named mENUNCIATE, based on this framework. We also report a preliminary performance evaluation on mENUNCIATE.

Index Terms— multimodal framework, mobile application, cross-platform, CAPT

1. INTRODUCTION

Mobile devices, including smartphones and tablets, are becoming more powerful and prevalent than ever before. Currently, mobile applications are under active development, especially for the iOS and Android platforms. However, many constraints exist for development due to resource limitations (e.g. screen size, processing power, network connection, etc.) of mobile devices.

Mobile application development also involves the challenge of how to support different mobile platforms. There are many kinds of smartphone/tablet platforms, such as Google's Android, Apple's iOS, Microsoft's Windows Phone, Nokia's Symbian, RIM's BlackBerry, HP's WebOS, and Samsung's Bada, etc. Developers have to learn platform-specific technologies, e.g. Objective-C for the iOS platform, Java for the Android platform etc., to implement their applications for each platform, which is time-consuming and expensive. Therefore, a cross-platform technology is highly desirable for application developers.

Many cross-platform HTML-based mobile application development frameworks have been developed, such as jQuery Mobile[1], jQTouch[2], etc. However, this HTML-based approach has some limitations in accessing low-level device capabilities. For example, there is no support of audio recording, which is a mandatory feature in speech-enabled mobile applications. These device capabilities can only be accessed through the use of native code (e.g., Objective-C for the iOS platform, Java for the Android platform, C# for the Windows Phone Platform, etc.).

Our group has previously developed a web-based computer-aided pronunciation training (CAPT) system named ENUNCIATE [1, 2] for Chinese learners of English. ENUNCIATE is accessible via the URL `http://enunciate.se.cuhk.edu.hk` within the campus network of The Chinese University of Hong Kong. CAPT can bring benefits to English learners with a self-paced, personalized and anxiety-free learning environment, and resolve the shortage of qualified English teachers due to large amounts of non-native English learners. Making the CAPT application ubiquitously accessible via mobile devices will significantly multiply its benefits.

In this paper, we focus on multimodal, mobile application development to port ENUNCIATE onto mobile devices as a mobile CAPT application named mENUNCIATE. We aim to leverage the cross-platform advantage of HTML-based approach while getting rid of its limitations in accessing native device functionalities by providing an extendable plugin-based architecture, and also to provide scalable speech recognition and synthesis services through the use of cloud computing. Therefore, we have established a cross-platform, multimodal mobile application framework and implemented mENUNCIATE based on this framework. This paper describes the architecture and implementation of the framework, and the development of mENUNCIATE. We also report a preliminary performance evaluation on mENUNCIATE.

[1] `http://jquerymobile.com/`

[2] `http://www.jqtouch.com/`

978-1-4673-2507-3/12/$31.00 © 2012 IEEE

2. PREVIOUS WORK

Recent literature [3, 4, 5] shows that framework development for speech-enabled mobile applications is a very active field. Table 1 compares the frameworks of WAMI, mTALK and PhoneGap developed by MIT, AT&T and Adobe respectively.

Table 1. Comparison of different mobile application development frameworks.

	WAMI	**mTALK**	**PhoneGap**
Supported Platforms	iOS, Android	iOS	Cross-platform
Architecture	Client-server	Client-server	Native client library
Client Side	Customized web browser	Customized web browser	SDK
Server Side	Cloud-based speech server	Cloud-based speech server	Not available
License	Open source	Proprietary	Open source
Limitations	No support to access native device functionalities	Platform-dependent development	No support on speech technologies

Both the MIT WAMI Toolkit [3] and the AT&T mTALK framework [4, 5] use a client-server architecture. The client is a customized web browser with specific multimodal capabilities (e.g. capable of recording audio for speech recognition). The server provides cloud-based services for speech recognition and speech synthesis. The Adobe PhoneGap[3] framework, which focuses on client technologies, encapsulates device-related functionalities for different mobile platforms, as well as exposes them through JavaScript APIs. This enables application developers to implement a mobile application by using HTML, CSS and JavaScript.

However, all these frameworks have some limitations when they are used for developing a speech-enabled mobile application that also needs to access additional native device capabilities besides audio recording, e.g. camera or GPS. The MIT WAMI mobile browser does not provide APIs or an extension mechanism to access native device functionalities. AT&T's mTALK framework is proprietary and requires developers to implement their application for each mobile platform separately (i.e. platform-dependent development). The PhoneGap framework has no built-in support for speech processing technologies.

3. ENUNCIATE BACKGROUND

3.1. Architecture

The web-based ENUNCIATE system is developed on the WAMI Toolkit and adopts the Spring[4] and the Hibernate[5] frameworks with a client-server architecture, as shown in Figure 1.

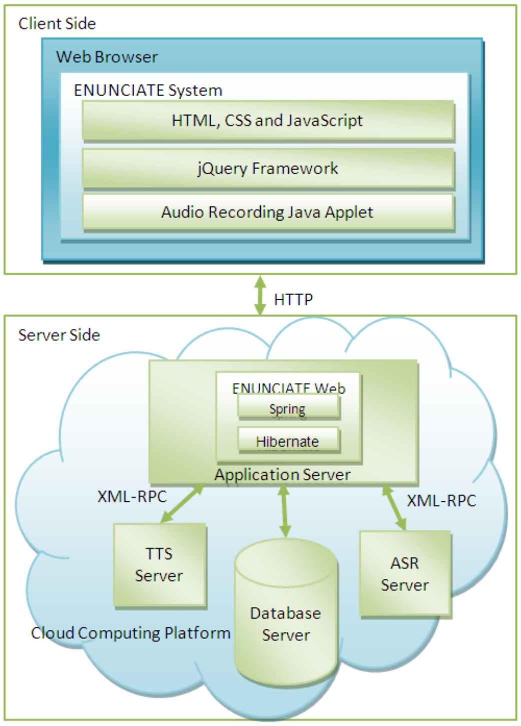


Fig. 1. The client-server architecture of ENUNCIATE.

On the client side, ENUNCIATE can be accessed via a web browser (e.g. Mozilla Firefox, Google Chrome or Internet Explorer). The GUI of ENUNCIATE is developed using HTML, CSS and JavaScript and adopts the jQuery framework. A Java Applet is used for audio recording.

On the server side, we have developed XML-RPC based automatic speech recognition (ASR) server and text-to-speech (TTS) server as well as database server for storing user accounts, pre-defined lessons and ENUNCIATE sessions [2] . The ENUNCIATE system is running on an application server (i.e. a Tomcat server). Besides, these servers can be hosted in the cloud, where multiple instances of the servers may be deployed depending on the requests traffic from clients.

3.2. Limitations

The ENUNCIATE system cannot be accessed directly in smartphones or tablets due to the following limitations:

(1) The audio recording Java applet is not supported in smartphones or tablets.

(2) The GUI of ENUNCIATE, originally designed for PC monitors, is not suitable for mobile devices with a small screen. Moreover, the touch screen of mobile devices calls for new UI changes.

(3) The TTS and ASR functionalities of the ENUNCIATE system cannot be accessed directly from mobile devices.

[3] `http://phonegap.com/`

[4] `http://www.springsource.org/`

[5] `http://www.hibernate.org/`

4. THE CROSS-PLATFORM FRAMEWORK

To port ENUNCIATE onto mobile devices, we need to overcome the limitations mentioned above. In addition, two more requirements should be fulfilled: (1) Support different mobile platforms; (2) Provide an extendable architecture for sustainable development.

Therefore, we have established a new multimodal, mobile application development framework with the client-server architecture, as shown in Figure 2.

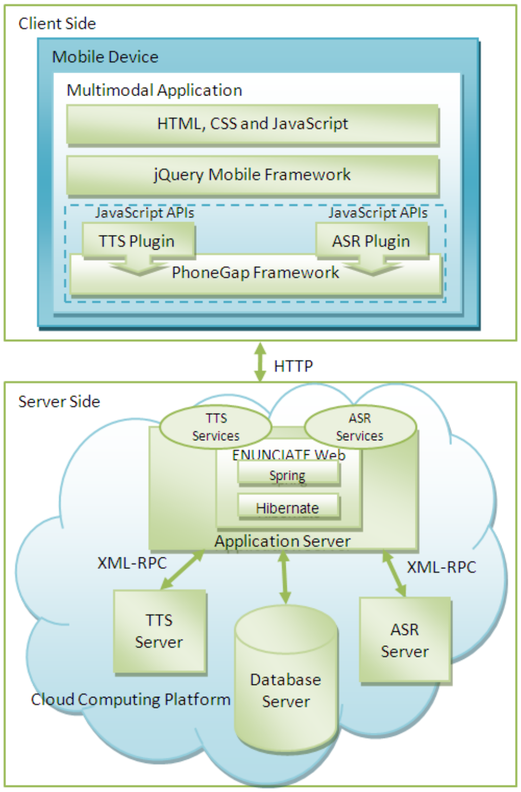


Fig. 2. The client-server architecture of the multimodal mobile application framework.

The client side resides on a mobile device (e.g. iPhone/iPad, Android Phone/Tablet, Windows Phone/Tablet, etc.), where a multimodal mobile application is installed. The dotted rectangle on the client side includes the PhoneGap framework, the TTS plugin and the ASR plugin. Each PhoneGap plugin is composed of two parts: (1) custom JavaScript APIs for application developers to access native functionalities and (2) platform-specific native code which is invoked from the custom JavaScript APIs.

The PhoneGap framework enables developers to implement a mobile application by using open standards-based HTML, CSS and JavaScript, which thus provides cross-platform capability. Moreover, the plugin-based architecture of PhoneGap brings high extendibility for developers to implement their application-specific plugins.

As shown in Figure 2, ENUNCIATE is extended with Representational State Transfer (REST) [6] based services, including the TTS services and the ASR services. These services are uniquely identified by Uniform Resources Identifier (URI) that can be invoked by the client side's TTS plugin and ASR plugin through HTTP protocol. These plugins can then provide the corresponding JavaScript APIs for developers to implement speech-enabled mobile applications.

To enable mobile devices to access the REST-based speech services conveniently, the client side and the server side communicate through the JavaScript Object Notation (JSON) message. JSON is a lightweight data-interchange format based on a subset of the JavaScript language. JSON is human-readable, easy to parse and widely supported by most programming languages.

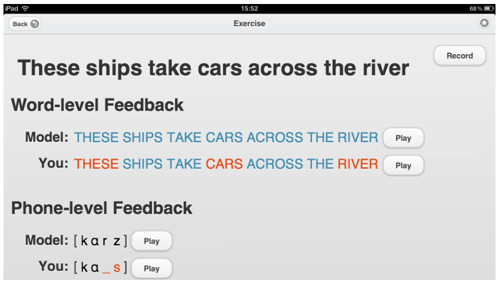
5. THE mENUNCIATE APPLICATION

5.1. Development

mENUNCIATE is developed based on our multimodal framework to port the web-based ENUNCIATE system onto mobile devices. mENUNCIATE inherits the same client-server architecture as ENUNCIATE, while extends ENUNCIATE with the REST-based web services and also implements an extendable plugin-based mobile client with the cross-platform GUI using HTML, CSS, JavaScript and the jQuery Mobile framework.

The major components of mENUNCIATE are the mENUNCIATE GUI, the TTS plugin, the ASR plugin and the REST-based TTS services and ASR services, as well as the backend application server, TTS server, ASR server and database server.

Figure 3 shows a practice on the sentence of "These ships take cars across the river.", for which mENUNCIATE detected that in the word "cars" the phone /r/ was deleted while the phone /z/ was mispronounced as /s/.



These ships take cars across the river

Word-level Feedback

Model: THESE SHIPS TAKE CARS ACROSS THE RIVER

You: THESE SHIPS TAKE CARS ACROSS THE RIVER

Phone-level Feedback

Model: [k ɑ r z]

You: [k ɑ _ s]

Fig. 3. The GUI of mENUNCIATE.

As explained below, mENUNCIATE provides three kinds of feedback: International Phonetic Alphabet (IPA) feedback, audio feedback and corrective feedback.

(1) IPA feedback: The user submits the sentence "These ships take cars across the river." to the server for pronunciation practice. This gets the canonical phonemes of the sentence by calling the text-to-phonemes service through the TTS plugin and shows them in the GUI.

(2) Audio feedback: The user listens to the model pronunciations (i.e. play the audio of this sentence). The TTS plugin will invoke the text-to-audio service and play the returned audio stream.

(3) Corrective feedback: The user starts recording an utterance and stops when finished. The ASR plugin will post the recorded speech to the ASR services for mispronunciation detection and diagnosis. The GUI will then highlight the mispronounced phones and their corresponding words in red, as shown in Figure 3.

5.2. Performance Evaluation

We have installed mENUNCIATE on an iPad (iPad 2) and conducted preliminary tests with the iPad connected to our internal Wi-Fi network. On average, it takes 2-3 seconds to get the diagnostic results for an utterance of 4-5 seconds long.

mENUNCIATE uses the same acoustic models (trained using the TIMIT corpus) as ENUNCIATE. To reuse the human-annotated transcriptions of the CU-CHLOE corpus [1] and make a quick performance evaluation for mENUNCIATE, we have collected a corpus named CU-CHLOE-MOBILE by using an iPad to record the playback of a subset of the CU-CHLOE corpus in a quiet recording room. The subset contains 600 utterances from 50 male and 50 female Cantonese-speaking learners of English reading the Aesop's Fable "The North Wind and the Sun", which has 6 sentences and covers all phonemic contrasts in English.

Table 2 compares recognition performance between the test set CU-CHLOE and the test set CU-CHLOE-MOBILE. FRR (False Rejection Rate) is the percentage of correct phone pronunciations that are erroneously rejected as mispronunciations, while FAR (False Acceptance Rate) is the percentage of mispronunciations that are accepted as correct pronunciations. DA (Diagnostic Accuracy) is the percentage of correctly recognized mispronunciations. As in [7], the evaluation experiment is configured to minimize FRR for the inherent trade-off between FRR and FAR, in order to avoid discouraging learners by rejecting correct pronunciations. Table 2 shows that there is slight performance degradation on the test set CU-CHLOE-MOBILE compared with CU-CHLOE. A possible reason is that CU-CHLOE-MOBILE was recorded in a quiet recording room and the microphone of iPad is of high quality.

Table 2. Comparison of recognition performance between CU-CHLOE and CU-CHLOE-MOBILE.

Test Set	FRR	FAR	DA
CU-CHLOE	9.20%	45.72%	48.86%
CU-CHLOE-MOBILE	9.54%	47.48%	48.41%

6. CONCLUSION AND FUTURE WORK

This paper has investigated the issues of cross-platform capability, architecture extendibility and application scalability on multimodal mobile application development. We have established a client-server based multimodal framework to solve these issues. In this framework, the client side provides plugin-based architecture for extendibility and enables developers to use HTML, CSS and JavaScript for cross-platform development; the server side provides REST-based speech services in the cloud that are scalable for large amounts of requests from mobile devices.

Compared with the WAMI Toolkit and the mTALK framework, our framework provides a cross-platform solution to develop speech-enabled mobile applications, and brings high extendibility with the plugin-based architecture. Based on this framework, we have successfully developed a mobile CAPT application for Chinese learners of English, named mENUNCIATE.

In the future, we will perform end-to-end empirical evaluation of the performance of mENUNCIATE, and work on acoustic model adaptation to sustain high performance in mispronunciation detection and diagnosis for mobile devices.

7. ACKNOWLEDGMENTS

This project is jointly sponsored by a grant from the CUHK Shun Hing Institute of Advanced Engineering and the NSFC/RGC Joint Research Scheme (project no. N_CUHK 414/09). We thank the three anonymous reviewers for their helpful comments.

8. REFERENCES

[1] H. Meng, W. Lo, A. M. Harrison, P. Lee, K. Wong, W. Leung, and F. Meng, "Development of automatic speech recognition and synthesis technologies to support chinese learners of english: The CUHK experience," in *APSIPA Annual Summit and Conference*, 2010.

[2] K. Yuen, W. Leung, P. Liu, K. Wong, X. Qian, W. Lo, and H. Meng, "Enunciate: An Internet-accessible computer-aided pronunciation training system and related user evaluations," in *Oriental COCOSDA*, 2011.

[3] A. Gruenstein, I. McGraw, and I. Badr, "The WAMI toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces," in *Proceedings of the 10th international conference on multimodal interfaces*, 2008.

[4] G. Di Fabbrizio, T. Okken, and J. G. Wilpon, "A speech mashup framework for multimodal mobile services," in *Proceedings of the 11th international conference on multimodal interfaces*, 2009.

[5] M. Johnston, G. Di Fabbrizio, and S. Urbanek, "mTalk - a multimodal browser for mobile services," in *INTERSPEECH*, 2011.

[6] R. T. Fielding, *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, Irvine, 2000.

[7] A. M. Harrison, W. Lo, X. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training," in *SLaTE*, 2009.