An embedded bimodal tool to enable second language learners to practise conversation online over unreliable internet connections

June Ye
Massey University
yecjn@hotmail.com

Russell Johnson
Massey University
R.S.Johnson@massey.ac.nz

Abstract

This paper reports on a project to prototype and evaluate an embedded e-learning tool that enables distance students of second languages to practise conversation with each other, or with their tutor, over slow or fast internet connections. In line with information appliance design concepts, the technology is invisible *to the student. Under fast internet conditions, the tool functions as an embedded VoIP telephone. But when it determines that one party's connection speed is too low for VoIP telephony it defaults to a Push-To-Talk mode analogous to "walkie-talkie" radio. Initial evaluation results have been promising although some time lag problems under very slow internet conditions are still being addressed.*

1. Introduction

Learning is a social relationship in which student-student and student-teacher communication plays a vital role. In distance learning ongoing communication must be facilitated by technology, be it the postal system, the telephone, satellite communication or the Internet. For distance education to be effective, communications need to be timely and accessible to all. Otherwise the teacher-student relationship will be disrupted.

There are several popular forms of computer-enabled communication today including email, asynchronous forums, real-time text-based chatting, internet telephony, and video conferencing. The first three have become standard features of e-learning systems. The last two have been introduced into multi-classroom teaching and corporate training, where learners have access to high-speed networks.

Before the popularity of voice over IP (VoIP) technology, PSTN (Public Switched Telephone Network) telephony was the most convenient and reliable way people could talk with each other when geographically separated. The main benefit of using VoIP is the low cost for long distance calling compared to using traditional PSTN. VoIP functionality can also be directly embedded into e-learning software and integrated with other functions such as text chat and video to provide custom communication features.

With the shift towards on-line delivery of distance education, the issues of computer literacy and accessibility come to the fore. According to UNESCO, less than two percent of the population in the developing countries have internet access [10]. Even among those distance students who can access the Internet at home or at a village school or learning centre, very few can be expected to have broadband service. Most would be making do with low speed and unstable dial-up connections. Internet telephony does not work over these connections. Moreover, installing and using standalone telephony software adds complexity to the learning environment.

This paper reports on a project to prototype and evaluate an easy-to-use embedded e-learning tool that enables distance learners of second languages to practise conversation with each other and with their tutors over slow or fast internet connections.

2. Background

LMSs (Learning Management Systems), where a student uses a browser running on top of a general purpose operating system to download material from a central repository page by page, are widely used in many universities and organizations for e-learning. This approach has some downsides for distance learning [9]. With a slow and unstable dialup Internet connection, it may be difficult for a remote learner to access LMS resources in a timely manner. Isolated learners need more support to enhance learning activities such as collaboration, interaction and individualisation, which are what a teaching-centred LMS lacks. Furthermore, a learner without computer literacy has to spend a lot of time to learn and become familiar with the multi-tasking GUI (Graphical User Interface) of the operation system before he/she can use any learning system.

The Learning Computer (LC) [1] addresses the limitations of Web-based LMS for distance education. It applies the concept of the single-activity information appliance [2] to e-learning, by replacing the operating system GUI with a specialised Learning Shell that transforms a PC or a laptop into an easy-to-use virtual appliance for distance study. The goal is to have the technology rapidly fade into the background, freeing the user to focus their efforts on the learning activity.

Unlike LMS which are light client and heavy server, the LC is heavy client to minimise the student's connectivity requirements and maximise integration and interactivity in the Shell. Updates take place automatically in the background, whenever an internet connection is available, or by CD-ROM. Because the simplicity of an information appliance minimises the load on the operating system and hardware, the Learning Computer can be run over cheaper, low spec or recycled computers [3].

The Learning Shell has a modular, component-based structure. Each component encapsulates a specific learning task (e.g. taking notes, or listening to a teacher enunciate words or phrases). The LC seamlessly integrates subsets of these components to support specific learning activities or modes (e.g. dictation in a second language).

Around the world, acquiring English language skills has become a key for unlocking access to higher education and improved employment opportunities. Here in New Zealand and in Thailand we are working with language teachers to develop the Learning Computer into an effective tool for helping students learn to read and speak a second language, thereby extending the reach of qualified native English-speaking teachers.

3. Conceptualisation and specification of online tool for conversation practice

Our goal with the LC is to use technology to bring some of the benefits of face-to-face teaching to distance learning, without complicating the learning environment. One of the greatest advantages on-campus second language students have over their distance counterparts is their greater opportunity to practise conversation. In second language learning, conversation is the most difficult task for most students, as it demands listening and speaking skills. To learn to converse in the real world, students have to spend a lot of time practising in different scenarios. Internal student have opportunities to practise with the teacher, tutor or other students face-to-face. Distance students, on the other hand, face the lonely prospect of practising with a tape recorder.

For this reason we decided to develop a component which would let distance students practise talking with each other and with their tutor over the Internet. In line with information appliance design concepts, the technology would be *invisible* to the student. In other words, the tool would be embedded into the LC as a learning component requiring no installation or configuration by the user. It would be simple and intuitive to use. And it would work even when internet access was too slow for conventional VoIP telephony.

The concept that emerged was of a *bimodal* conversation tool which automatically adapts to different connection conditions. Where both parties' connection speeds are fast enough, synchronous full-duplex conversation (FDC) is enabled – just like two users talking on the telephone. But if one party's connection speed is too slow, then the system defaults to an asynchronous Push-To-Talk (PTT) mode like the traditional half-duplex conversation over a radio channel, where one party has to press a button down to talk and then wait for a response from the other party before pressing the button to talk again.

Implementing the FDC mode requires embedding VoIP P2P functionality in the LC. Meeting the requirements for PTT is more complex. It requires integrating VoIP functionality with the recording, playback and transmission of audio files over the Internet. Many factors needed to be considered. The major one is users' bandwidths. VoIP is a service designed for broadband users. Lower speed dial-up connections usually only allow users to communicate with others from their personal computers by emails, online forums and text chatting. VoIP tools compress audio data at the sender, transmit the compressed voice stream between computers and/or telephones as IP datagrams, and decompress the stream at the receiver [4]. They do not work properly with low speed dial-up connections. A wide bandwidth is needed to transmit the audio stream. Otherwise the large amount of data that needs to be sent will cause congestion and severe delay. Severe voice delay will result in frequent breaking of the audio or video stream so that the receiver will find it very hard to understand the content. And it will destroy the proper communication sequence between two people.

Some VoIP services have become very popular. Skype [5] is in effect a very large P2P distributed system in which every client machine can be a server to other users without the need for a central server. Adobe Connect [6] integrates VoIP into an online conferencing system, which some universities have integrated into their distance teaching programmes. However, commercial VoIP applications like Skype and Connect are not suited to our project because they cannot be invisibly embedded into learning software.

Besides the bandwidth issue, reasonable voice quality should be assured to let the user hear clearly the talker's voice. The lower the echo and jitter, the better the voice quality.

Another problem is how users connect with each other for conversation when they belong to different networks. Even with P2P connections some control information must pass through a university-based server. Distance students typically study from their home and access the university through their ISP. Or they might connect from a workplace network. So the conversation tool must be NAT (Network Address Translation) or firewall friendly so that two or more users can connect with each other directly or via the server.

There are other issues such as echo cancelling, jitter buffering, voice samples compression, and UDP (User Datagram Protocol) socket programming. We had to be able to integrate the recording, playback and transmission of audio files to support the PTT mode. We needed to locate a VoIP SDK that allowed us to address all these factors. The SDK also had to be compatible with the IDE (Integrated Development Environment) in which the LC had been built. We selected Borland's Delphi 7 as our IDE for its high level RAD (Rapid Application Development) features, its ability of providing many low level access to the operating system APIs (Application Programming Interfaces), and its cross-platform portability with Linux [3].

The task was to design a reusable peer-to-peer conversation component for an LC for second language learning. Initially, however, the component is implemented as a standalone tool so that it can be tested and evaluated separately.

4. Prototyping

For the prototyping phase we began by designing some scenarios to define the functional and non-functional requirements of the tool. Figure 1, for example abstracts the PTT task flow when one party has a low speed connection, as well as when two parties both have high speed connections.

We analysed the existing LC architecture and its network communications and data management structure into which our tool would eventually be integrated. We were particularly interested in how data and messages are transmitted and processed between the server and the client. Unlike many other client/server-based systems, the LC network has a light server and heavy client structure – the client only accesses the server when it has to send messages to others or synchronize learning material with the server. The heart of this architecture is a set of high-level custom messaging protocols for periodic synchronisation of content between client LCs and a central Repository Manager (RM), including data files, database updates and student communications, over an unreliable network (or even by disk). This is necessary to support learners studying from rural areas or wherever else there are poor Internet connection conditions. The RM uses an FTP Server to control access to the Repository.

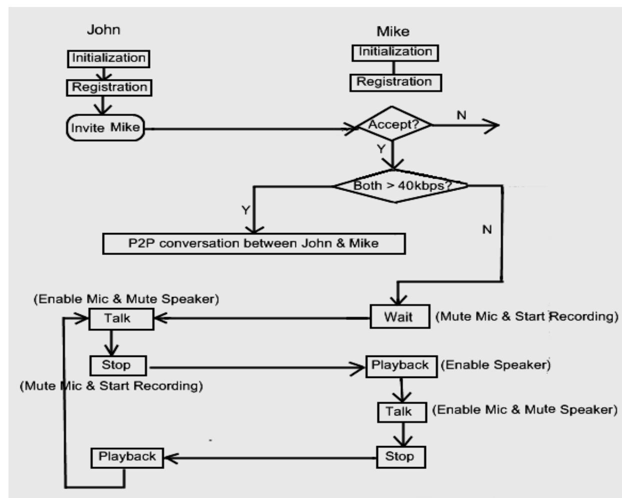

Figure 1. An Example Flow Chart including a scenario when one party is in low speed.

In a parallel manner, a set of protocols and a Call Manager have been implemented to manage P2P conversation. The Call Manager incorporates a third party VoIP server. When users come online they are automatically registered with the Call Manager, together with their connection speed and the discussion group they belong to. Their picture is then displayed to any other members of their group who are currently on line with a message that they are available. One party may invite another to a conversation by clicking on the second party's picture (Fig. 2).

If both parties are registered in the Call Manager with fast connection speeds then their tool is configured for FDC mode, the parties are registered in the Call Manager as busy, and other members of their

Figure 2. Esther's screen after being invited by John.

group are notified. The Call Manager plays no further role until notified by the clients that the call has ended (Fig. 3). However, if one of the parties is registered as having a slow connection, then the two client tools are configured for PTT mode. The Call Manager monitors the session and passes control information between the clients to keep them synchronised.

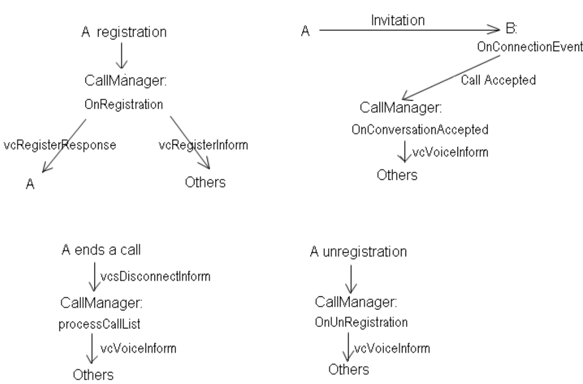

Figure 3. Message-passing to synchronise clients.

We tested the selected VoIP SDK in many ADSL and dialup connection pairs. The tests show 40 kbps is a safe lower limit for fast Internet connection when the SDK is used without encryption on voice. Several ways were tried for implementing the PPT mode efficiently. We first tried recording on the sender side, and then sending the recorded file to the receiver via the Call Manager. But this added to the system's complexity and meant long waiting times for the receiver. By experiment we discovered that even though the voice stream broke up at lower speeds and could not be understood in real-time, the VoIP client could accurately record the stream at the receiving end, enabling it to be played back clearly.

We also found that audio quality at the receiver's end was easily affected by the state of the sound card in the sender's computer. After tests, we found that dust-covered sound cards in older PCs could seriously degrade the voice quality of transmissions even when audio reception was unaffected. If we replaced the sound card in the sender's machine then the voice quality at the receiving end improved dramatically. However, this is not something that distance students could be expected to accomplish without technical help.

5. Evaluation

The purpose of evaluation is "to check that users can use the product and that they like it, particularly if the design concept is new" [7]. We have conducted an initial laboratory evaluation to provide proof of concept, to test the functionality of the underlying mechanisms and the usability of the interface, and to identify areas for further research and experiment.

It is very convenient to use the laboratory environment where the system has been developed to conduct the lab evaluation. Three volunteers with one of authors participated in two lab experiments. Each experiment has four tests and four short questionnaires, followed by one final questionnaire and interview. Observation and log files were also used for obtaining results from testers and data from the system. Interview questions were prepared just as a guideline. "Interviews typically include many open-ended questions where users are encouraged to explain themselves in depth" [8]. Testers completed the questionnaire before their interview, so that in the interview the experimenter could clarify or verify any important issues arising from the questionnaire, as well as from the observation.

The experiment was conducted in pairs with the experimenter connecting each volunteer's PC to either broadband or dial-up Internet service to test the four possible scenarios based on connection speeds (high-high, high-low, low-high, low-low). The volunteers followed prepared scripts based on oral English language exercises with each speaking turn taking 3-8 seconds.

All testers found the quality of the voice transmission was acceptable in either mode with low noise and almost no echo. The experiments also verified that all users could easily use and enjoyed the FDC mode. However, the PTT mode was less familiar to them and some initial instruction on its use was needed. While performance was acceptable when only one party was on low speed (the typical tutor-student scenario), they all considered that PTT was too slow when both parties were at low speeds (the typical student-student scenario). When one user was at low speed and the other at high speed, each user had to wait from 8 – 16 seconds – while their voice message was transmitted to the other party and the response was received and played – before they could start talking again. But when both users were at low speed, the waiting time doubled to around 16 – 32 seconds.

An analysis of log files on each client machine showed that there was a 7-12 seconds delay by the low bandwidth client in downloading some control messages from the Call Manager, apparently because the bandwidth was fully occupied by the voice stream traffic. We are exploring various avenues to reduce this latency. The simplest way is to locate a suitable SDK using a much lower bit rate codec than GSM 6.10 which we used in this project. Such voice codecs already exist both in proprietary and free form.

Results of these experiments (show as above) are very similar except some different personal usability opinions, so we think the lab experiment is sufficient. To test the tool for invisibility, we need to evaluate its functionality, usability and accessibility in the distance student's own, especially the rural learner for whom it has primarily been designed. This will be done in a field evaluation in future.

6. Conclusion and discussion

The goal of this project was to find a way to integrate online conversation practice between two distance students, or between a distance student and a tutor, into our Learning Computer architecture to enhance its application to a second language learning scenario.

From our initial user testing we conclude that the instant full duplex conversation is fast, straight-forward and telephone-like. But its broadband requirements make it most relevant to on-campus scenarios, and limit its applicability to more remote distance students unless they have access to satellite telephony.

Our push-to-talk mode works satisfactorily when only one party has a slow internet connection. This fits a typical tutor- distance student scenario.

However, in the most typical distance student-distance student scenario, two parties with low speed connections, our current implementation of PTT is too slow to maintain an effective conversation flow. While waiting up to 30 seconds for your learning partner to reply is a whole lot better than waiting for a week for a tape to arrive in the mail, it can still break concentration and disrupt learning. Therefore our priority is to research ways of reducing this wait time.

Another approach is to adapt the learning scenario so that the time lag is less of an issue, e.g. online dictation practice, or the integration of replay or revision features to occupy the wait time and maintain the learner's concentration.

We plan to further evaluate and refine the tool's interface, especially with a view to making the tool more intuitive in the PTT mode and obviate the need for the user to access help facilities. Through field testing we can determine whether we have assumed a wider familiarity with PTT "walkie-talkie" radio technology among distance students than actually is the case today.

The prototype also needs to be embedded into the existing LC architecture to evaluate how it can help a second language learner in improving oral skills.

To make voice communication happen between two people via very low Internet connections is a challenging task. But its potential for enriching the learning experiences of distance students the world over makes the effort worthwhile.

7. References

[1] Johnson, R., Kemp, E., Kemp, R., & Blakey, P. (2007). The learning computer: low bandwidth tool that bridges digital divide. *Educational Technology & Society*, 10 (4), 143-155. ISSN 1436-4522 (online) and 1176-3647 (print).

[2] Norman A. D. (1998). The Invisible Computer, Cambridge, Massachusetts, MIT Press.

[3] Allan, C. (2007). *Taking e-Learning Across The Divide. A Feasibility Study.* Institute of Information Sciences and Technology, Massey University, Palmerston North as at 17/01/07.

[4] Hersent, O., Petit, J.P., and Gurle, D. (2005). *Beyond VoIP Protocols – Understanding Voice Technology and Networking Techniques for IP Telephony*. John Wiley & Sons Ltd., England.

[5] http://skype.com/intl/en/

[6] http://www.adobe.com/products/connect/

[7] Sharp H, Rogers Y and Preece J (2007): Interaction Design: beyond human-computer interaction, 2nd ed., John Wiley and Sons Ltd , p586.

[8] Nielsen J (1993): Usability Engineering, Academic Press, p211.

[9] Johnson, R.; Kemp, E.; Kemp, R. & Blakey, P. (2002): From Electronic Textbook to Multidimensional Learning Environment: Overcoming the Loneliness of the Distance Learner. Proceedings of 2002 International Conference on Computers in Education, IEEE Press, Vol. 1, pp632-636

[10] Bindé, J. (ed) (2005):Towards Knowledge Societies. UNESCO Publishing, Paris. ISBN 92-3-204000-X. http://unesdoc.unesco.org/images/0014/001418/141843e.pdf.