EduNET: A Prototype Educational System Based on Handwritten and Verbal Interaction for Individuals with Multiple Sclerosis

Chang Hoon Yun, M.S.[1], Zhigang Deng, Ph.D.[1], Olin Johnson, Ph.D.[1] and Oscar Criner, Ph.D.[2]

[1]Department of Computer Science, University of Houston, Houston, TX 77004
[2]Department of Computer Science, Texas Southern University, Houston, TX 77004

ABSTRACT

We introduce a prototype educational system as an intelligent tutorial system for individuals having multiple sclerosis in various ranges and stages of symptoms. The educational system incorporates handwriting, speech recognition and speech synthesis technologies to aid the patients more effectively than currently available educational tools. The architecture of the system combines Tablet PCs and a High Performance Workstation (HPW) to perform as clients and a server. The Tablet PC provides the user interfaces to access the advanced tutorial system operated by the HPW. Our initial experiments showed our system could be used as an effective tool for the individuals with multiple sclerosis.

KEYWORDS

Educational System Software, Handwriting and Speech Interaction, Multiple Sclerosis

BACKGROUND AND PURPOSE

Approximately 400,000 Americans between the ages of 20 and 50 live with multiple sclerosis. The symptoms include simple numbness and tingling, loss of balance, partial or full paralysis, and difficulties in vision, cognition and speech. Usually their stages vary from one patient to another (1). The first symptom usually begins between the ages of 20 and 40 (2), the age group that is still actively engaging in many forms of learning. To accommodate most of them in the area of education, we develop a prototype educational system based on handwritten and verbal interaction. This can aid most of the mild symptom patients to enjoy the benefit of both verbal and handwriting capabilities to access the system. For the patients with hand paralysis or blindness, the system can still be accessed under speech-only mode to assist them. To demonstrate our system, we experimentally choose one application – solving simple arithmetic questions using the individuals' handwriting and/or speech.

METHODS

In our system, one server governs and manages connection initiation, termination and interprocess communication processes with many clients. The server is able to recognize and distinguish one client from another all times. Simultaneously, it also generates the arithmetic questions. Once the server receives the responses from the clients, it can verify the answers and send corresponding messages to the specific clients. We chose the HPW to be the server of the overall system since it has enough computational power to handle all responsibilities described above. The user never interacts with the HPW physically throughout the session. On the other hand, we chose the Tablet PC as the clients since it has the handwriting and integrated speech recognition capabilities. On top of the initial capacities of the Tablet PC, we added speech synthesis capacity so that the system can fully interact with the individuals with speech alone if necessary. Another reason for choosing the Tablet PC instead of the Personal Digital Assistant (PDA) is because of its reasonable screen size to enable the user to read and write on the screen.

We installed the Microsoft .NET Framework 1.1 as a software environment on both the server and the client. .NET Remoting is a part of the .NET Framework that provides interprocess and remote communication under the .NET environment (3). .NET Remoting also handles underlying interprocess communication between the server and the clients. We designed the software using Microsoft Visual Basic .NET 2003. Additional software environments include the Microsoft Speech Software Development Kit (SDK) 5.1 and the Microsoft Tablet PC Platform SDK 1.7. The Speech SDK contains the Speech Application Programming Interface (SAPI) version 5.1 that includes the text-to-speech (TTS) systems and the speech recognizer. They enable our program to have the speech related capabilities (4). The Tablet PC Platform SDK contains the "Digital INK" (INK) handwriting recognition component for developers using Microsoft C# or Visual Basic .NET programming languages. The INK enables the applications to have pen-enabled capabilities including handwriting recognition. We decided to use these software environments since they can be downloaded from Microsoft freely.

Name of the prototype system is called, "EduNET – Educational System", indicating that it is an educational program system (Edu) capable of the interprocess communication (NET) between the clients and the server. The client-side programs are installed into the Tablet PCs so that the user can interact with the handwriting and the speech. Figure 1 is a high-level diagram that describes how each component of the EduNET functions in the server-client model view (refer to SSP_CAC_YunC_pic1.jpg).

To prevent the system from erroneously recognizing the synthesized computer voice as the user's voice, it is required for the user to use headphone. We also ask the user to use the microphone during the session so that the system can recognize clearly the user's speeches.

When the user initiates the client-side EduNET program, a simple logon panel appears. At this point, there is no connection between the server and the client. A greeting message appears on the logon panel. The speech synthesizer synthesizes the greeting message into the speech so that the user can read the text and listen to the synthesized speech simultaneously. It asks the user to provide his or her name. Let's say that the name of the user is 'Alex'. Alex has an option either to speak his name using the microphone or write on the pad of the logon panel using a stylus. If he writes his name on the pad, the handwriting recognizer recognizes his handwriting and translates it into the text (refer to SSP_CAC_YunC_pic2.jpg). On the other hand, if he speaks 'Alex' using the microphone, the speech recognizer recognizes and translates it into the text. Either way, the system recognizes the name of Alex without him typing the name with a keyboard. Once he writes or speaks his name, he has an option either to click the 'OK' button or say 'okay'. The user system is designed to recognize the simple commands, 'OK', 'Clear' and 'Exit'. If he says 'Clear', the pad is cleared as the logon panel asks him to write or say his name again. If he speaks, 'Exit', the system asks whether he really wants to exit or not. If he says 'Yes' or click the 'Yes' button, the system is terminated.

Once the name is provided, the client-side logon panel disappears and the program prompts the server to retrieve the name of the user. The server obtains the name of the user and displays it on its own trace panel. Once the connection between the server and the client is established, the server sends the message to the client concerning the establishment of the connection. At the same time, it generates a simple arithmetic question and sends it to the client along with the connection establishment message. The activities of the server are displayed on the panel (refer to SSP_CAC_YunC_pic3.jpg).

On the client side, when the user system receives the messages from the server, a Study Room panel appears along with a simple message and the first arithmetic question (refer to SSP_CAC_YunC_pic4.jpg). Like the logon panel, the message and the question are displayed as the text and spoken as the speech. And like the login process, he can either answer the question using the speech or the handwriting. If he provides the wrong answer either in his writing or speech, the client-side program prompts the server to retrieve the response. The server, remembering a correct answer for the question, sends the message to the client informing that the response is incorrect and requests Alex to solve the problem again. The Study Room panel on the client-side displays and speaks the message from the server and waits for Alex to provide an answer (refer to SSP_CAC_YunC_pic5.jpg). On the other hand, if he provides right answer, the server generates next question and sends it to the client along with the message that the user correctly answered the question (refer to SSP_CAC_YunC_pic6.jpg). The Study Room panel displays and speaks the message from the server along with the next question and waits for him to provide answer for the new question (refer to SSP_CAC_YunC_pic7.jpg). The connection is maintained until Alex decides to quit. The process to quit the program is same as the process to quit the logon program. Once Alex confirms the desire to quit the program, the client side program sends the server a termination message. The server sends the acknowledgment message to the client and severs the connection with the client (refer to SSP_CAC_YunC_pic8.jpg). Finally, the client-side program receives the message from the server and terminates itself.

RESULTS

From the initiation to the termination, the user was able to interact with the client-side educational system in the Tablet PCs under speech-only, handwriting-only or mix of both speech and handwriting mode. Using one of three modes, the user was also able to receive and solve the arithmetic questions generated from the server. The server, on the other hand, was successful in establishing, terminating and managing the connection and interprocess communication processes along with the generation of the arithmetic questions. We also tested the multiple-clients case and found that the server was successful in distinguishing individual clients as well as providing questions and responding appropriately with each client.

DISCUSSION

While the prototype educational system was successful in studying simple arithmetic level mathematics, the completed version of our educational system will be capable of assisting individuals who study remedial and/or continuing education in science, technology, engineering and mathematics (STEM) as well as non-STEM based job-training courses. To complete the system, more sophisticated handwriting and speech-related engines are necessary so that the system can recognize and understand the users' handwriting and speech with higher accuracy. At the same time, as the next phase of the project, we will add the ability to include questions from advanced mathematics and other subjects.

CONCLUSION

As the initial phase of the project, we developed the prototype educational system based on handwritten and verbal interaction to accommodate the users with various levels of multiple sclerosis. In this phase, we concentrated on developing the interface system to let the users enjoy the benefit of enhanced accessibility. Users with hand paralysis or blindness can use the system with speech alone while the users with lesser symptoms can use the system with both the speech and the handwriting capabilities. Prior to the next phase, we plan to let individuals with multiple sclerosis use the system, and provide feedback to improve the design and functionalities.

REFERENCES

1. National Multiple Sclerosis Society (2006), “Just the Fact 2006-2007/ Frequently asked questions about multiple sclerosis and the National MS Society.” [http://www.nationalmssociety.org/Brochures-Just%20the.asp]
2. Shannon, J. B. (2003), “Movement Disorders Sourcebook.” Omnigraphics, Inc.
3. MacDonald, M. (2003), “Peer-to-Peer with VB .NET”, APress.
4. Microsoft (2002), “Microsoft Speech SDK (SAPI) 5.1” [http://www.microsoft.com/speech/download/sdk5.1]

CONTACT

Chang H. Yun, M.S.
University of Houston
Department of Computer Science
4800 Calhoun Rd,
Houston, TX 77004
Voice: (832) 563-1691
Email: cyun@uh.edu