Voice command adaptation for Jclic, for the special education context

M. LUCRECIA MORALEJO[1,3], STEFANIA OSTERMANN[2], CECILIA V. SANZ[3] AND PESADO PATRICIA[1,3]

[1] III LIDI, School of Computer Science. Universidad Nacional de La Plata, Argentina.
[2] School of Computer Science. Universidad Nacional de La Plata, Argentina.
[3] CIC (Comisión de Investigaciones Científicas), Buenos Aires, Argentina.

Abstract. *In this paper, the adaptation of an educational software application with voice commands for students with motor disability who have no speech impairments is proposed. As part of this process, some educational program and adaptive software applications were analyzed. Integration tests with several adaptive software applications studied with JClic were also carried out to analyze the assistance they can provide students with some type of motor disability in activity solving. Different voice recognition (VR) motors were studied, as well as their theoretical basis. The analysis of the VR motor Sphinx-4 was detailed, studying the design architecture and development of the selected educational tool, JClic. Finally, the development of a prototype with the adaptation of JClic was carried out, with the integration of Sphinx-4 to provide VR, in particular, for simple association activities.*

Keywords. *ICTs, educational software, motor disability, voice recognition, technical aides.*

1. Introduction

Nowadays, there is a large number of software oriented to education on its various levels. Many of these have been accepted or created taking into account student diversity, but others are just standard tools that do not offer any adaptation, which means they are targeted to a restricted set of students [1]. People affected by some kind of motor disability usually have difficulty in some basic skills related to perception (visual, auditory and tactile), communication, movement and/or handling. As a consequence, they face numerous obstacles and barriers that prevent them from developing skills, carrying out activities, relating to other people and the environment, etc. For people with special needs, the mere use of ICTs may represent the achievement of a high degree of autonomy in their personal lives [2].
One of the reasons for the limited deployment of ICTs in special education is the diversity and specificity of needs. Their use as tools, in this field, requires very complex or varied developments, some customized, that are going to be used by not-very-numerous groups.

The current situation presents great challenges to overcome for a disabled person to be in a position of equality with the rest of the population. Therefore, the environment has to be appropriately adapted and technical aides should be used that allow a maximum elimination of the barriers that prevent disabled people to interact with their environment. However, this has been the main conflicting issue: people with motor disabilities usually do not have the technical aides and adaptations they need to interact with a hostile environment [3].
These are the reasons behind the development of adaptations to a software application that is very widely used in the educational context, such as JClic, to facilitate its use by students with motor disabilities and thus encouraging their intellectual development.
Even though there is a known relation between motor disability and speech development difficulties, this is not so in every case. This paper is aimed at people with motor problems, but with little or no consequences in language development. This subset of people was selected because there is a wider variety of technical aides for people with motor disability that use various parts of the body, and we considered that the use of voice would be a good alternative if the person affected by the disability had no difficulty in oral expression. Also, this type of adaptation would require less effort from the person to use the computer, which would help prevent injuries caused by a "repetitive strain".

2. Selection of Jclic as tool to be adapted

In the market, there is a large number of educational software applications available. With the purpose of having a diverse panorama of available applications, an analysis was carried out with those which offer various functionalities and could be used on different educational levels. Among those, JClic, Textoys, Hot Potatoes, Markin, Lim, and Wink were included.
Among these applications that were analyzed, the JClic tool, which is an environment for creating, carrying out and assessing multimedia educational activities, was selected. It is formed by three main components – JClicAuthor, used for creating activities, JClicPlayer, used to solve activities, and JClicReports, used to compile data from solved activities. This application is used to carry out various types of educational activities: puzzles, associations, text exercises, crosswords, etc. [4]. In general, activities are not presented alone, but packed into projects. A project is formed by a set of activities and one or more sequences that indicate the order in which they are to be shown.
Some of the features that led to the selection of this software application were that it is developed under a GPL license, which means that the source code of the program is available for study and analysis. Thus, it was possible to carry out the integration proposed. It is also one of the most widely used software applications for carrying out educational activities (it has been used in the

educational context for years now), so it was considered that it would be interesting to develop a prototype to widen user diversity for these activities.
A second strong reason for this selection is that JClic can be used on various operating systems, such as Windows, Linux, Solaris and Mac OS X. This is because JClic was entirely developed with Java technology, which is multiplatform.

3. Voice recognition

Technological advances have provided human beings new and greater possibilities of developing a fuller lifestyle, but at the same time, this lifestyle continuously demands new and specific knowledge and skills for individuals to be able to take advantage of the possibilities being offered. In the case of people with some type of disability, the progressive complexity of the social media may however have the opposite effect to the desired social progress [5].
Thus, voice recognition is an alternative to communicate with computers, allowing people with motor disabilities who cannot access the standard keyboard and/or mouse to, through speech, perform actions that would not be possible for them without this technology; in other words, the purpose is to convert human speech into actions that the computer can interpret.
This technology is a part of Artificial Intelligence whose purpose is to allow voice communication between human beings and electronic computers, i.e., the process of converting a spoken message into text that allows the user to communicate with the computer. The problem to solve with any VR system is that of achieving the cooperation of a set of data from various knowledge sources (acoustics, phonetics, phonology, lexicography, syntax, semantics and pragmatics) in the presence of inevitable ambiguities, uncertainties and errors to arrive at an acceptable interpretation of the acoustic message received [6]. A voice recognition system is a computational tool that is able to process voice signals issued by human beings and recognize the information they contain and convert it into text or issue orders that act on a process [7]. Various disciplines are involved in its development, such as: physiology, acoustics, signal processing, artificial intelligence, and computer science.
There are some greatly significant components for VR systems: the dictionary, grammar, the acoustic model, and the language model. The dictionary represents the set of words or sounds to be recognized. Unlike a standard dictionary, each input is not necessarily a single word; it can be as long as a sentence or two. The smallest vocabularies may include one or two sounds to be recognized, whereas very large vocabularies may have hundreds of thousands or more. Grammar is defined based on the words that have to be accepted by the application, and can be given through a style that is similar to the BNF.
The language model can be tackled through statistical models (Statistical Model Language - SLM) or by using finite state grammars (FSG) [8]. A statistical model captures word and word sequence probability. It is used in the decoder to limit the search and, in general, makes a significant contribution to recognition accuracy. A good model is that which accurately

models the expected input. It is characterized by its order, in terms of "n-gram", where "n" indicates the size of the window over which statistics are computed. In general, the larger "n", the more accurate the model. Also, as "n" increases, more data are required to ensure a correct estimation of statistics. A finite state grammar defines possible words, as well as their possible orders.
An acoustic model is created from recordings, their respective transcriptions, and the use of software to create statistical representations of the sounds forming each word. The performance of the recognition achieved by the acoustic model can be further improved by means of a language model, which helps avoid the ambiguity among several similar words produced by the acoustic model.
For the selection of the tool to be used, various voice recognition software applications were analyzed, including Loquendo, Xvoice, NicoToolkit, Sphinx, and Dragon Naturally Speaking. Their main features, functionalities and requirements were studied.
Among the applications analyzed, Sphinx in its version 4 was selected. It is a system developed at Carnegie Mellon University (CMU) [9]. This framework is a system based on hidden Markov models (HMM) so, as a first step for its operation, it must first learn the characteristics (or parameters) of a set of sound units, and then use the knowledge acquired from these units to find the most likely sequence of sound units for a given voice signal. This particular tool was selected because it is widely used by researchers and developers working in the area of voice recognition and, therefore, it is constantly developed and updated.
Due to its licensing characteristics, it can be freely used for any development and research activities. Also, its source code can be obtained, in case any modification were required or its low level operation were to be studied. It is completely developed with Java technology, the same as JClic. Thus, it served the purpose of integrating both components without the problems of language incompatibility. Additionally, it has been designed with a high degree of flexibility and modularity, where each system element can be easily replaced or modified. It is through the Configuration Manager that the framework provides the possibility of dynamically loading and configuring the various modules, during runtime. Thus, the components that are going to be used, and their specific configuration, are determined. In particular, the dictionary and grammar to be used during recognition can be specified. Below, the specific proposal for this work is presented.

4. Adaptation proposal

The adaptation proposed has tackled the modification of JClic activities so that they can be solved through the use of voice commands. To this end, the simple association type of activity was initially considered.
In this type of activity that can be created in JClic, the user has to discover the relations that exist between two sets of information. That is, two groups

of data with the same number of elements are presented, where each element in the source data set corresponds to an element in the target set. This one-to-one relation is what makes this a simple association, in contrast with complex associations, where each source element may correspond to 0, 1, or more target elements.
As a first step to carry out this integration, some decisions had to be made, as detailed below.

4.1. Stage 1: Analysis

One of the decisions considered was how to inform that the activity will be done using voice commands.
It was considered that in this situation, the user needs the assistance of the teacher, since the latter is in charge of deciding if the use of VR is appropriate for each specific student. To do this, the program shows a prompt on the screen before starting the activity. It asks the user to indicate if voice recognition will be used.
Another important issue was deciding on the mechanism that should be provided to identify each interactive element on the screen, with the purpose of solving the activity. In this regard, various possibilities were analyzed. This identification used by the user to name an element will be called label from here on.
First, the possibility of using the letters of the alphabet as labels was considered, but this turned out to be impracticable due to the phonetic similarity between certain letters in Spanish, such as “b” and “d,” which considerably decreased recognition success rates.
On the other hand, if the number of checkboxes to be used increased, it was more natural to use combinations of digits (e.g., 10) than using combinations of letters (e.g., ab). Also, not all letters could be used; those that caused phonetic conflicts, such as the ones already mentioned, or those whose pronunciation was complex, such as the case of letter "r," had to be removed from the dictionary. This considered, the decision was made to use numbers for the creation of labels. This solution presents certain advantages in relation to the first option proposed.
Additionally, it was decided to include all necessary adaptations to avoid difficulties in the pronunciation of certain numbers. To do this, alternative pronunciations to the correct word were considered. For instance, users can say “tes” instead of “tres,” “tinco” instead of “cinco,” “acetar” instead of “aceptar,” among others.
Even though this decision results in a larger dictionary, it has positive consequences as regards the number of users for whom the prototype would be accessible. Thus, a balance between application performance and product usability was attempted.
The second issue that had to be solved was that of knowing when the user finishes naming the two elements to be joined. To do that, the use of “connecting” words was considered. For instance, “uno con tres aceptar” (one with three accept), which is interpreted as follows: the first number

("uno," one) represents a checkbox from the first set of data, the connecting word “con” (with) indicates that the user is about to name the checkbox from the second set, represented by the second number in the phrase ("cinco," five). The word “aceptar” (accept) confirms that the user wants to join both checkboxes.
Also, as regards labels, a decision had to be made on where to add the necessary code for the label to be inserted in the component representing the checkbox with the information. It should be mentioned that they are generated when JClicPlayer is run but only if the user indicates that voice commands will be used to carry out the activity. This involved a decision, since the presentation of the information from both sets had to remain random, so that the activity did not appear already solved because of the use of labels.
Finally, the necessary code was added in such a manner that the application shows a message prompting for confirmation of what the user said. Thus, when the user names the checkboxes to be joined, the program presents a message with the words that were recognized. To confirm the recognition, the user says “aceptar” (accept); otherwise, the word used is “cancelar” (cancel).
Below, the second stage of the work is presented, which was deciding on issues related to the VR motor and implementing those decisions.

4.2. Stage 2: Configuration of Sphinx-4

First, in order to use Sphinx, the application has to be downloaded from the official site [5]. The site also offers the source code of the tool for those who wish to make changes. If, however, as in our case, no modifications are to be introduced, all that has to be done is including the .jar file in the application where it will be integrated.
Currently, Sphinx-4 has models that have been created with SphinxTrain (training tool included), and it can be downloaded from the cmusphinx.org site.
At first, the idea of creating the dictionary using the WSJ_8gau_13dCep _16k_40mel_130Hz_6800Hz model that is included with the distribution of Sphinx-4 was considered as a valid alternative, by replacing English phonemes for those corresponding to Spanish. There are reviewed works in the VR area that perform this type of solution[1].
Even though phonemes correspond to the English language, during a first stage they were used to generate the dictionary for the integration with JClic.
This solution was partially valid, since the recognizer worked with a high success rate. However, two shortcomings were found. On the one hand, there were errors in recognizer accuracy in noisy environments. This would be a problem in those cases when the adaptations were to be used in schools, where there are several students in the same classroom. On the other hand, if

[1] Among these, the Mouse Advanced GNU Speech (Magnus) project was consulted: http://magnusproject.wordpress.com/

the dictionary were to be extended to use words with the letter "ñ," there were no phonemes in English that could represent the corresponding sound.
Based on these conclusions, it was decided to switch to a model based on the Spanish language. After some research on the topic, two viable alternatives were found. One of the options was training the recognizer with the SphinxTrain tool, while the other was using models that had already been trained and tested. For this development, an already trained model was selected, but some tests were also done with the training tool in order to understand and study its operation.
To do this, an already trained model that was available on the Web and whose use was free, was used. The project is called Diálogos Inteligentes Multimodales en Español (DIME, Intelligent Multimodal Dialogues in Spanish), and it offers more than one acoustic model. The model selected for this work is called DIMEx30-T22 [10].
Using this list of phonetic units, the dictionary to be used in the integration with JClic was created. It would have been possible to incorporate the dictionary exactly as presented by DIMEx30, but there were some words that were missing, so it was redefined using the same original phonetic units. As regards the language model, the definition of the acoustic model and its architecture, they were used exactly as offered by DIMEx30.
To incorporate these files to JClic, first a .jar file had to be created which, by standard, should follow the directory structure of the models provided by Sphinx-4.
After creating the .jar file, it was included in the classpath of the application.
Also, Sphinx-4 had to be configured to incorporate the new acoustic model files, dictionary, grammar, and language model. This was done through the configuration file (Configuration Manager). In the following section, aspects related to the development of the prototype are detailed.

4.3. Stage 3: Prototype development

In this section, aspects of the prototype including both components used for the integration will be discussed. One of these is the procedure followed to incorporate the voice recognition framework into JClic. To do it, a class representing the recognizer was created in JClic, called VoiceRecognizer, where the main methods are included, such as the method used for its creation, as well as the method that is responsible for carrying out the recognition itself. A package called "recognition" was created within the "src" package of JClic. Then, this class is used in the builder method of the Player class if the user chooses to work using voice recognition. This is where the recognizer is created to start working.
Also, the class representing the recognizer was configured to inherit from SwingWorker, even if Swing is not used, so that JClic and the recognizer run on separate threads that interact with each other, so as to parallelize tasks. Thus, both components can be executed seamlessly.
To carry out the task of solving a simple association activity, upon its creation, the recognizer runs a method called getCommand() in the class

representing the corresponding activity. This method is responsible for processing the voice input from the user and making the corresponding decisions.
When an “aceptar” (accept) voice input is received, the system shows a message with the values that are going to be processed; the user has to confirm the values for the action to be performed.
For the confirmation, the word “aceptar” has to be uttered again. After confirmation, a method is invoked that is responsible for executing the action that the user wishes to perform. This method looks for the checkboxes that were named, if they exist and have not already been selected. Then, it checks within the internal structure of the elements if the correspondence is correct, i.e., if the cells selected are part of the solution. If so, the elements are removed from the set of possible elements to be chosen, and it moves on to the next correspondence, until getting to the last one. When the last correspondence is checked, the activity finishes.
JClic provides a module that can keep track of the time used in each activity, attempts, correct answers, etc. Even though time can vary if voice recognition is used, the counters for attempts and correct answers were kept unchanged to allow the teacher to evaluate student performance for the activity. For this reason, a message prompting the user to confirm the answer was added, since most recognizers introduce a certain error rate. This means that the recognizer could interpret a wrong answer and JClic would record it as a failed attempt on the part of the student, harming student performance evaluation. With these additions, the teacher can use the error counter provided by default by JClic.
The prototype developed so far includes, as already mentioned, the resolution of simple association activities. However, as part of this work, a proposal has been made for a possible strategy to extend the prototype to the remaining activities. This will be tackled in future works. In the following section, the integration strategies proposed so far are assessed.

5. Assessment and conclusion

The prototype described in this article was provided to experts in the field (working in the various areas involved in this work) for them to express their opinions.
This type of test was carried out first in order to analyze the results and take them into account for future lines of research work. After this stage, the prototype will be tested with end users, including both teachers and students. The reason to carry out this testing with the end users as a second stage was to avoid having students experience possible failure situations, typical of the software testing and strategy itself stages. Also, this methodology offers the advantages of producing quality feedback and a thorough analysis by the experts. Experts offer their thoughts about the object to be assessed. Through this expert opinion, it is expected to obtain reasonably good assessments and guesses in situations where no exact quantifications can be obtained, or doing

so is not advisable [11]. However, these assessments can, and should, be confirmed or modified in time, as information on the study object is collected.
A survey with close-ended and open-ended questions was used as assessment instrument to collect the information needed to asses the prototype.
The following results were obtained as a conclusion of the surveys answered by the experts: The selection of the educational software to be adapted was good, as well as the use of voice commands as technical aid. As one of the experts mentioned, this option can be used as a complement with other tools and is not necessarily better or worse than any other adaptation, but a different alternative that opens a road of new possibilities. Even though only a few experts commented on the selection of the voice recognition motor, those who did agreed that it was correct. The main aspects to be highlighted are its availability and possibilities as regard functionality. In the context of this work, it is considered that the use of Sphinx-4 was convenient, in agreement with the feedback received from the experts.
Finally, the solution strategy proposed was analyzed, and the experts expressed their agreement and offered some alternatives to take into account in future works. Some of these aspects are detailed in the following section.

6. Future lines of work

Even though a significant amount of knowledge has been collected on various tools, both educational and in relation to VR, there are still certain modifications, improvements and extensions to be developed in the adaptation presented, considering as well some suggestions analyzed after the assessment carried out.
The following future lines of work are proposed:

- Carrying out tests with students and teachers.
- Allowing label configuration (the teacher could choose to label each checkbox as desired)

One of the improvements that has already been implemented is how to analyze if voice commands will be used to solve the activity. Before, every time a project was loaded from JClicPlayer, the user was asked if the activity would be solved by means of voice commands. Since JClic is used beyond the context of special education, this prompt was oftentimes unnecessary, since no student would be using voice commands. In the cases of special education, it will be a decision made for each student. For this reason, this decision was initially the responsibility of the teacher. The teacher decides if, when the project is loaded, the prompt to use voice commands is shown or not.

Currently, the prototype is being extended to the remaining activities in JClic. The next type of activities to be added will be those of complex association and memory games.

References

1. Sánchez Montoya, R. (2007). "Capacidades visibles, tecnologías invisibles. Perspectivas y estudios de casos". Seminario Internacional Virtual: "Las nuevas tecnologías de la información y la comunicación aplicadas a las necesidades educativas especiales". Perú. http://www.ordenadorydiscapacidad.net/Capacidades.pdf
2. Castellano, Sacco, Zurueta (2003). "La utilización de software de uso general y aplicaciones específicas en el área de las discapacidades motrices". IV Congreso Iberoamericano de Informática en la Educación Especial. http://www.niee.ufrgs.br/eventos/CIIEE/2003/
3. Perez, F.J. y Rodríguez Vázquez, J. (2004). "Tecnología. Educación y diversidad: retos y realidades de la inclusión digital. Propuestas de futuro". 3º Congreso Nacional de Tecnología. Educación y Diversidad. Conclusiones. Biblioteca TECNONEET. http://www.tecnoneet.org/conclu04.php.
4. http://clic.xtec.cat/es/jclic/index.htm.
5. http://www.tecnoneet.org/docs/2002/2-82002.pdf.
6. Bernal Bermúdez, Bobadilla Sancho y Gómez Vilda (2000). "Reconocimiento de voz y fonética acústica". México, Alfaomega grupo Editor.
7. Rocha, Luis (1986). "Sistemas de reconocimiento de voz". Revista telegráfica electrónica. Agosto, 1172-1180.
8. http://sphinx.subwiki.com/sphinx/index.php/Language_model.
9. http://cmusphinx.sourceforge.net/sphinx4/
10. http://leibniz.iimas.unam.mx/~luis/DIME/recursos.html.
11. http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/401a500/ntp_401.pdf.