

Adaptación a Jclic mediante comandos por voz, para el ámbito de la educación especial

M. Lucrecia Moralejo¹, Stefania Ostermann², Cecilia V. Sanz³, Pesado Patricia³

¹Becaria CIC (Comisión de Investigación Científica) - II LIDI, Facultad de Informática. Universidad de La Plata, La Plata, Buenos Aires, Argentina. Imoralejo@lidi.info.unlp.edu.ar.

²Facultad de Informática. Universidad de La Plata, La Plata, Buenos Aires, Argentina. stefaniaostermann@gmail.com

³III LIDI, Facultad de Informática. Universidad de La Plata, La Plata, Buenos Aires, Argentina. {csanz, ppesado}@lidi.info.unlp.edu.ar.

Abstract. En el presente trabajo se propone la adaptación de un software educativo mediante comandos por voz, para alumnos con deficiencia motriz y que no se han visto afectados en el habla. Durante este proceso se analizaron algunos programas educativos y software adaptativos. Además se realizaron pruebas de integración de varios software adaptativos estudiados con Jclic, para analizar la ayuda que pueden ofrecer a los alumnos con algún tipo de deficiencia motriz en la resolución de actividades. Se estudiaron diferentes motores de reconocimiento de voz (RV), y sus fundamentos teóricos. Se profundizó el análisis del motor de RV Sphinx-4, estudiando la arquitectura de diseño y desarrollo de la herramienta educativa elegida, Jclic. Finalmente, se obtuvo el desarrollo de un prototipo de adaptación del programa Jclic, con la integración de Sphinx-4 para proveer RV, en particular, para las actividades de asociación simple.

Keywords: TIC, software educativo, deficiencia motriz, reconocimiento por voz, ayudas técnicas.

1 Introducción

Actualmente, existe una gran cantidad de software orientado a la educación en sus distintos niveles. Muchos de ellos han sido adaptados o creados teniendo en cuenta la diversidad de alumnos, pero otros son sólo herramientas estándares que no brindan adaptación alguna, por lo que están destinados a un conjunto restringido de alumnos. Las personas que están afectadas de algún tipo de disminución motriz, suelen evidenciar dificultades en algunas habilidades básicas relacionadas con la percepción (visual, auditiva y táctil), la comunicación, el desplazamiento y/o la manipulación. En consecuencia, se encuentran con numerosos obstáculos y barreras que les impiden el

desarrollo de habilidades, la ejecución de actividades, la relación con las personas y el entorno, etc. Para las personas con necesidades especiales, la mera utilización de las TIC puede representar la consecución de un elevado grado de autonomía en su vida personal.

Una de las razones de la escasa implantación de las TIC en la educación especial es la diversidad y la especificidad de las necesidades. Su utilización como herramientas, en este campo, requiere desarrollos muy complejos y variados, algunos personalizados, que además van a ser utilizados por colectivos poco numerosos.

La situación actual presenta grandes retos a superar para que una persona con disminución pueda estar en una posición de igualdad de condiciones, respecto de las demás personas. Por lo tanto, se debe adaptar adecuadamente el entorno y utilizar ayudas técnicas, que permitan eliminar al máximo de barreras, que dificultan que una persona con discapacidad pueda interactuar en él. Y sin embargo, éste ha sido el punto principal de conflicto: las personas con disminución motriz no disponen con frecuencia de las ayudas técnicas y adaptaciones necesarias para interactuar en un entorno hostil.

Estas razones motivan el desarrollo de adaptaciones a un software muy utilizado en el ámbito educativo como es JClic, para facilitar su uso por parte de alumnos con dificultades motoras, y así incentivar su desarrollo intelectual.

Si bien se conoce la relación entre la discapacidad motora y la dificultad en el desarrollo del lenguaje, esto no se da en todos los casos. Este trabajo está destinado a las personas con problemas motores, pero sin consecuencias o con consecuencias leves en el desarrollo del lenguaje. Se pensó en este subconjunto de personas, ya que existen más variedad de ayudas técnicas para personas con discapacidad motriz mediante la utilización de diferentes partes del cuerpo y se considera que sería una buena alternativa, el uso de la voz, si la persona afectada se expresa oralmente sin dificultades. Además, requeriría un menor esfuerzo si la persona pudiera usar la voz para manipular el ordenador y se evitarían las lesiones producidas por “esfuerzo repetitivo”.

2 Elección de Jclic como herramienta a adaptar

Existen en el mercado, una gran cantidad de software educativo. Con el objetivo de tener un panorama variado de los programas disponibles, se analizaron algunos de ellos, los cuales ofrecen diferentes funcionalidades y además se podrían utilizar en diversos niveles educativos. Entre ellos, se encuentran JClic, Textos, Hot Potatoes, Markin, Lim, y Wink.

De estos programas analizados, se optó por adaptar la herramienta JClic, que es un entorno para la creación, realización y evaluación de actividades educativas multimedia. Está compuesto por tres componentes fundamentales, JClicAuthor que se utiliza para la creación de las actividades, JClicPlayer para la resolución de las mismas y JClicReports para recopilar datos de las actividades resueltas. Esta aplicación sirve para realizar diversos tipos de actividades educativas: rompecabezas, asociaciones, ejercicios de texto, palabras cruzadas, etc. [1]. Las actividades, por lo

general, no se presentan solas, sino empaquetadas en proyectos. Un proyecto está formado por un conjunto de actividades y una o más secuencias, que indican el orden en que se han de mostrar.

Algunas de las características que llevaron a la elección de este software, fueron que está desarrollado bajo licencia GPL, lo que provee la oportunidad de contar con el código fuente del programa para poder estudiarlo y analizarlo. De esta manera, se pudo llevar a cabo la integración propuesta. También es uno de los software más utilizados para realizar actividades educativas (ya lleva años de uso en el ámbito), de manera que se creyó que sería interesante comenzar el desarrollo de un prototipo para ampliar la diversidad de usuarios de dichas actividades.

Otro punto fuerte en esta elección es que es posible utilizar JClic en distintos sistemas operativos, tales como Windows,

Linux, Solaris y Mac OS X. Esta cualidad viene dada porque JClic se encuentra enteramente desarrollado con tecnología Java, la cual es multiplataforma.

3 Reconocimiento de voz

El avance tecnológico ha aportado al ser humano nuevas y mayores posibilidades de desarrollar un modo de vida más completo, pero al mismo tiempo exige continuamente nuevos y específicos conocimientos y habilidades en el individuo para poder hacer uso de las posibilidades que le ofrecen. En las personas con algún tipo de discapacidad, la progresiva complejidad del medio social puede tener, sin embargo el efecto contrario al buscado por el progreso social [2].

Así se encuentra en el reconocimiento de voz una alternativa para la comunicación con la computadora, permitiendo que las personas con discapacidades motoras que no pueden acceder al teclado estándar y/o al mouse puedan, con el habla, realizar acciones que sin esta tecnología no le serían posibles, en otras palabras, el objetivo es convertir el habla humana en acciones interpretables por la computadora.

Esta tecnología, es una parte de la Inteligencia Artificial, que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas, es decir, es el proceso de conversión de un mensaje hablado a texto, que permite al usuario una comunicación con la computadora. El problema que se plantea en un sistema de RV es el de hacer cooperar un conjunto de informaciones que provienen de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades, incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido [3]. Un sistema de reconocimiento de voz es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano, y reconocer la información contenida en ésta, convirtiéndola en texto o emitiendo órdenes que actúan sobre un proceso. En su desarrollo intervienen diversas disciplinas, tales como: la fisiología, la acústica, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación.

Existen algunos componentes de gran importancia para los sistemas de RV, que son: el diccionario, la gramática, el modelo acústico y el modelo de lenguaje. Donde

el diccionario representa el conjunto de palabras o sonidos a reconocer. A diferencia de un diccionario normal, cada entrada no tiene por qué ser una única palabra. Estas pueden ser tan largas como una oración o dos. Los vocabularios pequeños pueden tener una o dos sonidos a reconocer, mientras que los vocabularios muy grandes pueden tener cientos de miles o más. La gramática se define a partir de las palabras que debe aceptar la aplicación, y puede estar dada a través de un estilo similar a BNF.

El modelo de lenguaje puede ser abordado a través de modelos estadísticos (Statistical Model Language - SLM) o utilizando gramáticas de estado finito (Finite Grammar State - FGS) [4]. Un modelo estadístico captura la probabilidad de las palabras y de las secuencias de palabras. Es utilizado en el decodificador para limitar la búsqueda, y generalmente, hace una contribución significativa a la exactitud del reconocimiento. Un buen modelo es aquel que modela con precisión la entrada esperada. Se caracteriza por su orden, en términos de “n-gram”, donde “n” indica el tamaño de la ventana sobre la cual se computan las estadísticas. En general cuanto más grande sea “n” más preciso será el modelo. También, a mayor “n” se necesitan más datos para asegurar que las estadísticas se estiman sólidamente. Una gramática de estados finitos define las posibles palabras, así como también el posible orden de dichas palabras.

Un modelo acústico se crea a partir de grabaciones, sus respectivas transcripciones, y el uso de software para crear representaciones estadísticas de los sonidos que componen cada palabra. La performance del reconocimiento producida por el modelo acústico puede mejorarse aún más, mediante un modelo de lenguaje, el cual contribuye a evitar ambigüedad entre varias palabras similares producidas por el modelo acústico.

Para la elección de la herramienta a utilizar, se analizaron distintos softwares de reconocimiento de voz, entre ellos Loquendo, Xvoice, NicoTollkit, Sphinx y Dragon Naturally Speaking. Se estudiaron sus principales características, funcionalidades y requisitos.

De los analizados, se optó por Sphinx, particularmente la versión 4. Es un sistema desarrollado en la Universidad de Carnegie Mellon (CMU). Este framework es un sistema basado en los Modelos Ocultos de Markov (HMM, Hidden Markov Model), por lo que, para su funcionamiento primero debe aprender las características (o parámetros) de un conjunto de unidades de sonido, y luego utilizar lo que ha aprendido de estas unidades para encontrar la secuencia de unidades de sonido más probable para una señal de voz dada. Se optó por esta herramienta, ya que es ampliamente utilizada por investigadores y desarrolladores que se dedican al área del reconocimiento de voz, y en consecuencia, se encuentra en constante desarrollo y actualización.

Por las características de su licencia, es posible utilizarlo libremente en cualquier desarrollo e investigación. Además, también se puede obtener su código fuente, en caso de que sea necesaria alguna modificación o estudiar su funcionamiento a bajo nivel. Está completamente desarrollado con tecnología Java, al igual que JClc. De manera que servía al propósito de integrar ambos componentes sin dificultades provocadas por incompatibilidad de lenguajes. Además, ha sido diseñado con un alto grado de flexibilidad y modularidad, donde cada elemento del sistema puede ser fácilmente reemplazado o modificado. A través del Configuration Manager, es que el framework brinda la posibilidad de cargar y configurar los distintos módulos

dinámicamente, en tiempo de ejecución. Determinando así, qué componentes van a ser usados y la configuración particular de cada uno de ellos. En particular, es posible indicar el diccionario y la gramática que se utilizaran durante el reconocimiento. A continuación, se presenta la propuesta particular para este trabajo.

4 Propuesta de adaptación

La adaptación propuesta ha abordado la modificación de las actividades de JClic de manera que se puedan resolver a través de la utilización de comandos por voz. Para ello se tomó, inicialmente, la actividad del tipo asociación simple.

En este caso de actividad, que JClic permite crear, el usuario tiene que descubrir las relaciones existentes entre dos conjuntos de información. Es decir, se presentan dos grupos de datos que tienen el mismo número de elementos, donde a cada elemento del origen le corresponde un elemento del destino. Es por ello que se la denomina simple, a diferencia de la asociación compleja, donde a cada elemento del origen puede corresponderle 0, 1, o más elementos del destino.

Como primera medida para llevar a cabo esta integración se debieron tomar algunas decisiones, las cuales se detallan a continuación.

4.1 Etapa 1: Análisis

Una de las decisiones que se consideró fue, cómo tomar conocimiento de que se desea realizar la actividad utilizando comandos por voz.

Se consideró que en esta situación, el usuario deba contar con la asistencia del docente, ya que es éste quien toma la decisión para cada alumno en particular, si es adecuado o no utilizar RV en la resolución. El programa, para ello, muestra un mensaje en pantalla al momento de comenzar la actividad. Este pide que se indique si se desea utilizar reconocimiento de voz.

Otra cuestión de suma importancia, ha sido decidir qué mecanismo proveer para identificar cada elemento de la pantalla que presente interactividad, con el fin de resolver la actividad. Para esto se analizaron diferentes posibilidades. Esta identificación que utiliza el usuario para nombrar un elemento se la denominará etiqueta, de aquí en más.

En primer instancia, se pensó en utilizar las letras del alfabeto como etiquetas, pero al momento de llevarlo a la práctica, se encontró la dificultad de que ciertas letras, tales como la “b” y la “d”, eran muy similares en su pronunciación, por lo que la tasa de aciertos del reconocedor disminuía considerablemente.

Por otro lado, si se ampliaba el número de casilleros a utilizar, resultaba más natural usar combinaciones de dígitos (por ejemplo 10) que utilizar letras (por ejemplo ab). También, se debían utilizar letras alternadas, quitando del diccionario del RV, aquellas que causaban conflictos como los ya mencionados o aquellas que resultaban muy complejas en cuanto a su pronunciación (por ejemplo, el caso de la letra ‘r’). Considerado esto, se decidió la posibilidad de utilizar números para la creación de las etiquetas. Esta solución presenta ciertas ventajas, respecto a la planteada anteriormente.

Además, se decidió realizar las adaptaciones necesarias para evitar dificultades de pronunciación de ciertos números. Para esto, se tuvieron en cuenta otras palabras alternativas a la correcta, por ejemplo, se admite que el usuario diga “tes” en lugar de “tres”, “tinco” en lugar de “cinco”, “acetar” en lugar de “aceptar”, entre otras.

Si bien esta decisión implica un diccionario de mayor tamaño, presenta consecuencias positivas en cuanto al aumento de usuarios que podrían utilizar el prototipo. Así, se intentó lograr un equilibrio entre performance de la aplicación y usabilidad del producto.

El segundo tema a resolver fue el de conocer cuándo el usuario termina de nombrar los dos elementos a unir. Para ello, se pensó en utilizar palabras “nexo”. Por ejemplo, “uno con tres aceptar”; lo que se interpreta de esta sentencia es lo siguiente: el primer número representa un casillero del primer conjunto de información, la palabra “con” (nexo) indica que se va a nombrar el casillero del segundo conjunto, representado por el segundo número de la frase. La palabra “aceptar” indica que el usuario quiere realizar la unión de los casilleros nombrados.

También, con respecto a las etiquetas, se debió pensar en qué momento agregar el código necesario para que se inserte la etiqueta en el componente que representa al casillero con la información. Cabe mencionar, que se generan cuando se ejecuta JClickPlayer, sólo si se indica que se desea realizar la actividad con comandos por voz. Esto implicó una decisión, ya que había que mantener la presentación de la información de los dos conjuntos en forma aleatoria, de manera que no apareciera la actividad resuelta, a causa de las etiquetas.

Finalmente, se agregó el código necesario de manera que la aplicación muestre un mensaje pidiendo confirmación de lo dicho por el usuario. Así, cuando éste nombra los casilleros que desea unir, el programa presenta un mensaje mostrando las palabras reconocidas. Para dar confirmación positiva al mensaje, se debe decir “aceptar”, y en caso contrario, “cancelar”. A continuación, se presenta la segunda etapa de trabajo, que ha sido decidir (y llevar a la práctica) cuestiones vinculadas al motor de RV.

4.2 Etapa 2: Configuración de Sphinx-4

En primer lugar, para utilizar Sphinx, se debe descargar la aplicación desde el sitio oficial [5]. Allí está disponible el código fuente de la herramienta, aunque si no se desea modificar código (como en nuestro caso), alcanza con incluir el archivo .jar en la aplicación donde se va a integrar.

Actualmente, Sphinx-4 dispone de modelos que han sido creados utilizando SphinxTrain (herramienta que provee para el entrenamiento), y puede descargarse desde el sitio de cmusphinx.org.

En un principio, se pensó como una alternativa válida crear el diccionario utilizando el modelo WSJ_8gau_13dCep_16k_40mel_130Hz_6800Hz que viene incluido con la distribución de Sphinx-4 y, si bien está entrenado para el idioma en inglés, reemplazando los fonemas puede reconocer español. Existen trabajos revisados del área de RV, que realizan este tipo de solución¹.

¹Entre ellos, se consultó el proyecto Mouse Advanced GNU Speech (Magnus): <http://magnusproject.wordpress.com/>

Si bien los fonemas pertenecen al idioma inglés, en un primer momento, fueron utilizados para generar el diccionario para la integración con JClic.

Esta solución fue parcialmente válida, ya que el reconocedor funcionaba con un alto porcentaje de acierto. Pero, a pesar de esto, se encontraron dos falencias. Por un lado, había errores en la precisión del reconocedor en ambientes ruidosos. Esto sería un problema en los casos en que la adaptación fuera utilizada en escuelas, donde las aulas se comparten entre varios alumnos. Por otro lado, si se deseaba extender el diccionario y utilizar palabras con la letra “ñ”, no existían fonemas en el idioma inglés que lo represente.

A partir de estas conclusiones, se decidió utilizar un modelo basado en el idioma español. Luego de investigar sobre el tema, surgieron dos alternativas viables. Por un lado, se podía entrenar el reconocedor, utilizando la herramienta SphinxTrain, y por otro, utilizar modelos ya entrenados y testeados. En el presente desarrollo se optó por adoptar un modelo ya entrenado, pero se hicieron además algunas pruebas con el entrenador, de manera tal, de entender y estudiar su funcionamiento.

Para esto se utilizó un modelo ya entrenado, disponible en la web y de libre uso. El proyecto se llama Diálogos Inteligentes Multimodales en Español (DIME), dentro del cual hay más de un modelo acústico. El modelo elegido para este trabajo recibe el nombre de DIMEx30-T22 [6].

A partir de esta lista de unidades fonéticas se creó el diccionario a utilizar en la integración con JClic. Podría haberse incorporado el diccionario, tal cual lo presenta DIMEx30, pero había palabras que no se encontraban en él, por lo que se optó por redefinirlo, respetando las unidades fonéticas presentadas. Respecto del modelo de lenguaje, la definición del modelo acústico y su arquitectura, se respetó el proporcionado por DIMEx30.

Para incorporar estos archivos a la aplicación JClic, se debió crear en primer lugar un archivo .jar que, por convención, debía respetar la estructura de directorios de los modelos provistos por Sphinx-4.

Luego de armado el archivo .jar, se incluyó en el classpath de la aplicación.

También, se debió configurar Sphinx-4 para incorporar los nuevos archivos del modelo acústico, el diccionario, la gramática y el modelo de lenguaje. Esto se realizó a través del archivo de configuración (Configuration Manager). En la siguiente sección, se detallan cuestiones referidas al desarrollo del prototipo.

4.3 Etapa 3: Desarrollo del prototipo

En esta sección se describirán aspectos del prototipo que incluyen a ambos componentes utilizados para la integración. Una de ellas, es cómo se realizó la incorporación del framework de reconocimiento de voz a JClic. Para ello, se creó una clase en JClic que representa al reconocedor, llamada VoiceRecognizer, donde se encuentran sus principales métodos, tales como el método que se usó para crearlo, así como también, el método que se encarga de realizar el reconocimiento. Se generó un paquete llamado “reconocimiento” dentro del paquete “src” de JClic. Luego, esta clase, es utilizada en el método constructor de la clase Player, si el usuario eligió trabajar con reconocimiento de voz. Allí se crea el reconocedor para empezar a trabajar.

También, se hizo que la clase que representa al reconocedor heredara de SwingWorker aunque no utilice Swing, de manera que JClic y el reconocedor se

ejecuten en hilos separados, interactuando entre ellos, para paralelizar tareas. De esta manera, ambos componentes, pueden ejecutarse sin problemas.

Para llevar a cabo la tarea de resolver una actividad de tipo Asociación Simple, lo que se implementó fue, que al crearse, el reconocedor ejecute un método llamado `getCommand()` en la clase que representa dicha actividad. Este método es el encargado de procesar la entrada de voz del usuario y tomar las decisiones correspondientes.

Al recibir la entrada de voz “aceptar”, el sistema muestra un cartel con los valores que se van a procesar, el usuario deberá confirmar estos valores para que la acción se lleve a cabo.

Para la confirmación es necesario pronunciar nuevamente la palabra “aceptar”. Luego de confirmado, se invoca a un método que se encarga de ejecutar la acción que el usuario desea realizar. En este método se buscan los casilleros nombrados, si existen y no fueron elegidos antes. Luego, se verifica dentro de la estructura interna del elemento, si forman una correspondencia correcta, es decir, si las celdas seleccionadas son parte de la solución. Si es así, se eliminan de los posibles elementos a elegir y se continúa con la próxima correspondencia, hasta llegar a la última. Cuando se llega a ésta, se da por terminada la actividad.

JClic provee un módulo capaz de contabilizar el tiempo empleado en cada actividad, intentos, aciertos, etc. Si bien el tiempo puede variar si se utiliza reconocimiento de voz, se pensó en mantener igualdad en el contador de intentos y aciertos para que el docente pueda evaluar al alumno que está resolviendo la actividad. Es por esta razón, que se decidió agregar un cartel donde el usuario vea y confirme que es lo que desea unir, ya que existe, en la mayoría de los reconocedores, cierta tasa de error, con lo cual, podría darse la situación en que se procese una entrada errónea y JClic lo contara como intento fallido, perjudicando la evaluación del alumno. Con los agregados mencionados, el docente que creó la actividad podrá usar el contador de errores que provee, por defecto, JClic.

El prototipo desarrollado al momento abarca, como se dijo, la resolución de las actividades de asociación simple. Sin embargo, como parte de este trabajo, se ha hecho una propuesta de cuál sería la estrategia para extender el prototipo al resto de las actividades y se abordará esto en trabajos futuros. A continuación se presenta la evaluación realizada de las estrategias de integración planteadas al momento.

5 Evaluación y conclusión

Se decidió someter el prototipo presentado a juicio de expertos (vinculados a las distintas áreas que se involucran en este trabajo) para que ellos expresaran sus opiniones respecto de este proyecto.

Se consideró más apropiado realizar primero este tipo de prueba, y analizar los resultados para tomarlos como líneas futuras de trabajo e investigación. Después de esta etapa será posible testear el prototipo con los usuarios finales, los cuales serían, docentes y alumnos. Esto se creyó importante, para no someter a los alumnos a situaciones de posibles fracasos propios del testeado del software y de la estrategia en sí

misma. Además, esta metodología tiene como ventaja la calidad de la respuesta y el nivel de profundización por parte del experto. Este puede ofrecer sus reflexiones acerca del objeto a evaluar. Mediante el juicio de expertos, se pretende tener estimaciones razonablemente buenas, las mejores conjeturas, en situaciones donde no se pueden o no es conveniente obtener cuantificaciones exactas [7]. Sin embargo, estas estimaciones pueden y deben ser confirmadas o modificadas a lo largo del tiempo, según se vaya recopilando información sobre el objeto de estudio.

Como instrumento de evaluación se eligió una encuesta con preguntas abiertas y cerradas, de manera tal de poder recoger la información que se cree necesaria para someter a juicio el prototipo.

Como conclusión de las encuestas realizadas a expertos, se obtuvieron los siguientes resultados: se ha realizado una buena elección del software educativo a adaptar, como así también, la utilización de comandos por voz como ayuda técnica. Como mencionó uno de los expertos, esta opción puede usarse de forma complementaria con otras herramientas y no necesariamente es mejor o peor que otra adaptación, sino que es una alternativa diferente, la cual abre un camino de nuevas posibilidades. Si bien unos pocos expertos se manifestaron acerca de la elección del motor de reconocimiento de voz, coincidieron en que la misma es acertada. El aspecto fundamental a resaltar es su disponibilidad y sus posibilidades en cuanto a funcionalidad. En el marco de este trabajo, se considera que, la utilización de sphinx-4 resultó conveniente, rescatando también lo manifestado por los encuestados.

Finalmente, se analizó la estrategia planteada de solución, y los expertos manifestaron su acuerdo con la misma y presentaron algunas alternativas a tener en cuenta en trabajos futuros. En la siguiente sección se detallan algunos de estos aspectos.

6 Líneas de trabajo futuras

Si bien se ha obtenido un amplio conocimiento de diferentes herramientas, tanto educativas como relacionadas con RV, quedan ciertas modificaciones, mejoras y extensiones a desarrollar en la adaptación presentada, considerando además algunas sugerencias analizadas a partir de la evaluación realizada.

Como líneas de trabajo futuras, se plantean:

- Realizar pruebas con alumnos y docentes del área.
- Permitir la configuración de las etiquetas (el docente podría elegir rotular cada casillero como lo crea más conveniente)

Una de las mejoras ya realizadas, ha sido la forma de analizar si utilizar o no comandos por voz en la resolución de la actividad. En un principio, siempre que se cargaba un proyecto desde el JClicPlayer se consultaba al usuario, si deseaba realizar la actividad con comandos por voz. Dado que JClic no sólo se usa en el ámbito de educación especial, muchas veces esto no era necesario, ya que ningún alumno lo utilizaría. En los casos de educación especial, será una decisión a nivel de cada alumno. Por esta razón, se trabajó para que esta decisión quedara a cargo del docente

inicialmente. El docente decide si se debe preguntar, al cargar el proyecto, si se desean usar los comandos por voz o no

Actualmente se está trabajando en la extensión del prototipo al resto de las actividades de JClic. En principio se comenzará con las actividades de asociación compleja y los juegos de memoria.

7 Referencias Bibliográficas

1. <http://clic.xtec.cat/es/jclic/index.htm>
2. <http://www.tecnoneet.org/docs/2002/2-82002.pdf>
3. Bernal Bermúdez, Bobadilla Sancho y Gómez Vilda, Reconocimiento de voz y fonética acústica. México, Alfaomega grupo Editor, 2000.
4. http://sphinx.subwiki.com/sphinx/index.php/Language_model
5. <http://cmusphinx.sourceforge.net/sphinx4/>
6. <http://leibniz.iimas.unam.mx/~luis/DIME/recursos.html>
7. http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/401a500/ntp_401.pdf
8. Sánchez Montoya, Ordenador y Discapacidad - Guía práctica a las personas con necesidades educativas especiales. Madrid, 2002
9. Rocha Luis, Sistemas de reconocimiento de voz. Revista telegráfica electrónica. Agosto 1986. Pp. 1172-1180.