¿Cómo incorporar voz a nuestras aplicaciones?

M. Carmen Juan Lizandra
(mcarmen@dsic.upv.es)

La incorporación de voz a aplicaciones facilita su interacción con el usuario. El manejo de cualquier aplicación es más sencillo si es posible usar la voz. De esta forma el usuario no tiene que utilizar los mecanismos de entrada habituales del ordenador, teclado o ratón. También es útil para aquellas personas que, por el trabajo que desarrollan, no tienen las manos disponibles para poder usar el teclado o el ratón. E incluso para personas discapacitadas, que no pueden utilizar sus manos. Todos estos usuarios pueden utilizar los programas que se manejan utilizando la voz. Para todos estos usuarios la incorporación de la voz es vital ya que si no fuera así sería como si dichos programas no existieran para ellos.

Actualmente, existen numerosos programas que incluyen voz. La mayoría de ellos son procesadores de texto que permiten incluir el texto mediante dictado en lugar de tener que teclearlo. IBM ha sido de una de las empresas pioneras en este campo, y actualmente oferta VoiceType Simply Speaking Gold que es un sistema de dictado sencillo y discreto. Al decir discreto se entiende que a la hora de pronunciar las palabras deben existir pausas entre ellas. También están en el mercado ViaVoice y ViaVoice Gold, con mayores capacidades que el anterior, ya que en este caso el sistema de dictado es continuo, sin pausa entre palabras. En el número 6 del Manual Formativo de Acta se incluye un artículo *Reconocimiento de voz* [Diaz, 97], en el que se comentan las características del VoiceType Simply Speaking versión 3.02, que es una versión anterior de ViaVoice. Además, IBM ha desarrollado VoiceType Developer's Toolkit, que permite incorporar voz a aplicaciones creadas con entornos de desarrollo tales como Microsoft Visual C++, versión 4.0 o superior, e IBM VisualAge para C++ y Windows, versión 3.5 o superior.

En este artículo, en primer lugar se incluyen una serie de conocimientos básicos para el reconocimiento de voz. Después el artículo se centra en VoiceType Developer's Toolkit. En primer lugar se comentará la estructura y el funcionamiento del reconocimiento de voz en aplicaciones que utilizan VoiceType Developer's Toolkit. En este punto también se describen las características básicas del toolkit y se enumerarán las funciones más utilizadas de su librería. Después se indican los pasos para crear aplicaciones: con comandos y control; para dictado; y tanto para dictado como con comandos y control. Por último se incluye una aplicación ejemplo a la que se ha incorporado voz utilizando el Toolkit.

CONOCIMIENTOS BÁSICOS PARA EL RECONOCIMIENTO DE VOZ

¿Qué es el reconocimiento de voz?

Implícitamente, todos sabemos lo que es el reconocimiento de voz. Lo hacemos cada día, cada vez que mantenemos una conversación con alguien. Pero veamos con un poco más de detalle este proceso. Pensar en el hecho de escribir un mensaje que alguien está dictando para dejárselo a algún amigo. A primera vista parece una operación simple, pero no lo es tanto. El reconocimiento de voz es especialmente complejo si debe realizarse en un ordenador. En este caso, el ordenador debe traducir lo que el usuario pronuncia en forma de texto o comandos que identifiquen e interpreten componentes individuales del mensaje. Es decir, el ordenador debe responder adecuadamente en base a lo escuchado.

Las unidades del discurso son palabras. En el papel, las palabras están compuestas por letras. Cuando se habla, estas palabras están compuestas por sonidos. Luego, cuando se pasa de las palabras pronunciadas a las escritas, se realiza la conversión de sonidos a letras. Sin embargo, los sonidos no corresponden uno a uno a las letras que se utilizan.

Además existen otros factores que pueden dificultar esta tarea, por ejemplo el ruido ambiental.

Existen dos posibilidades a la hora de realizar el reconocimiento de voz:

- Comandos y control
- Dictado

En el caso de comandos y control el usuario simplemente habla por el micrófono para controlar la aplicación. En lugar de seleccionar menús o iconos en la pantalla con el ratón o el teclado, se seleccionan utilizando la voz. Por ejemplo, se puede decir “Abrir fichero” en lugar de seleccionar primero el menú Fichero y después la opción Abrir. Esto se conoce como un comando de voz.

El reconocimiento de voz también permite el uso de la voz para introducir texto de forma libre. Esto se conoce como dictado.

Existe otra característica en los sistemas de reconocimiento de voz y es la dependencia del usuario. Un sistema de reconocimiento de voz puede ser dependiente o independiente del usuario. Un sistema de reconocimiento dependiente del usuario requiere que el usuario entrene explícitamente al sistema para que éste se adapte a sus características de voz propias. Este proceso se conoce como entrenamiento. Por otro lado, un sistema independiente del usuario no necesita entrenamiento.

Así pues, una aplicación con voz será aquella aplicación que esté diseñada para reconocer e interpretar la voz.

Otras definiciones importantes

Un **lenguaje de origen** es el idioma utilizado por el usuario en una aplicación con voz.

Un **vocabulario** es una lista de palabras válidas que se utilizan a la hora de traducir el discurso a texto o comandos.

Un **modelo de uso de palabras** aporta información estadística de secuencias de palabras.

Una **gramática** define la sintaxis, o conjunto de reglas, para las palabras y frases que un usuario puede pronunciar.

Las **pronunciacione**s son las posibles representaciones fonéticas de una palabra.

VOICETYPE DEVELOPER’S TOOLKIT

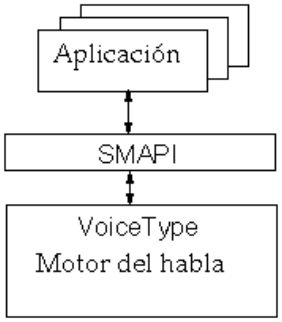
VoiceType Developer’s Toolkit soporta tanto dictado como comandos y control. Las aplicaciones de dictado utilizan habla con pausas entre palabras, aunque las nuevas versiones ya incorporan el dictado continuo, sin pausas entre palabras. Las aplicaciones de comandos y control utilizan discurso continuo, habla sin pausas entre palabras. Estas aplicaciones pueden utilizar discurso continuo porque utilizan pequeños vocabularios que limitan lo que el usuario puede decir a la aplicación.

En este punto se comentará la estructura y el funcionamiento del reconocimiento de voz utilizando VoiceType Developer’s Toolkit.

¿Qué es el motor del habla?

La parte más importante en un sistema de reconocimiento de voz se conoce como motor de reconocimiento del habla o simplemente motor del habla. El motor del habla reconoce las entradas de voz y las traduce a texto que una aplicación entiende. La aplicación decide qué hacer con el texto reconocido.

Las aplicaciones con voz acceden al motor del habla y a varios recursos del habla a través de la API (Application Programming Interface) de reconocimiento de voz. Para VoiceType Developer's Toolkit, esta API se conoce como SMAPI (Speech Manager API). SMAPI es una API convencional. Esto significa que la API está definida como parte de los recursos, en este caso, SMAPI está definida como parte del motor del habla. Con SMAPI, la voz se convierte en un recurso más para todas las aplicaciones, como otros recursos tales como ratón, vídeo, etc.



Recursos del habla

El motor del habla utiliza los siguientes recursos para procesar las palabras pronunciadas:

- Lenguajes de origen
  - Dominios
    - Vocabularios (y extensiones de usuario)
    - Modelos de uso de palabras (y extensiones de usuario)
    - Pronunciaciones (y extensiones de usuario)
  - Modelos de voz (y extensiones de usuario)

VoiceType soporta como lenguaje de origen: inglés americano y cinco lenguajes europeos (inglés británico, francés, alemán, italiano y español).

Cada lenguaje de origen puede incluir varios dominios diferentes que se usan junto con el motor del habla para decodificar el discurso.

El motor del habla utiliza un vocabulario para emparejar las palabras que éste contiene con las entradas (voz) y traducirlas a texto o comandos.

El motor del habla utiliza conjuntamente vocabularios y modelos de uso de palabras para seleccionar el mejor emparejamiento de una palabra o frase.

La aplicación especifica el conjunto de palabras activas activando uno o más vocabularios. Existen tres tipos de vocabularios (comandos, gramáticas y dictado) y el motor del habla los trata de forma diferente.

Vocabularios de comandos. Se utilizan para reconocer palabras o frases de una lista creada dinámicamente en tiempo de ejecución. Los vocabularios de comandos se utilizan con discurso continuo. Por ejemplo, un vocabulario de comandos podría usarse para reconocer menús simples.

Las palabras en un vocabulario de comandos se codifican basándose completamente en cómo suenan. Después de devolver una palabra o frase reconocida, el motor se detiene y espera a que la aplicación solicite la siguiente palabra. El motor se detiene porque es muy probable que el estado de la aplicación varíe como respuesta al comando, y esto podría llevar a un nuevo conjunto de vocabularios activos y válidos.

Para vocabularios de comandos, el modelo de uso de palabras se define por las palabras de la lista. Cada palabra o frase en el vocabulario tiene equiprobabilidad de ocurrir. El usuario únicamente puede pronunciar los comandos especificados en la lista.

Vocabularios de gramáticas. Se usan para reconocer palabras o frases contenidas en una gramática compilada, creada cuando se creó la aplicación. Se usan en discurso continuo.

Después de devolver una palabra o frase reconocida en un vocabulario de gramática, el motor se detiene y espera a que la aplicación solicite la siguiente palabra. La aplicación puede cambiar el conjunto de vocabularios activos mientras el motor estaba detenido.

Para vocabularios de gramáticas, el fichero de gramática define el modelo de uso de palabras, ya que define formalmente el conjunto de palabras disponibles y las secuencias de palabras que el motor del habla puede reconocer. El usuario únicamente puede pronunciar los comandos que estén definidos en la gramática.

Vocabularios de dictado. Utilizados para reconocer palabras de texto de forma libre, pronunciadas utili-

zando pausas. Se puede utilizar un vocabulario de dictado cuando el usuario necesita introducir texto. El modelo de uso de palabras es una base de datos compuesta por secuencias de palabras que aparecen con relativa frecuencia en el lenguaje escrito en un dominio. Durante el dictado, el modelo de uso de palabras ayuda al motor en la selección del mejor emparejamiento para una palabra. Por el contexto, el modelo del uso de palabras posibilita diferenciar palabras que suenan igual acústicamente, tales como hola y ola.

Las palabras en un vocabulario de dictado se decodifican en base a las palabras que tienen a su izquierda y derecha así como de su sonido. Las palabras de la izquierda se refiere a palabras ya decodificadas y las de la derecha a palabras por decodificar.

Durante el proceso de decodificación, el motor del habla selecciona el emparejamiento para las palabras pronunciadas. Esta primera "decodificación" se conoce como palabras débiles.

El motor no se detiene después de cada palabra reconocida. Sigue devolviendo palabras. El motor utiliza el modelo de uso de palabras y el contexto de las palabras vecinas para restringir la selección a una palabra, ésta se conoce como palabra estable.

El motor devuelve una frase parcial de las últimas palabras decodificadas. Esta frase incluye las palabras que se han transformado en palabras estables así como las palabras seleccionadas como palabras débiles.

Para vocabularios de comandos y gramáticas, se utiliza la herramienta de construcción de diccionarios para construir un diccionario que contenga todas las pronunciaciones de las palabras del vocabulario.

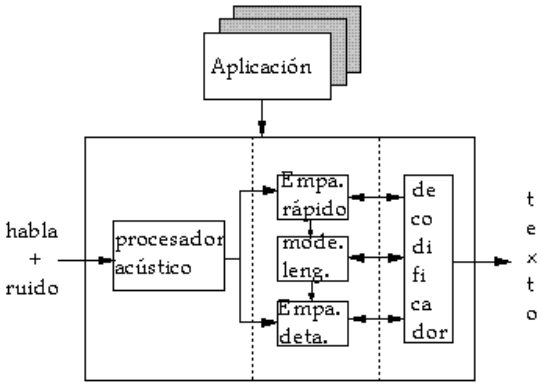
Extensiones de usuario. Expanden el alcance del reconocimiento de voz por el motor. Se pueden definir extensiones dependientes para añadir palabras, pronunciaciones e información de uso de palabras a la aplicación, para incluir palabras que no forman parte del dominio predefinido.

Modelos de voz. El motor del habla utiliza un modelo de voz independiente del usuario cuando decodifica el discurso. La mayoría de los usuarios no tendrán que entrenar al sistema para conseguir un reconocimiento de voz aceptable.

Pero, se pueden crear opcionalmente modelos de voz personalizados grabando una secuencia de sentencias numeradas y preescritas, que es dependiente del lenguaje. El modelo de voz personal contiene características del usuario, tales como acento, que se extrae durante este proceso. VoiceType Developer's Toolkit incluye una aplicación para realizar este proceso.

Arquitectura del sistema del habla

El motor del habla tiene una tarea bastante compleja que realizar: coger la entrada de audio y traducirla a texto reconocido que entienda una aplicación. La siguiente figura muestra la arquitectura lógica del sistema de reconocimiento de voz de VoiceType. En ella se muestran los principales componentes del motor del habla, y cómo interacciona una aplicación con el motor a través de SMAPI.



El **procesador acústico** toma como entrada la señal de audio y la convierte en la forma adecuada para su uso. El procesador acústico consta de dos componentes: el procesador de señales y el etiquetador.

El **procesador de señales** analiza la señal de audio introducida por el micrófono. Esta señal de audio contiene tanto los datos del discurso como ruido ambiental. El procesador de señales debe adaptarse al entorno acústico actual para reducir el impacto del ruido ambiental. El análisis realizado por el procesador de señales produce un conjunto de números, o características, que representan un segmento de tiempo de una centésima de segundo del discurso. Estas características capturan los aspectos importantes del discurso de una forma compacta (tales como energías en diferentes bandas de frecuencia, como las energías del ecualizador en un estéreo).

El **etiquetador** convierte la salida del procesador de señales en un conjunto de etiquetas que identifican varias categorías de sonidos. El etiquetador utiliza planti-

llas, o prototipos, para clasificar los diferentes sonidos del discurso. Estos prototipos corresponden grosso modo con los fonemas del lenguaje hablado. La salida del etiquetador refleja la entrada del discurso determinando la categoría de los sonidos en cada centésima de segundo.

Emparejamiento de palabras. El componente de emparejamiento de palabras realiza la siguiente fase del reconocimiento de voz. Su propósito es identificar las palabras candidatas y clasificar las posibilidades basándose en el análisis acústico e información contextual. Este proceso incluye los pasos: emparejamiento acústico rápido, modelado del lenguaje y emparejamiento acústico detallado.

Emparejamiento acústico rápido. Realiza un emparejamiento aproximado frente a todas las palabras del vocabulario. Esto produce una pequeña lista de palabras candidatas.

Modelado del lenguaje. Analiza las probabilidades de las secuencias de palabras, independientemente de sus formas acústicas. El modelado del lenguaje ayuda a predecir una palabra futura basándose en palabras ya introducidas.

La probabilidad asociada con cada palabra en un contexto se deriva de la frecuencia relativa en un texto representativo. Un modelo de lenguaje se construye a partir de un texto representativo y es específico del dominio (por ejemplo, oficina, periodismo, etc.). El sistema precompila las probabilidades de cada palabra examinando un contexto con varios millones de palabras.

Para vocabularios de comandos dinámicos, el modelo del lenguaje es uniforme, es decir, cada palabra o frase del vocabulario tiene equiprobabilidad de ocurrir.

Para vocabularios basados en gramáticas, el modelo del lenguaje se representa por un fichero que contiene la gramática compilada. La extensión del fichero es: 'FSG'.

Emparejamiento acústico detallado. Realiza un emparejamiento acústico más preciso en el conjunto, más pequeño, de palabras candidatas. Este proceso es computacionalmente más costoso que el emparejamiento rápido. Produce una lista ordenada de palabras candidatas.

Búsqueda. El componente final del motor del habla es el decodificador. Busca la secuencia de palabras más probable utilizando la acústica y los valores del modelo del lenguaje. El decodificador calcula las probabilidades de las cadenas de palabras (sentencias parciales o completas) y utiliza una pila de búsqueda para encontrar la sentencia completa más probable dado el valor del emparejamiento de la palabra a partir del emparejamiento acústico y el modelo del lenguaje. Esta cadena de palabras es la sentencia decodificada.

Interfaz de programación de la aplicación. Además del simple reconocimiento de voz existen más funciones en el motor del habla, incluyendo el manejo de vocabularios dinámicos, funciones de bases de datos para consulta y selección de usuarios instalados, lenguajes y dominios, y la capacidad de añadir nuevas palabras a los vocabularios de los usuarios. La interfaz de programación de la aplicación del motor del VoiceType es SMAPI, que soporta:

- Verificar la versión de la API.
- Establecer una sesión de base de datos para preguntar por los parámetros del sistema (lenguaje, dominio, usuario, etc.).
- Establecer una sesión de reconocimiento.
- Establecer vocabularios.
- Establecer los parámetros del motor del habla.
- Procesar las entradas de voz.
- Añadir nuevas palabras al vocabulario del usuario.
- Manejo de errores.
- Desconexión del motor del habla.
- Cerrar una sesión de habla.

Características básicas de Voicetype Developer's Toolkit

Las versiones de VoiceType Developer's Toolkit actualmente disponibles son: la 4.3 y SDK. Este software es gratis y se puede conseguir en la red accediendo a: [IBM1, 1998]. Además se puede encontrar información sobre el Toolkit en: [IBM2, 1998].

Los componentes del Toolkit son: información sobre la licencia de software, cabeceras y librerías de la interfaz de programación, compiladores de gramáticas, constructores de diccionarios, ejemplos de código y documentación en línea.

El paquete de distribución, generalmente regalado en los distribuidores de IBM a programadores que quieran incorporar voz a sus aplicaciones, contiene los siguientes componentes: Runtime de IBM VoiceType, ficheros para comandos y control, ficheros para dictado, programa de calibración de audio, programas de mantenimiento de vocabularios y programa de instalación.

Para el correcto funcionamiento del Toolkit los requerimientos mínimos del sistema son:

- Pentium 90 MHz con 16 Mb de RAM
- 60 MB de espacio en el disco duro
- Windows 95 o superior
- Cualquier visualizador de HTML para consultar la documentación
- Compilador deseado (Visual C++ o VisualAge)
- Tarjeta de sonido de 16-bits, como SoundBlaster 16 o Pro Audio Spectrum 16. Éstas son las recomendadas, pero funciona también con otras tarjetas de sonido
- Micrófono Andrea ANC-500 o equivalente. Éste de nuevo es el recomendado, pero funciona con cualquier micrófono. Para consultar las características del micrófono Andrea ANC-500 acceder a: [Andrea, 1998]

Para el correcto funcionamiento del Toolkit es necesario disponer de una de las siguientes runtimes:

- ViaVoice
- ViaVoice Gold
- ViaVoice Runtime v. 4.3
- VoiceType Dictation Runtime v. 3.1
- Simple Speaking
- Simple Speaking Gold

Funciones de la librería de Voicetype Developer's Toolkit

En la librería se disponen de una serie de funciones para realizar todas las operaciones necesarias con el fin de incorporar voz a nuestras aplicaciones. A continuación se indicarán algunas de las más utilizadas para la programación utilizando SMAPI, por orden alfabético:

- **SmApiVersionCheck**. Verifica la versión actual del SMAPI. Esta función comprueba cuando la versión del SMAPI utilizada para compilar la aplicación es compatible con la API actualmente instalada en el sistema.
- **SmConnect**. Establece una sesión con el motor del habla. El tipo de sesión deseado y otra información necesaria se establece en los atributos SMAPI. Una vez que se ha establecido una sesión, no se puede cambiar su tipo. Para cambiar el tipo de una sesión, llamar a *SmDisconnect* y después llamar a *SmConnect* de nuevo.
- **SmDefineVocab**. Define un nuevo vocabulario. Esta función crea dinámicamente un nuevo vocabulario que posteriormente se utilizará al llamar a la función *SmEnableVocab*. El vocabulario creado en este caso consta sólo de las palabras especificadas en la llamada. Esta función se puede utilizar para crear vocabularios de comandos en una aplicación. La llamada devuelve una lista de palabras que no tienen pronunciación. Las pronunciaciones se pueden encontrar en un vocabulario predefinido y en un vocabulario personal del usuario. Los vocabularios predefinidos no son dinámicos. Esta llamada es válida únicamente cuando el motor del habla no está decodificando el habla a texto.
- **SmDisableVocab**. Deshabilita un vocabulario definido y ya no se usa por el motor del habla para decodificar el habla a texto durante una sesión de reconocimiento. Únicamente se deshabilita el vocabulario especificado. Cualquier otro vocabulario permanece activo.
- **SmDisconnect**. Cierra la sesión con el motor del habla.
- **SmEnableVocab**. Habilita un vocabulario definido para ser usado por el motor del habla. Posibilita el paso del habla a texto durante una sesión de reconocimiento.
- **SmGetMsgType**. Devuelve el tipo de mensaje de la estructura asociada con la entrada.
- **SmMicOff**. Deshabilita el micrófono, aunque, en una sesión de reconocimiento, después de deshabilitar el micrófono, el motor del habla continua reconociendo las palabras ya pronunciadas. Dependiendo de la velocidad y de lo que se ha pronunciado antes de desconectar el micrófono, este proceso puede llevar varios segundos hasta que se complete.

- **SmMicOn**. Habilita el micrófono. La aplicación debe utilizar *SmRecognizedNextWord* para empezar la decodificación.
- **SmOpen**. Establece una conexión SMAPI e inicializa los valores de una estructura de conexión.
- **SmReceiveMsg**. Recibe un mensaje del motor del habla. Esta función aporta el método a través del cual la aplicación recibe el mensaje asíncrono, incluyendo mensajes asíncronos no solicitados como SM_RECOGNIZED_TEXT del motor del habla. Esta función recibe un mensaje completo del motor del habla.
- **SmRecognizeNextWord**. Habilita el reconocimiento de la siguiente palabra. Esta función busca la siguiente palabra a decodificar. Cuando se está ejecutando, el motor busca en el vocabulario actual habilitado una palabra que empareje con la escuchada. El vocabulario que contiene el mejor emparejamiento determina qué hacer después. Si la palabra pertenece a un vocabulario de dictado, el motor envía un mensaje SM_RECOGNIZED_TEXT a la aplicación y continúa decodificando. Si la palabra pertenece a un vocabulario de comandos, el motor manda la palabra y algunas opciones alternativas a la aplicación en un mensaje SM_RECOGNIZED_WORD. El motor se detiene y espera a que la aplicación le solicite otra palabra. Si una palabra aparece en dos o más vocabularios habilitados al mismo tiempo, el motor selecciona la palabra del vocabulario de comandos habilitado en último lugar.
- **SmSetArg**. Es una macro que rellena una estructura *SmArg*. Esta función establece los componentes del parámetro *arg*. El puntero a *arg* o una lista de argumentos creados de forma similar se pueden pasar como argumentos a un número de funciones tales como *SmOpen* y *SmConnect*.

DESARROLLANDO APLICACIONES

Aplicación con comandos y control

Para crear una aplicación con comandos y control es necesario realizar los siguientes pasos:

1. Identificar qué es lo que el usuario puede decir a la aplicación (el vocabulario).
2. Crear un fichero de gramática y/o definir un vocabulario de comandos dinámico para representar a este vocabulario.
3. Compilar la gramática.
4. Construir un diccionario de pronunciaciones para el vocabulario.
5. Comprobar el vocabulario.
6. Crear la interfaz de la aplicación.

Identificar qué es lo que puede decir el usuario. El primer paso a la hora de incorporar comandos y controles a una aplicación es decidir qué es lo que el usuario puede decir. Por ejemplo se debe saber si el usuario va a poder seleccionar botones y entradas de menú mediante la voz, qué opciones de menús va a poder seleccionar. Si el usuario va a realizar dictado, etc.

Cada colección de palabras y frases que el usuario pueda decir es un vocabulario. VoiceType permite tener múltiples vocabularios al mismo tiempo.

Creando un vocabulario. Se puede especificar un vocabulario de dos formas: bien como una gramática estructurada o como un vocabulario de comandos dinámico.

Para crear los ficheros de gramáticas se utiliza un editor de texto básico. La gramática se especifica utilizando un lenguaje de control de reconocimiento de voz especializado, o SRCL (Speech Recognition Control Language). SRCL es un lenguaje lógico de alto nivel que permite el uso de patrones repetidos y parámetros que se pueden sustituir para definir una sintaxis y conjunto de frases válido. Por ejemplo, un símbolo <dígito> puede definirse como representante de las palabras “cero” hasta “nueve”. Ahora, si se utiliza un número de múltiples dígitos en cualquier otro lugar en la gramática, se puede definir utilizando el símbolo definido <dígito>. Es decir, un número de tres dígitos se puede definir como <dígito><dígito><dígito>.

Los vocabularios de comandos dinámicos son listas de palabras y/o frases que se definen en tiempo de ejecución. Los vocabularios dinámicos son bastante útiles cuando no se conocen todas las palabras del vocabulario cuando se está creando la gramática.

La principal diferencia entre vocabularios de gramáticas y vocabularios de comandos dinámicos son los operadores de sustitución y repetición. Los vocabularios de gramáticas soportan sustitución, mientras que los vocabularios dinámicos no. Esto hace que los vocabularios

de gramáticas sean más adecuados para vocabularios más complejos, y los vocabularios dinámicos más adecuados para comandos de voz simples.

Compilando la gramática. El compilador de gramáticas del VoiceType Developer's Toolkit convierte un fichero de gramática definido en sintaxis SRCL a fichero binario (FSG) que el motor del habla puede usar. En tiempo de ejecución, el motor del habla utiliza el fichero FSG para determinar las palabras y frases que el usuario puede pronunciar. Se pueden compilar múltiples gramáticas para la misma aplicación o compartir gramáticas entre múltiples aplicaciones.

Construyendo un diccionario. Una vez creada la gramática y/o el vocabulario de comandos dinámico, se debe crear un diccionario que contenga la pronunciación de todas las palabras del vocabulario. El motor del habla utiliza este diccionario para saber cómo deben pronunciarse las palabras a reconocer. VoiceType Developer's Toolkit proporciona un constructor de diccionarios que permite la creación automática de un diccionario. Lee y extrae las palabras del vocabulario y asegura que existe una pronunciación para cada una de ellas. Si una palabra no está en el diccionario, se añade. Cuando todas las pronunciaciones son correctas, el constructor de diccionarios genera un nuevo fichero de diccionario.

Comprobando el vocabulario. VoiceType Developer's Toolkit aporta una herramienta para la comprobación de gramáticas que permite comprobar las gramáticas compiladas y los vocabularios de comandos dinámicos. Utilizando esta herramienta, el usuario pronuncia las palabras del vocabulario por el micrófono. Si no reconoce alguna palabra, es necesario modificar el fichero de la gramática, cuya extensión es: 'BNF' o el vocabulario de comandos, o reintroducir la palabra utilizando el constructor de diccionarios para resolver el problema. Si hay más de un vocabulario, en primer lugar se deben comprobar individualmente, y luego combinarlos.

Creando la interfaz de la aplicación. Utilizando la SMAPI, la aplicación puede interaccionar con el motor del habla. Las llamadas necesarias a SMAPI para aplicaciones de comandos y control incluyen establecer una conexión con el motor del habla, hacer que el motor empiece a procesar el discurso, establecer las gramáticas y los vocabularios de comandos dinámicos y activarlos, procesar el discurso reconocido, y desconectar del motor cuando se haya finalizado el proceso.

Aplicación para dictado

En este tipo de aplicaciones es necesario crear la interfaz de la aplicación.

La aplicación necesitará utilizar la SMAPI para interaccionar con el motor del habla a la hora de establecer una conexión, establecer un vocabulario de dictado, y procesar el texto reconocido. Dado que el objetivo es producir texto completamente correcto, se debe manejar la corrección de errores. Esto incluye escuchar lo pronunciado, mostrar palabras alternativas y actualizar el vocabulario.

Las llamadas necesarias a SMAPI para aplicaciones de dictado incluyen establecer una conexión con el motor del habla, hacer que el motor empiece a procesar el discurso, establecer vocabularios de dictado, procesar las entradas de voz, reproducir lo reconocido, corregir las palabras reconocidas erróneamente y desconectar el motor cuando haya acabado.

Aplicación tanto para comandos y control como para dictado

El proceso para desarrollar una aplicación que incorpore tanto comandos y control, y dictado no es muy diferente a los procesos descritos en los puntos anteriores. La consideración principal es el orden en el que se habilitan los vocabularios de comandos y controles, y dictado en la aplicación. Para las partes de comandos y control de la aplicación, es necesario definir una gramática, compilarla, construir un diccionario de pronunciaciones para la gramática y comprobarla. También es necesario crear la interfaz de la aplicación que soporte las características de comandos y control. Para las partes de dictado, se debe crear la interfaz que soporte el dictado.

APLICACIÓN EJEMPLO

Seguidamente se comentará, por encima, una aplicación a la que se le ha incorporado voz utilizando el Toolkit. La aplicación está diseñada para ser utilizada por periodoncistas, y concretamente para que le ayude en el diagnóstico y tratamiento de la enfermedad periodontal. No se va a profundizar en conceptos relacionados con periodoncia.

La información más importante que un periodoncista necesita conocer de un paciente para saber el grado de la enfermedad periodontal, gingivitis o periodontitis, es la profundidad de sondaje. Para obtener la profundidad de sondaje se introduce una sonda dental en la encía del diente a sondar y se determina los milímetros de la sonda que están por dentro de la encía. Actualmente dichas medidas se obtienen visualmente.

La aplicación básicamente es una base de datos de pacientes. Pero no es una base de datos en la que únicamente se incluyen datos de pacientes, sino que incorpora toda una serie de datos, facilidades y hardware específicos para el uso por periodoncistas. Por ejemplo, entre el hardware que se puede incorporar al ordenador, con el fin de obtener el máximo provecho de la aplicación, se encuentra una cámara a la que se ha incorporado una sonda. Dicha conjunción de elementos se utiliza para que el clínico realice el sondaje de los dientes y dicho sondaje se plasme en fotos que la aplicación debe almacenar. El funcionamiento es el siguiente, el usuario selecciona la opción de sondar, bien por voz o manual, y cuando está realizando el sondaje en pantalla aparece la imagen que está visualizando la cámara. Cuando en pantalla aparece la imagen correcta, la aplicación debe congelar, capturar y almacenar dicha imagen. Dicho proceso se realizará tras recibir el ordenador la orden adecuada. La orden podría recibirla vía: teclado, ratón o voz. El periodoncista tiene las manos ocupadas manteniendo la cámara y no puede utilizar ni el teclado ni el ratón. Luego para utilizar el teclado o el ratón sería necesaria la presencia de la enfermera. O bien que la aplicación funcione con voz, esta última desde luego es la ideal y la que permite la aplicación ejemplo. El funcionamiento es el siguiente: cuando en pantalla aparezca la imagen correcta el periodoncista debe pronunciar alguna palabra, y seguidamente se congela y almacena dicha imagen. En este caso, la palabra clave es 'Ya'. Al pronunciar 'Ya' automáticamente se congela y almacena la imagen. Esta imagen será analizada por la aplicación y se obtendrá el valor de la profundidad de sondaje. Si no se hubiera incorporado la voz a esta aplicación no sería posible su uso sin la ayuda de la enfermera, que sería la que utilizaría el teclado o el ratón para interaccionar con la aplicación, mientras el periodoncista realiza el sondaje. Incorporando voz a esta aplicación se facilita el trabajo al periodoncista y hace posible su uso por una única persona.

En la siguiente imagen se muestra la pantalla principal de la aplicación.

La aplicación ejemplo puede funcionar utilizando el teclado y el ratón, voz o indistintamente una u otra, si está seleccionada la opción de habilitar la voz. Para que esté habilitada la voz se debe seleccionar el botón  . Si está seleccionada la voz y se selecciona de nuevo, se deshabilita dicha modalidad.

La aplicación dispone de un diccionario de palabras clave que asocia con identificadores de botones, menús, o cualquier otra opción que haga que el programa realice alguna operación. Concretamente, todos los botones y todas las entradas de menú se pueden seleccionar utilizando el teclado, el ratón o la voz. Además la introducción de los datos del sondaje también puede ser por voz, es decir, que el periodoncista no disponga de la cámara y que las mediciones las efectúe visualmente y los datos los introduzca por voz a medida que realiza el sondaje. En definitiva, prácticamente todas las operaciones que pueden realizarse con el teclado y con el ratón pueden realizarse utilizando la voz.

En principio la aplicación con voz funciona con independencia del usuario que la utilice. Aunque puede entrenarse para adaptarla más a la forma de hablar de cada persona.

CONCLUSIONES

La incorporación de voz a aplicaciones es un mecanismo que facilita la interacción con la máquina. Una de las ventajas importantes que presenta es que la inclusión de textos, de hojas y hojas, pueda realizarse mediante dictado, en lugar de teclear todo el texto. También facilita el acceso a opciones de los menús o añadir determinada información a la aplicación. Pero quizá su principal ventaja sea su uso por personas que no pueden utilizar sus manos para interaccionar con el ordenador. Personas que por las circunstancias que sean no pueden utilizar los dispositivos de entrada habituales de un ordena-

dor o personas discapacitadas físicamente. En todos estos casos los programas que llevan incorporada voz son muy valiosos y de otro modo sería como si no existieran para estos usuarios.

Actualmente, en el mercado existen numerosas aplicaciones a las que se les ha incorporado voz. Y con las facilidades que ofrece VoiceType Developer's Toolkit cada vez irán apareciendo más.

Concluyendo, VoiceType Developer's Toolkit es una herramienta muy útil para incorporar voz a nuestras aplicaciones y además es gratis.

REFERENCIAS

Andrea Electronic Corporations, Internet, http://www.andreaelectronics.com/prodmn. html, 1998

Díaz-Hellín Sepúlveda, Fernando, Reconocimiento de voz, Acta Manual Formativo, nº 6, 1997

IBM1, Internet, http://www.software. ibm.com/is/voicetype/dev_prods.html, 1998

IBM2, Internet, http://www.software. ibm.com/is/voicetype, 1998