Capítulo 4. SAPI.

4.1 ¿Qué es Microsoft Speech Application Program Interface (SAPI)?

La SAPI es una interfaz de reconocimiento del habla y de síntesis de voz para la programación de aplicaciones basadas en Win32 (Intel Win32s, Windows NT, Windows 95/98, MIPS Windows NT, DEC Alpha Windows NT, Power PC Windows NT, Windows XP). "Actúa como una capa de abstracción entre las aplicaciones y los motores (máquinas sintetizadoras y reconocedoras) sobre tecnología del habla" [11]. También es una interfaz entre los motores y el hardware de reconocimiento de voz (*SR. Speech Recognition*) y de síntesis de voz (*TTS. Text to Speech*).

La SAPI tiene interfaces para 4 aplicaciones fundamentales que son:

- Reconocimiento de comandos. El usuario da como entrada una serie de comandos predefinidos por medio del habla que serán transformados a texto escrito y que la aplicación entenderá como ordenes y ejecutará. Esta aplicación es en la que nosotros profundizaremos.
- Texto hablado (*Text to Speech*). Transforma un texto escrito a texto hablado por medio de síntesis del habla.
- Dictado. Transformación de habla continua a texto escrito que será usado en alguna aplicación (procesador de palabras).
- Telefonía. El usuario se comunica con alguna aplicación por medio del teléfono.

La SAPI tiene varias interfaces que sirven para las aplicaciones antes descritas. Estas interfaces están divididas es dos niveles [12]:

- *High level Interfaces* (Interfaces de alto nivel). Fueron diseñadas para hacer más fácil la implementación, pero se pierde un poco el control. Son para resultados rápidos pero pueden ser no tan efectivas. Son simples, exigen poco código y pueden compartir recursos.
- *Low level Interfaces* (Interfaces de bajo nivel). Estas interfaces son más difíciles de implementar que las de alto nivel, pero dan más control en la aplicación. Son más eficientes y más flexibles.

En el diseño de una aplicación se debe decidir con que nivel se va a trabajar ya que son distintos y cada uno tiene sus ventajas y sus desventajas.

Las interfaces de alto nivel son [12]:

- *Voice command*. Se encarga del reconocimiento de comandos por medio del habla.
- *Voice dictation.* Se encarga del reconocimiento continuo del habla.
- *Voice Text.* Se encarga de transformar un texto escrito a voz (sintetizar voz).
- *Voice Telephony*. Se encarga del reconocimiento continuo del habla por medio de líneas telefónicas.

Las interfaces de bajo nivel son [12]:

- *Direct Speech Recognition*. Se encarga del reconocimiento de voz a bajo nivel.
- *Direct Text to Speech*. Se encarga del la síntesis de voz a bajo nivel.

Como se mencionó al inicio del capitulo la SAPI necesita del sintetizador y el reconocedor. El reconocedor que nosotros utilizaremos es el Reconocedor *Microsoft Speech Recognition engine.*

El Microsoft Speech Recognition 4.0.4.2512 es el reconocedor (máquina) que utilizamos para este proyecto. El nombre del programa ejecutable es actcnc.exe su tamaño es de 6Mb. Este da las capacidades de reconocimiento a la aplicación siendo posible el interactuar con él y emitir comandos verbales que a su vez recibirán respuestas auditivas por parte de la aplicación.

Como se había mencionado antes, para crear aplicaciones que utilicen reconocimiento de voz, la SAPI utiliza una máquina (reconocedor), nosotros utilizamos Microsoft Speech Recognition, pero puede ser cualquiera que sea compatible con SAPI.

4.2 Gramática.

En el proceso del reconocimiento de voz, se necesita saber que palabras pueden ser reconocidas; a esto se le conoce con el nombre de gramática [12]. El reconocedor usa la gramática para determinar a que palabra se parece más lo que reconoció.

En el caso del reconocimiento de comandos, las palabras permitidas están limitadas a los comandos elegidos, es decir tienen un dominio específico y limitado a diferencia del dictado. La gramática define reglas que dicen que palabras pueden ser dichas y esto facilita

el reconocimiento. En vez de poder reconocer cualquier palabra, éste se limita a reconocer solo las palabras que correspondan a la gramática.

Ahora explicaremos con más detalle las interfaces de alto nivel ya que con una de estas es con la que trabajamos en este proyecto.

4.3 Interfaces de Alto nivel.

Las interfaces de la SAPI están hechas como *Control Object Model interfaces* (COM interfaces); Delphi Object Pascal y Visual Basic pueden usarlas fácilmente. También existen estas mismas interfaces en una forma simplificada llamada *Automation interfaces* para ser usadas por lenguajes como BVA y finalmente también pueden ser usadas como controles ActiveX, para ser usados como componentes en ambientes de desarrollo que los soportan como es el caso de Delphi 7. [12], lenguaje sobre el cual trabajamos.

Las interfaces son:

- COM. Contiene *Voice Text API, Voice Command API, Voice Dictation API* y *Voice Telephony API.* Estos APIs de alto nivel son implementados para llamar a los APIs de bajo nivel.
- *Automation.* Contiene *Voice Text Object Automation* y *Voice Command Object Automation*. No existe interface para *Voice Dictation.*
- ActiveX. Simplifica el proceso de construir aplicaciones con SAPI. Con uno de estos controles es con el que trabajamos.

Cuando se instala SAPI 4, se hace en dos pasos. En el primer paso se instalan las interfaces COM y *Automation.* En la segunda parte se instalan los controles del ActiveX (Ver Apéndice D. Instalación de SAPI en español). A continuación veremos de forma más detallada como se usan estos controladores [12].

4.3.1 ActiveX

El *Speech Development Kit* con el que trabajamos (SAPI 4) contiene 6 controles de ActiveX que dan fácil acceso al reconocimiento y síntesis de voz. Estos son:

- *Microsoft Direct Speech Recognition.*
- *Microsoft Voice Commands* ó *Voice Commands Control.* Es un control del ActiveX que envuelve todas las funciones del *Voice Command API* de alto nivel. Este es el que nosotros usamos en este proyecto. Para entender a más detalle a que nivel se encuentra *Voice Command Control* ver figura 4.1.
- *Microsoft Dictation* ó *Voice Dictation Control*. Es un control del ActiveX que envuelve todas las funciones del *Voice Dictation API* de alto nivel.
- *Microsoft Direct Speech Synthesis.*
- *Microsoft Voice Text* ó *Text To Speech Control.* Es un control del ActiveX que envuelve todas las funciones del *Voice Text API* de alto nivel.
- *Microsoft Speech telephony*. Implementa una interfaz que permite reconocimiento de voz y síntesis de voz.

Las versiones *Direct* dan acceso completo a las interfaces SAPI para síntesis y reconocimiento de voz. Estas trabajan con las interfaces de bajo nivel dando mayor

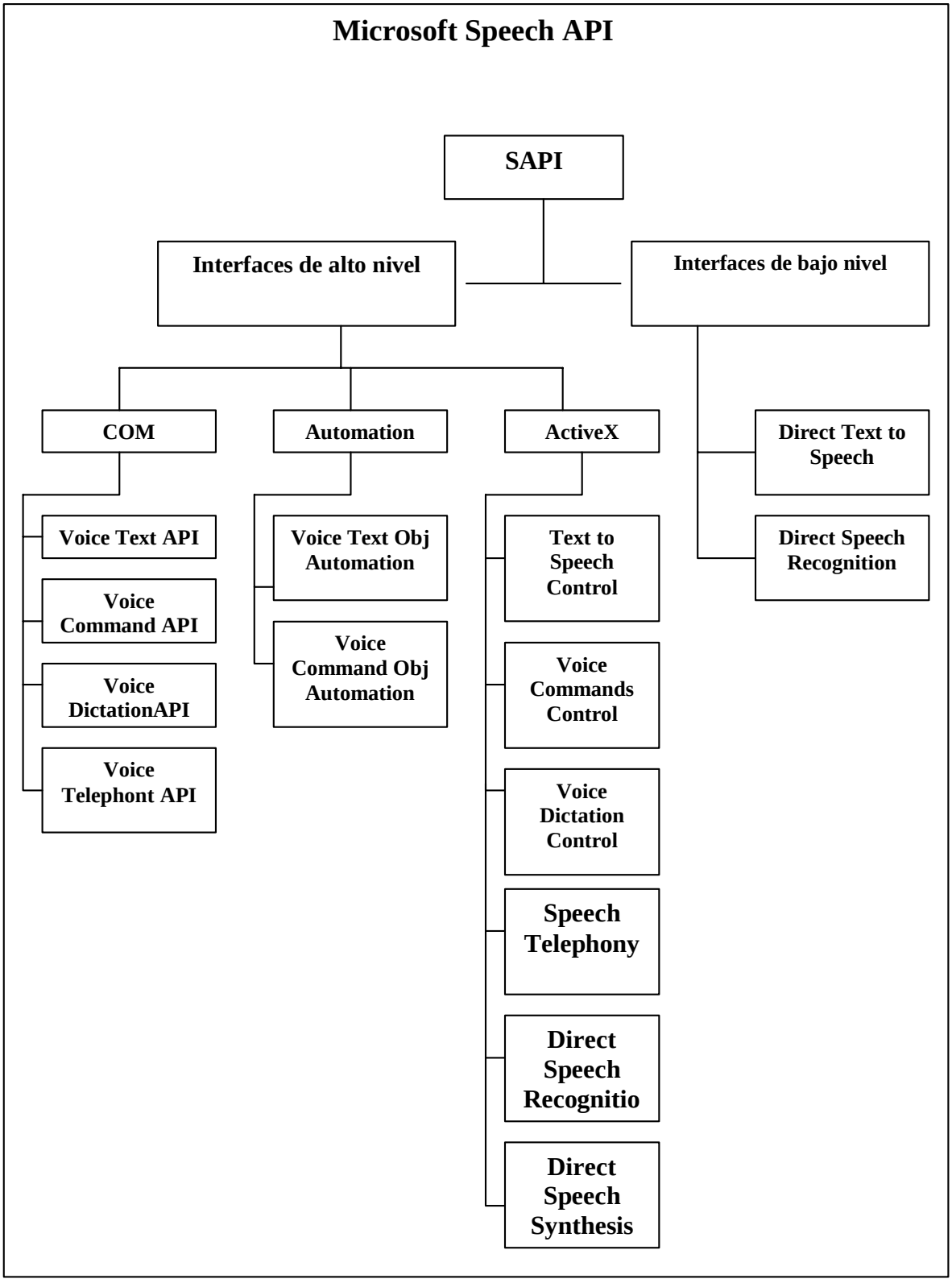

Figura 4.1 Diagrama SAPI (basado en [12])

velocidad y más control de programación. Voice Commands and Voice Text trabajan con las interfaces de alto nivel. Estas interfaces son más limitadas y un poco más lentas que las Direct APIs, pero proveen recursos automáticos y memoria [12].

4.3.1.1 Voice Commands Control.

El reconocimiento de voz *Command and Control* permite al usuario hablar palabras, frases u oraciones de una lista de frases (menu) que la computadora espera escuchar. Es una forma de implementar menús de voz que comparten el reconocedor y trabajan con comandos de voz. Para implementar estos menús, se hace uso de propiedades, métodos y eventos propios del objeto *Command and Control* (Ver Apéndice E. Documento de Especificación de métodos, propiedades y eventos del *Command and Control* del AvtiveX.). El objeto *Command and Control* es menos flexible que el *Direct Recognition Object*, pero es más fácil de usar [13].

Es recomendable usar *Command and Control* cuando la aplicación sea de reconocimiento de comandos (palabras) porque hace la aplicación mucho más fácil de usar. Otras razones del cuando es conveniente usar Command and Control son:

- Una aplicación diseñada para que el usuario conteste preguntas específicas como SI/NO. El reconocedor fácilmente entiende estas respuestas y algunas otras repuestas cortas.

- Activar macros. Hace más fácil el uso de la aplicación ya que es más facil decir cosas como “abrir archivo”, que aprenderte el comando Cntrl + A o que usar el ratón.
- Humaniza la computadora. El reconocimiento de voz puede hacer que la computadora parezca como una persona. Alguien como el usuario habla y recibe una contestación. Esta capacidad puede hacer a los juegos mas realistas y aplicaciones de entretenimiento o educacionales mas amigables [13].

El uso específico de reconocimiento *Commmand and Control* es dependiente de la aplicación para la que se dirija. Algunas de las aplicaciones para las que es conveniente son:

- Juegos y entretenimiento. Hace más amigable los juegos si el usuario interactúa por medio de la voz con el sistema.
- Entrada de datos. Para los capturistas de bases de datos resulta más fácil leer los datos que teclearlos
- Edición de documentos. Es más fácil decir “cambiar a negritas” que usar el ratón.
- Telefonía.

Sin embargo, también el uso del *Command and Control,* tiene limitaciones (que son las mismas para cualquier tipo de reconocimiento). Estas limitantes son:

- Calidad del micrófono.
- Ruido en el ambiente.
- Sonidos generados por la computadora.

- El reconocimiento de voz comete errores.
- Este es exclusivo del reconocimiento de comandos. *Command and Control* necesitan los comandos exactos.