

03063



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“DISEÑO DE INTERFACES VERBALES CON
AGENTES CONVERSACIONALES”**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN INGENIERÍA
(COMPUTACIÓN)**

P R E S E N T A:

RAMÓN RAMÍREZ GUZMÁN

**DIRECTOR DE TESIS: MAT. MARÍA CONCEPCIÓN ANA
LUISA SOLÍS GONZÁLEZ COSÍO**

México, D.F.

2005.

4:350507



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS.

A Ana Luisa Solis (mi tutora): Por brindarme su ayuda y darme su apoyo para que el presente trabajo llegara a los niveles presentados.

A Dr. Fernando Gamboa, Dr. Jesús Savage, Dra. María Elena Martínez y Dr. Boris Escalante (revisores de tesis): Por transmitirme sus conocimientos y sus consejos para la mejora del presente trabajo.

A Erika E. Rocha Cordova: Por su apoyo y ayuda que me brindo en todo momento, ya que gracias a ello se pudo concluir el presente trabajo.

A José Luis Villarreal Benítez: Ya que su apoyo y guía me ha ayudado a mejorar mis habilidades en mi desarrollo tanto académico como profesional.

A Marcelo Pérez Medel: Que además de su apoyo y sus enseñanzas, me ha dado la pauta para obtener los logros que he alcanzado.


A la UNAM: Que me permitió crecer en el ámbito académico y me permite seguir creciendo en el ámbito laboral.

A mi familia: Por el apoyo moral que me brindaron.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo profesional.

NOMBRE: Ramírez Guzmán Ramón

FECHA: 22 / Noviembre / 2005

FIRMA: 

ÍNDICE.

- Índice.....	1
- Introducción.....	3
- Objetivos	4
- Contribución y relevancia	4
- Metas	4
- Metodología.....	5
1 Capítulo 1. Interfaces inteligentes	7
1.1 Introducción.....	8
1.2 Tipos de interfaces inteligentes.....	11
1.3 Aplicaciones	14
2 Capítulo 2. Agentes conversacionales.....	17
2.1 Agentes virtuales.....	18
2.2 Conversación	21
2.3 Arquitectura.....	22
3 Capítulo 3. Procesamiento del lenguaje natural	27
3.1 Palabras	28
3.2 Sintaxis.....	33
3.3 Semántica	37
4 Capítulo 4. Herramientas de síntesis de voz	39
4.1 Reconocimiento de voz.....	40
4.2 Síntesis de texto a voz	44
5 Capítulo 5. Arquitectura de una interfaz basada en diálogo en español.....	47
5.1 Arquitectura	48
5.2 Definición de la metodología para el PLN	51
5.3 Procesador del lenguaje natural.....	53
5.3.1 Pre-Procesamiento del diálogo	54
5.3.2 Procesamiento del diálogo.....	55
6 Capítulo 6. Desarrollo y resultados de la interfaz para el diálogo.....	59
6.1 Caso de uso	60
6.2 Diseño de la interfaz siguiendo la metodología definida	62
6.3 Pruebas del sistema.....	67
Conclusiones	71
Literatura citada	75

INTRODUCCIÓN.

Una de las partes más importantes para las aplicaciones son sus interfaces, ya que son éstas las que permiten al usuario interactuar con los programas. Con el tiempo han ido modificándose, tanto en su aspecto como en la forma de utilizarse, por tal motivo existen distintos tipos de interfaces. Dentro de estos tipos se encuentran las interfaces inteligentes, entre las cuales también se encuentran varios tipos de acuerdo a la forma en que se utilizan o, el objetivo para el cual fueron diseñadas.

Este tipo de interfaces mejora la interacción entre los usuarios y los sistemas computacionales, ya que brindan un control más natural para los humanos, además pueden dar ayuda más comprensible a la que muestran las aplicaciones comúnmente. Además, con este tipo de interfaces se pueden tener sistemas muy complejos, que pueden ser utilizados por usuarios que no tengan un alto conocimiento del contexto en el que se desenvuelve, ya que su forma de interactuar será por medio de actos sencillos (y en ocasiones controlados) o por acciones de la vida cotidiana.

Dentro de las interfaces inteligentes se encuentran los agentes virtuales, y más en específico los agentes conversacionales, los cuales para ser considerados como tales, deben cumplir con ciertas características. Unas de sus características primordiales, es precisamente su capacidad de mantener una conversación con el usuario, la cual puede ser tan restringida o amplia como el sistema la tenga definida. La complejidad de la conversación que puede mantener el agente, depende del contexto y los algoritmos para dicho fin.

De los algoritmos que se necesitan para la conversación, se tienen tres tipos: Los que realizan el reconocimiento de voz, los dedicados al procesamiento del lenguaje natural, y los especializados en pasar un texto escrito a un sonido que reproduce dicho texto hablado, esto último se conoce como TTS (Text To Speech). Dentro del procesamiento del lenguaje natural, también se tiene una subdivisión de acuerdo al planteamiento de dicho problema y las partes que lo componen, que son precisamente lo que debe revisarse en las frases, estas son: las palabras, la sintaxis y la gramática. Para cada una de estas se cuenta con un conjunto de algoritmos, donde se pueden encontrar algunos que se utilizan en las tres partes y están basados en el mismo principio o método.

Para el reconocimiento de voz, y la síntesis de texto a voz (TTS), se manejan las mismas bases, aunque los algoritmos sean distintos por tratarse de problemas diferentes. Estos algoritmos no se revisan a fondo ya que se utilizaron bibliotecas que permitieran agregar dichos funcionamientos a la aplicación, pero si se revisan las bases teóricas ya que son necesarias para poder utilizar las herramientas mencionadas.

Para el diseño de la interfaz verbal se replanteo el problema, y se modificaron los algoritmos según convino. El motivo fue que los algoritmos se enfocan en el idioma inglés, y el objetivo era tener dicha interfaz en español, para dicho fin se analizó el lenguaje y se modificaron los algoritmos para adecuarse a la estructura de dicho idioma. Como base para el diseño de la interfaz y el enlace

de los distintos módulos, se planteo un esquema general, el cual contiene todas las partes involucradas y las relaciones entre ellas.

Por último se muestra un ejemplo de la construcción de una interfaz verbal utilizando la metodología presentada en este trabajo. Con este ejemplo, además de ver la forma en que debe ser utilizada la metodología, también se muestra como pueden ser empleados distintos algoritmos para cada parte, y como influye la elección en el rendimiento del sistema. Los algoritmos utilizados en el ejemplo no son los más sofisticados pero cumplen con su cometido.

OBJETIVOS.

Se tienen varias ventajas en el uso de tecnologías del lenguaje para el control de agentes. Mientras la entrada hablada es considerada como una interfaz más natural para el usuario, nos puede brindar nuevas dimensiones para dicho control. El uso del control por medio del habla permite al usuario concentrarse en la escena visual disminuyendo la necesidad de adaptarse a los dispositivos de entrada y salida. El habla puede también llevar a más información abstracta o a comandos de alto nivel.

El objetivo perseguido es diseñar un procesador de lenguaje natural (en español), capaz de utilizarse como una interfaz basada en diálogo que libere a los usuarios de los dispositivos de entrada y salida tradicionales. Esta se añadirá a un agente controlado por el habla (agente conversacional), enfocándose a la entrada verbal como control directo para el avatar sin tomar en cuenta el comportamiento autónomo. Por tal motivo se requiere que el procesador arroje una respuesta verbal por medio del agente, así como un conjunto de instrucciones para que el avatar realice ciertos movimientos, ya sea adecuados o necesarios para cada entrada verbal del usuario (por ejemplo: que se mueva hacia un lugar en específico o de una indicación con las manos).

CONTRIBUCIÓN Y RELEVANCIA.

El problema del diseño de un procesador del lenguaje natural para el idioma español va más allá de seguir las ideas y los métodos utilizados en otros procesadores de este tipo, debido a que los existentes, como ya se ha mencionado, se han basado en técnicas diseñadas para lenguajes sintéticos. Al ser el español un lenguaje analítico, se debe replantear el diseño del procesador de tal manera que se obtengan nuevas técnicas enfocadas a este tipo de lenguajes.

METAS.

- Lograr una interfaz para un agente conversacional que pueda entender la mayor parte de las frases utilizadas dentro de un contexto específico.
- Se requiere que la interfaz pueda generar una respuesta coherente a la frase emitida por el usuario.

- La interfaz generará un conjunto de instrucciones, las cuales le indicarán al agente el comportamiento que tendrá de acuerdo a la entrada verbal del usuario y la situación de la conversación.

Dicha interfaz estará constituida por el procesador del lenguaje natural.

METODOLOGÍA.

Para el diseño de las interfaces verbales, se planteó una metodología, tomando en cuenta trabajos anteriores y algoritmos previamente desarrollados. Dentro de ésta metodología se tiene las siguientes consideraciones:

- Las partes de reconocimiento de voz y síntesis de texto a voz (Text To Speech o TTS), son implementadas por medio de herramientas previamente desarrolladas, dichas herramientas son revisadas en el capítulo 4.
- El idioma seleccionado para las interfaces es el español, esto debido a que es uno de los objetivos que se persiguen en el presente trabajo.
- El enfoque inicial que se le dio al problema fue obtener una interfaz verbal para tutores en laboratorios virtuales (aunque posteriormente se vio que podía ser utilizada para agentes en otros contextos), de ahí que el contexto deba estar definido para la metodología planteada.

Tomando en cuenta dichas consideraciones, la metodología presentada contiene las siguientes partes, explicadas con mayor detalle en el capítulo 5.

- Primero se define el contexto donde se va a desenvolver el agente.
- Después se define un sublenguaje el cual debe contener al menos la mayoría de las palabras que se utilizan en dicho contexto y necesariamente las más comunes.
- Se construyen todas las estructuras sintácticas, tomando en cuenta el contexto y el sublenguaje.
- Se toman dichas estructuras para formar un autómata que revisará la sintaxis de la frase del usuario.
- Se revisan las posibles respuestas con base en las estructuras sintácticas obtenidas, con el objetivo de obtener el conjunto de posibles respuestas y acciones del agente, así como la relación que tienen unas con otras.
- Enlazar dicha metodología con las herramientas antes mencionadas, donde la salida del reconocedor de voz (la cual debe ser una cadena de caracteres) es la entrada al módulo programado siguiendo dicha metodología, y la salida de éste último es la entrada de la herramienta que realiza el TTS.

CAPÍTULO 1

INTERFACES INTELIGENTES

1.1 Introducción.

Desde el inicio de la computación, las aplicaciones han tenido una forma de comunicarse con el usuario y viceversa, a la parte que hace posible dicha comunicación se le conoce como interfaz. Así como las aplicaciones y el equipo de cómputo, las interfaces han evolucionado con el paso del tiempo, ya sea por las nuevas capacidades de los equipos de cómputo, los nuevos métodos o algoritmos empleados, o por simple necesidad. Pero una tendencia de la evolución de las interfaces es siempre facilitar el trabajo del usuario, y una de las formas que se han empleado para lograr dicho objetivo es tratar que la interfaz trate de simular una actividad más natural para los usuarios.

Anteriormente la comunicación del usuario hacia la computadora se hacía por medio de tarjetas perforadas, lo cual complicaba el uso de los sistemas de cómputo, pero con la aparición del teclado asemejando a una máquina de escribir, se logró facilitar el uso de las aplicaciones y mejorar el desempeño de los usuarios. La familiaridad que se tenía con las máquinas de escribir ayudó al teclado a ser aceptado por los usuarios de los equipos de cómputo, a tal medida que aun en estos tiempos es un accesorio muy indispensable para casi todos los sistemas computacionales. De esta forma han surgido nuevas maneras novedosas de interactuar, como lo fue en su tiempo el ratón, y como lo están siendo los nuevos dispositivos que van apareciendo en el mercado [11].

Pero la evolución de las interfaces no se queda únicamente en los dispositivos, conocidos como hardware, sino que también van evolucionando como aplicaciones. Anteriormente cuando ya se tenía el teclado, la interacción se realizaba por medio de comandos, posteriormente surgieron las interfaces gráficas de usuario (GUI), las cuales presentan a las aplicaciones en determinada área del dispositivo de despliegue, a estas áreas se les denomina ventanas las cuales contienen componentes (como los botones) que son utilizados para que el usuario le diga a la aplicación que realice una acción en específico.

De esta manera las interfaces también han ido evolucionando dentro de las aplicaciones, lo que es conocido como software, donde la tendencia siempre ha sido hacer que el usuario se sienta más cómodo utilizando el nuevo diseño. En las nuevas interfaces, se ha tratado de que la máquina ofrezca posibilidades de interacción donde el usuario realice acciones comunes de su vida cotidiana, ya que de esta manera no se necesita un entrenamiento extra para poder utilizar los sistemas que tengan este tipo de interfaz.

Pero esas interfaces no son comunes, ya que deben tener un cierto grado de inteligencia para poder cumplir con su cometido, sin tener la intervención de un humano diciendo qué pasos seguir para resolver el problema. Por tanto, estas interfaces son autónomas, ya que interactúan con el usuario emulando el comportamiento de otro humano con un cierto nivel de complejidad. A este tipo de interfaces se les conoce como interfaces inteligentes, y aunque no intenten emular el comportamiento humano, si es una interfaz con cierto grado de inteligencia para resolver un problema en la

interacción, ya sea con el usuario o el entorno que lo rodea, es considerada como tal. Esto debido a que las características deseables de una interfaz inteligente según las consultas realizadas por el programa europeo IST (Information Society Technologies), dichas características son resumidas a continuación con base en [11]:

- Facilidad para su manejo, que muestre las funciones al usuario y que reduzca el tiempo de aprendizaje.
- Presentación inteligente de la información, teniendo en cuenta las necesidades del usuario y la naturaleza de los dispositivos con que se cuenta.
- Acceso de la información por parte de todos los elementos que la necesiten.
- Capacidad de aprendizaje y mejora de las capacidades del usuario.
- Dispositivos específicos.

Hay que notar que esta es sólo una propuesta de la IST, y que pueden existir otras características o definiciones, ya que es parecido a la definición de realidad virtual, cada quien la define como quiere, pero todos (o al menos la mayoría) contienen una idea general. Por lo mismo, las interfaces inteligentes pueden contener una o varias de estas características, pero al menos debe tener una para ser considerada como tal, por lo que es necesario saber a que se refiere cada una de éstas.

De la primera de las características, con la facilidad para su manejo se refiere a que no debe complicar las acciones que puede realizar el usuario para interactuar con el sistema, por lo que éste debe ofrecer herramientas suficientes que el usuario pueda entender con cierta facilidad, con lo que se elimina el entrenamiento para conocer la forma en que se utilice la aplicación. Dichas herramientas deben mostrar las funciones que puede realizar el usuario, de esta manera todas estas funciones deben estar plasmadas en la interfaz, esto no quiere decir que debe tener una representación visual, ya que como se ha visto, no todas las interfaces son gráficas, por tanto la forma en que se plasman dichas funciones en la interfaz es en como se utiliza, ya que la manera en que se utilice debe reflejarse en el sistema, por ejemplo, si la interfaz hace que el usuario realice una acción que asemeje el movimiento de un bloque, ese bloque debe tener una representación dentro del sistema que realice una traslación con respecto al movimiento que realiza el usuario.

Este tipo de acciones disminuye las capacidades que debe tener el usuario para poder utilizar el sistema, sin disminuir las capacidades de este último, y si el usuario no llega a tener los conocimientos suficientes, el entrenamiento requerido para adquirirlos disminuye su complejidad con lo que se reduce el tiempo para adquirirlos.

La segunda característica se refiere a que la información que genera el sistema debe presentarse de una forma coherente, y de una manera que pueda reflejar la forma de utilizar la interfaz. Para esto se debe tener en cuenta los dispositivos que tiene el sistema, ya que de éstos dependerá la forma en

que se pueda presentar la información, esto es por que no se puede presentar una respuesta gráfica si no se tiene un dispositivo de despliegue. Se puede aprovechar que en la actualidad se cuenta con un gran número de dispositivos que pueden ser conectados a las computadoras, lo que ocasiona que se tengan muchas posibilidades de mostrar la información, no sólo gráficamente, sino que también por medio de un dispositivo mecánico o de otra índole.

Pero no se debe olvidar que la información que obtiene el sistema como salida es para el usuario, y por tal motivo se deben tener en cuenta sus necesidades para elegir los dispositivos y la forma de utilizarlos para representar dicha salida. Aunque no se debe olvidar las limitaciones de los dispositivos con que se cuenta, ya que no siempre se puede contar con todo lo que se requiere, en tal caso se deben realizar modificaciones (en este caso sobre la representación de la salida) para poder representar la información de la manera más apegada a lo más conveniente para el usuario.

En la otra característica se toma en cuenta que la información que contiene el sistema, ya sea de entrada o de salida, debe estar al alcance de quien la necesite, esto es que dicha información la deben poder acceder cualquier parte del sistema así como el o los usuarios que estén involucrados en el sistema. De esta forma, la interfaz debe tener un control total sobre la información que contiene el sistema, y debe ser capaz de administrarla y distribuirla entre los distintos elementos que la soliciten. Aunque cada uno de los elementos en particular necesitan sólo de una parte de toda la información, por tal motivo además de realizar las acciones anteriormente mencionadas, también debe encapsular la información adecuada que necesita cada elemento, de tal forma que la puedan entender.

La capacidad de aprendizaje del sistema es la forma en que el sistema registra las preferencias de los usuarios al ser utilizado, de esta forma se puede hacer ya sea más agradable o incluso utilizar datos de sesiones anteriores cada vez que se inicia una nueva. Otra de las posibilidades es tener una sesión de inicio y utilizar dichos datos todas las veces que se utilice el sistema, aunque con lo que se mencionó, el sistema podría registrar un cambio en las preferencias que tiene el usuario a la hora de usar el sistema. Pero estos cambios no son los de la clásica configuración del sistema, sino que él mismo debe tener la capacidad de poder captar los cambios que pueden ser pertinentes para mejorar el desempeño de los usuarios.

Para la última de las características mencionadas se debe tener en cuenta el objetivo que persigue el sistema y la forma en que el usuario debe ingresar los datos y como se deben presentársele. Como se mencionó anteriormente, en la actualidad se cuentan con un gran número de dispositivos que pueden ser utilizados como interfaz, dentro de éstos hay un número que ya tienen integradas sus interfaces inteligentes como las tarjetas con chips que últimamente se están utilizando. Pero donde más importancia tienen los dispositivos y por tanto mayor es el número de los que pueden ser utilizados, es en la realidad virtual.

De los dispositivos que se tienen para este fin se encuentran los guantes que tienen integrados sistemas para simular el tacto a la hora que el usuario toca o incluso toma algún objeto dentro del ambiente virtual, también se cuenta con lápices conectados a un brazo mecánico que ayudan al usuario a seguir el contorno de las geometrias mostradas o incluso puede ser utilizado para seleccionar alguna de ellas o modificarla. Otro de los dispositivos más comunes dentro de dicha área son los HMD (Head Mounted Displays), también conocidos como cascos de realidad virtual, estos además de poder presentar el ambiente virtual con efectos que aumentan el realismo haciendo que los objetos se vean como si en realidad estuvieran en la posición mostrada, también pueden contar con sensores que registran los movimientos de la cabeza y/o los cambios de lugar que realice el usuario al caminar.



Figura 1.1. Interfaz inteligente para realidad virtual, propiedad de SIA sistemas.
<http://www.siasistemas.com/sitio2/020604.htm>

Se puede observar que cada una de las características señaladas, pueden estar presentes en una interfaz inteligente, ya sea que sólo se cuente con una o que estén integradas todas en un mismo sistema. Es importante recordar que las características que se mostraron pueden diferir de las consideradas por otro organismo o personas que tengan su propio conjunto, pero en general se pueden referir los mismos objetivos que deben de cumplir este tipo de interfaces, y las mostradas aquí, pueden servir como una referencia confiable abalada por el organismo IST europeo.

1.2. Tipos de interfaces inteligentes.

Dentro de las interfaces inteligentes también se tienen tipos que las distinguen unas de otras, los más marcados son las que están basadas en hardware y las basadas en software, pero esta es una clasificación muy general ya que únicamente se toma en cuenta el ambiente donde se desenvuelve. En una clasificación más minuciosa entre estas interfaces se pueden tomar en cuenta sus objetivos, contextos, y otras características que las distinguen unas de otras [12].

Dentro de esta clasificación se pueden encontrar las llamadas plantillas, en las cuales el usuario va llenando datos dentro del sistema para finalmente tener una salida adecuada considerando la información que el usuario proporcionó al sistema, de esta manera se forma un documento, sitio web, o hasta una aplicación que se comportará de acuerdo a lo que el usuario le pidió al sistema.

En la figura 1.2. se puede ver parte de una interfaz de este tipo en la cual le ayuda al usuario a construir una base de datos para un propósito en

particular. En dicho ejemplo se puede observar que las primeras tres características mencionadas para las interfaces inteligentes están plasmadas en la aplicación, ya que este tipo de interfaces que contienen algunos programas facilita el uso del sistema y disminuye el grado de conocimientos que debe tener el usuario para utilizarlo, además de que muestra la información de las funciones que puede realizar de una manera clara y concisa, por último se puede ver que todos los elementos que intervienen en la aplicación tienen acceso a la información que necesitan para la creación de la base de datos.

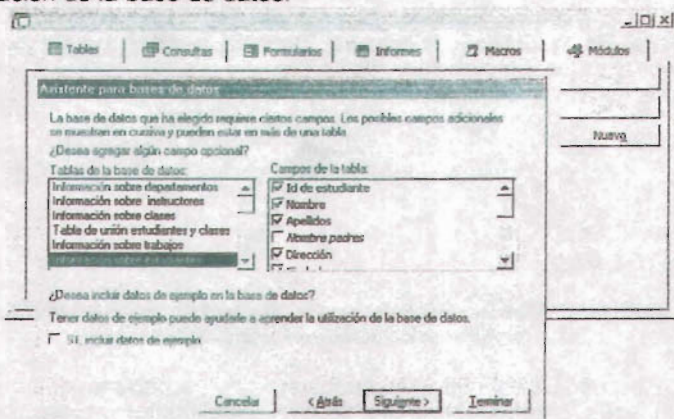


Figura 1.2. Interfaz inteligente para la creación de bases de datos. Interfaz perteneciente al programa Access, del conjunto de aplicaciones Office 97, propiedad de Microsoft.

Otro tipo de interfaces son los tutores dentro de las aplicaciones, éstos pueden resolver problemas sencillos o dar información de la aplicación o resolución de problemas más complicados. Estos se pueden ver con mayor frecuencia en los programas que integran el paquete Office de Microsoft, y aunque sencillos son considerados como interfaces inteligentes debido a la ayuda prestada y la forma de presentarla y procesar las peticiones de información que puede solicitar el usuario. En la figura 1.3. se muestra una interfaz de este tipo perteneciente a uno de los programas mencionados.

Aunque en su forma se ve simple, el poder de procesamiento y búsqueda de datos de la aplicación que está detrás es bastante bueno, además de que puede brindar una ayuda muy eficaz para el usuario. Los sistemas de este tipo no siempre son de esta forma, ya que existen otros que son considerados sistemas de realidad virtual donde un agente virtual puede ayudar al usuario a desarrollar tareas dentro de un entorno virtual como el programa Steve [3] que muestra a los alumnos a manipular la maquinaria que existe en un barco, este sistema se muestra en la figura 1.4.

Otro tipo de interfaces inteligentes son los agentes inteligentes que se utilizan en las aplicaciones de realidad virtual, estos pueden ser como el mostrado en la figura 1.4, y aunque también caiga en la categoría de tutor inteligente, no deja de ser un agente virtual. Dentro de éstos agentes se tienen

otros de los cuales forman parte los agentes conversacionales, estos últimos se explican en capítulos posteriores.

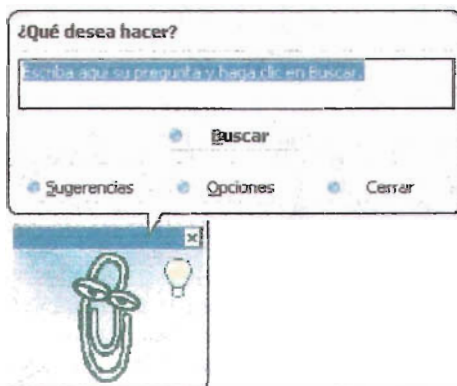


Figura 1.3. Tutor inteligente, Integrado a la aplicación Word del conjunto de aplicaciones Office 97 propiedad de Microsoft.

Existen otros tipos de interfaces inteligentes, como por ejemplo dentro del hardware, algunos de los dispositivos utilizados en realidad virtual retroalimentan al usuario por medio de sus sentidos con información acerca del entorno virtual donde se realiza la inmersión del usuario. Por ejemplo, algunos guantes regresan señales para emular que el usuario está tocando

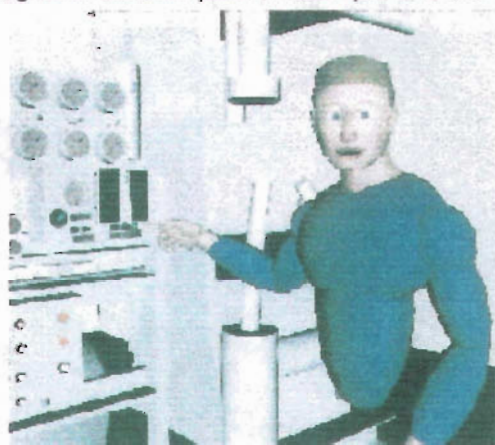


Figura 1.4. Aplicación de un tutor inteligente con realidad virtual [3], un objeto que sólo existe en el ambiente virtual.

También se pueden mencionar los chips llamados inteligentes los cuales pueden llevar el manejo de las transacciones de su dueño, además de que pueden ser utilizados de diversas formas, ya sea como tarjeta de débito, telefónica u otras utilidades que últimamente se les están dando.

Aunque ese tipo de interfaces no es de gran relevancia para el desarrollo de este trabajo, si es importante saber que existen, ya que pueden convivir con el tipo de sistema que se presenta en el presente trabajo.

1.3 Aplicaciones.

El número de aplicaciones que pueden tener el conjunto de interfaces inteligentes es muy grande, sólo depende de la forma de ver como aplicarlas a las necesidades de los usuarios. Se pueden tener interfaces aplicadas a los problemas ya descritos, como las plantillas utilizadas en diversas aplicaciones, o los tutores donde también se tienen varias.

Pero el tipo de interfaz importante para este trabajo son los agentes inteligentes, los cuales pueden ser aplicados de distintas formas en muchos contextos. Uno de los contextos ya mencionados es el de los tutores virtuales, donde son muy útiles para poder realizar los procesos de una manera correcta apoyándose de estos agentes para corregir errores o conocer nuevas posibilidades.

Una de las aplicaciones que ya han sido desarrolladas utilizando un agente inteligente es REA [3], la cual se muestra en la figura 1.5. El objetivo de este agente inteligente es mostrar casas para poder venderlas, donde su grado de interacción asemeja las acciones cotidianas de los humanos, de tal forma que incrementa su realismo y permite a los usuarios obtener de una forma sencilla la información que buscan acerca del contexto que maneja.



Figura 1.5. Agente inteligente para la venta de casas [3].

Dicho sistema además de mostrar la casa por fuera, también puede mostrar sus interiores, aumentando así el número de respuestas distintas que puede generar, y la información que puede proporcionar a los usuarios. Este es un buen ejemplo de una interfaz inteligente, ya que cumple con los objetivos que persiguen.

Aquí se puede observar además la tendencia que tienen las interfaces últimamente, la cual es hacer que se pueda realizar la interacción con el usuario de forma más natural para él, lo cual implica que la forma de interactuar con el sistema sea realizando actos simples como mover objetos o incluso comunicarse verbalmente como se muestra en esta última aplicación. También, aunque no se preste mucha atención este tipo de movimientos, en la interacción con un agente inteligente, da mucha vista poder ver que mueve la geometría asociada a él (conocida como avatar), y si asemeja a un humano, los movimientos corporales del avatar pueden agregar información a las respuestas que este le proporcione. Esto último se puede ver en el mismo sistema REA, ya que el agente puede hacer un ademán que los humanos identifican como que el objeto al cual se está refiriendo tiene un gran tamaño, esto lo logra alargando los brazos un arriba y otro abajo.

De esta forma se puede ver la importancia del desarrollo de interfaces inteligentes para ayudar a los usuarios a llevar a cabo sus tareas, o poder interactuar de mejor manera con los sistemas computacionales. Además al bajar el nivel de complejidad para el uso de este tipo de interfaces, disminuyen por tanto el tiempo de formación necesario para poder aprender a utilizar el sistema, y en algunos casos no se elimina dicha formación ya que la forma de interactuar con esos sistemas es muy natural para el usuario hasta llegar al punto que pueden llegar a parecerse a las actividades de su vida cotidiana (o al menos ese es el objetivo).

CAPÍTULO 2

AGENTES CONVERSACIONALES

2.1. Agentes virtuales.

Como se mencionó en el capítulo anterior, los distintos sistemas de realidad virtual utilizan un conjunto de interfaces muy variado, en donde se pueden encontrar dispositivos comunes como el ratón o el teclado, pero también se tienen dispositivos menos comunes en los sistemas computacionales, tales como los HMD (Head Mounted Display), o las CAVES. Otro tipo de interfaz es las que presentan los sistemas sin tener salidas físicas, tal es el caso de las GUI (Graphic User Interface), pero en los sistemas de realidad virtual pueden mencionarse los agentes virtuales.

Los agentes virtuales son una herramienta muy popular en el contexto de la realidad virtual, estos son utilizados para diversos propósitos. Una de sus características primordiales es que existen únicamente en la computadora, es decir, es una aplicación que no tiene parte física, por lo mismo se le denomina virtual. Otra característica es su representación dentro del ambiente virtual, la forma del agente tiene que ver con el objetivo para el cual se creó, ya que en algunos de los sistemas se requiere de mayor realismo y en otros simplemente que presenten los resultados de las operaciones que realiza.

Su representación, como ya se dijo, no necesariamente debe ser gráfica en un sistema de despliegue, sino que basta con presentar su resultado, por lo tanto no están encapsulados a utilizarse únicamente dentro de una computadora, por el contrario, también se están utilizando para guiar robots dentro de un entorno real, además de controlar algunas de sus acciones, como levantar objetos o manipularlos. Esto no quiere decir que el agente virtual sea el robot, sino que el robot utiliza un agente virtual, que como se mencionó es una aplicación, en este caso funcionando ya sea en el procesador del robot o en una computadora que le manda sus instrucciones al robot por medio de señales.

Esa es una forma de utilizar agentes virtuales, donde sale a relucir que el agente tiene cierta inteligencia, ya que debe ejecutar una tarea en específico cuando ocurra un evento previamente considerado, como que se presente un obstáculo inesperado en su camino. El agente internamente resuelve el problema presentado y manda las instrucciones necesarias para resolverlo, por lo que tuvo que ser programado con métodos de inteligencia artificial que resuelvan ese tipo de situaciones. Entonces los agentes pueden tener cierto grado de inteligencia gracias a la rama de la inteligencia artificial.

De esta forma tenemos los agentes inteligentes, que a diferencia de los primeros agentes virtuales, la interacción con el usuario, así como los procesos que pueden realizar son más complejos logrando un ambiente más amigable. Esto ha ayudado a que los sistemas de realidad virtual evolucionen y presenten mejores formas de interacción de tal manera que el usuario realice acciones de su vida cotidiana, las cuales se interpretan de cierta forma en el ambiente virtual dando un resultado similar en el sistema virtual al que se daría en el mundo real, por ejemplo, el tomar un bloque con la mano y trasladarlo a otra posición dentro del ambiente presentado.

Con los grados de interacción que están presentando los sistemas de realidad virtual, se necesita tener un intermediario entre estos y los usuarios, debido a que los humanos pueden llegar a inhibirse al estar trabajando con una máquina que tenga tal nivel de interacción, y por consecuencia disminuiría su confianza hacia el sistema. Al presentarse una representación gráfica familiar a las acciones que realice el usuario, este mostraría mayor confianza y se podría desenvolver de una mejor manera, además la presentación de un agente virtual busca el objetivo de que el usuario no se sienta solo dentro del ambiente virtual, y en ocasiones tenga una guía para evitar errores o incluso para aprender el manejo de alguna maquinaria u otro tipo de aparatos.

Así se ve la importancia de utilizar agentes virtuales, ya que además de amenizar el sistema, pueden proporcionar ayuda y permiten que el número de usuarios sea mayor, debido a que son menos los conocimientos necesarios para utilizar los sistemas de realidad virtual que los contengan. Por lo que se ha mencionado se tienen distintos tipos de agentes virtuales, divididos según las propiedades que tienen, donde cada uno se utiliza para distintos propósitos. Los tipos de agentes virtuales están clasificados como sigue, aunque hay que tener en cuenta que estos pueden incluir otra clasificación dentro de ellos mismos.

- Agentes estáticos.
- Agentes representativos.
- Agentes inteligentes.

Los agentes estáticos son agentes virtuales que simplemente se agregan al entorno virtual para, por así decirlo, adornar el ambiente y aumentar el realismo, ya que no es lo mismo navegar en una ciudad virtual sin personas que con avatares simulando a éstas, aún cuando no presenten ningún movimiento. Se puede notar que este tipo de agentes son los más fáciles de ingresar dentro de un ambiente virtual, ya que únicamente consta de su modelado y ubicarlo dentro del entorno, lo que actualmente no representa mucho esfuerzo para el diseño y programación, aunque también representa poca ayuda al sistema.

Aun siendo tan simples como se ven hoy en día, son muy importantes debido a que son el inicio de los agentes virtuales, y es el primer paso casi obligado para poder empezar a desarrollar ambientes con agentes virtuales. Esto no quiere decir que sean imprescindibles en un sistema de realidad virtual que contenga agentes virtuales, por el contrario, los sistemas más actuales los sustituyen utilizando en cambio agentes inteligentes, pero son la base para llegar a estos últimos, ya que todo los agentes virtuales pueden considerarse como agentes estáticos con algún proceso añadido.

Los agentes representativos son utilizados para ubicar al usuario dentro del entorno virtual, es decir, son agentes que representan al usuario en el ambiente, ya sea para otros usuarios o como referencia para el mismo. Estos surgen como una necesidad, ya que el usuario podría desubicarse dentro del entorno virtual al no ver donde estaba exactamente, y aunque se hace una

analogía de la posición del usuario con la posición de la cámara, el efecto tiende a marear y no da una idea completa de donde está parado el usuario.

Otra de las necesidades de utilizar agentes representativos, es en sistemas donde se enlazan dos o más usuarios para trabajar en el mismo ambiente, esta es una forma de conocer la posición de los otros usuarios. Es aquí donde se ve la importancia de este tipo de agentes, ya que sin ellos sería muy difícil trabajar en entornos virtuales colaboratorios como los laboratorios virtuales. Y aunque en la mayoría de los sistemas colaboratorios no es necesario ver a los otros usuarios, o una representación virtual de éstos, hay algunos donde es imprescindible (como los ya mencionados), debido a que una parte importante del objetivo de dichos sistemas puede ser el conocer la posición de cada usuario, ya sea para su ubicación dentro de un ambiente real o simplemente para poder localizarlos.

Como se ha mencionado, los agentes estáticos no realizan ninguna acción, simplemente se quedan en una posición en específico, y los agentes representativos van a realizar exactamente lo que el usuario indica (con sus respectivas restricciones) por lo que el agente no tiene autonomía. Cuando un agente realiza una o varias acciones por sí solo sin que el usuario se lo pida, se puede considerar que tiene una cierta autonomía, pero por el simple hecho de estar moviendo alguna parte de su geometría representativa, no quiere decir que pueda ser considerado como un agente inteligente.

Como agentes inteligentes se toman aquellos avatares que pueden realizar un proceso mental no tan simple, es decir, no con el simple hecho de mover su geometría o cambiarla de posición se puede considerar como agente inteligente. Para poder ser considerado como tal debe realizar procesos que no sean tan triviales, como el cálculo de una trayectoria de un punto a otro dentro de un cuarto con objetos en su interior, en este caso el agente debe de procesar todos los datos que pueda obtener de su entorno y trazar una ruta que lo lleve desde el punto inicial hasta el punto final sin atravesar los obstáculos mencionados. En dicho ejemplo se muestra que el agente debe realizar un proceso no tan trivial para una máquina, ya que no es lo mismo mover a un agente en línea recta atravesando varios obstáculos a que el agente mismo resuelva la forma de librar dichos obstáculos y llegar al punto objetivo sin atravesar ninguno.

Es aquí donde se ve que el agente llega a tener cierto grado de inteligencia, por lo que es considerado como un agente inteligente. La forma en que este calcula la trayectoria es por medio de algoritmos propios del área de la inteligencia artificial, y es considerada como inteligencia artificial por que puede resolver dicho problema con distintos cuartos, posición, tipo y número de objetos sin intervención humana. Y es verdad que un humano escribió dicho algoritmo, pero a la hora de resolver el problema, ningún humano, aún el que lo programo, no interviene en el proceso de la búsqueda de la trayectoria.

Pero no todos los agentes inteligentes calculan simplemente rutas, sino que pueden tener distintos procesos para resolver varios tipos de problemas o circunstancias que van apareciendo. Dentro de dichas circunstancias puede

estar el seguir al usuario, o darle orientación cuando se detecte que está cometiendo un error, etc. Por lo que un agente inteligente puede resolver más de un sólo problema, por lo que su complejidad puede alcanzar niveles muy altos.

Un caso especial de los agentes inteligentes son los agentes conversacionales, donde su característica primordial es que pueden interactuar con el usuario utilizando un lenguaje, por lo que se puede entablar una conversación entre los dos, el cual puede ser tan general como el sistema lo permita, como se discute en [3] y [9]. Estos agentes amenizan de gran forma la interacción entre el sistema y el usuario, ya que al poder procesar una parte del lenguaje (ya que estos se desenvuelven dentro de un contexto), le es más familiar utilizar el sistema debido a que realiza una actividad cotidiana la cual es la comunicación por medio de la conversación.

2.2. Conversación.

Para los agentes conversacionales, la comunicación con el usuario es primordial, por tanto, para lograr este nivel de interacción se deben conocer las distintas partes que pueden componer una conversación, ya que no simplemente se conforma de un conjunto de frases o diálogos. En primer lugar se conoce que para lograr una comunicación se deben tener un emisor y un receptor, que en el caso de una conversación, dos sujetos toman dichos papeles alternándose por turnos.

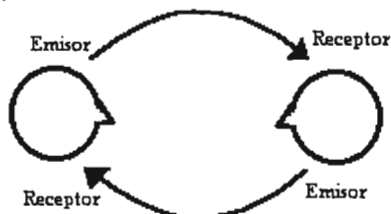


Figura 2.1. Esquema general de una conversación entre dos sujetos

El emisor se encarga de mandar mensajes con cierta información a través de diálogos y/o movimientos corporales, los cuales se enfocan a un tema en particular. El receptor se encarga de entender dichos mensajes, además puede retroalimentar el flujo de la conversación, dando ciertas señales ya sean verbales o corporales, para indicar el seguimiento del tema, o al querer iniciar una interrupción ya sea para cambiar los papeles o indicar que no se comprendió el mensaje recibido.

Pero para que una comunicación pueda ser una conversación se necesita que el emisor y el receptor intercambien papeles, de tal forma que los dos realicen ambas funciones alternadamente como se menciono anteriormente. Entonces el sujeto 1 comienza a dar un mensaje al sujeto 2, por lo que el primero resulta ser el emisor y el segundo el receptor, al terminar su diálogo, el sujeto 2 da una respuesta por medio de otro diálogo por lo que se intercambian papeles, de esta forma sucesivamente la conversación va tomando su rumbo.

Los turnos son un componente muy importante como se menciona en [2] y [3], ya que brinda la oportunidad de llevar el control de la conversación, siendo que al conocer el turno de cada sujeto se puede saber qué función está realizando y qué acciones puede o no puede realizar. Estas acciones están enfocadas a la comunicación, aunque no significa que forzosamente sean verbales, ya que existen acciones o comportamientos dentro de la comunicación que involucran movimientos corporales o gestuales sin utilizar alguna mención verbal.

Este conjunto de acciones que existen dentro de la conversación es conocido como comportamientos de la comunicación, los cuales se pueden llegar a dividir entre comportamientos verbales y no verbales ([3] y [6]). Estos conllevan la guía del estado de la conversación, lo que permite saber si el receptor está entendiendo lo que el emisor le está comunicando, o si el emisor terminó su diálogo y es momento de cambiar turnos, o si el receptor quiere cambiar turnos repentinamente ya sea para corregir o agregar información, o algún otro motivo que afecte el curso de la conversación.

Los comportamientos no verbales pueden dividirse entre los gestuales y los corporales, donde los primeros se refieren a los movimientos dentro de la cara, como puede ser el levantar las cejas o el hacer una mueca. Estos pueden expresar emociones y agregar información a la conversación, ya que con estos movimientos faciales, el emisor puede darse cuenta de que el receptor entiende o no lo que le está comunicando, o si la información le fue grata entre otras cosas.

Los movimientos corporales son aquellos que utilizan una parte del cuerpo distinta a la cara, con esto también se toma en cuenta el movimiento de la cabeza. Estos pueden ser utilizados para agregar información más detallada tal como las dimensiones de un objeto, de la misma manera pueden ser utilizados para saber si el receptor comprende lo que el emisor le informa, un ejemplo de esto es el famoso movimiento de cabeza de atrás hacia delante indicando el seguimiento del tema.

De los comportamientos no verbales se tienen las respuestas cortas como el "sí" para dar a entender que va por buen camino la conversación, o el "esteee..", para indicar una pausa en lo que se arma correctamente la frase que se dirá. Estos comportamientos pueden entrar directamente en la frase a analizar, por tal motivo es recomendable que el agente tenga un sistema para reconocerlas y no interfieran en el análisis del diálogo del usuario, aunque esto depende de que tan robusto deba ser al sistema.

2.3. Arquitectura.

Los distintos factores mencionados que ocurren en la conversación, son tomados en cuenta para el diseño de agentes conversacionales, ya que como se ha mencionado, la conversación no consta únicamente del intercambio de diálogos hablados, sino que también son considerados los movimientos corporales como parte de ésta. Por tanto, tomando lo anterior en cuenta se

tienen una serie de módulos que pueden contener dichos agentes, y aunque existen distintas propuestas como en [6] o [8], los componentes considerados en el presente trabajo son:

- Procesamiento del lenguaje natural.
- Reconocimiento de voz.
- Síntesis de texto a voz.
- Animación del avatar.
- Captura de movimientos del usuario.

Donde el procesamiento del lenguaje natural (PLN) es su pieza más importante, debido a que sin este el agente no puede interpretar los diálogos del usuario y generar una respuesta a ellos. Los otros módulos pueden ser reemplazados, ya que el diálogo puede entablarse por medio de mensajes escritos y respuestas también escritas o acciones gráficas o físicas, con esto último se elimina la necesidad de asociar una geometría a un agente conversacional. En esencia, el PLN debe tomar el diálogo del usuario para desglosarlo y analizarlo con el objetivo de crear una respuesta adecuada, ya sea por medio de un diálogo, una serie de movimientos, o algún otro tipo de acción del sistema, o incluso una combinación de estas.

Para el procesamiento de las frases, el PLN se apoya en la base de conocimiento del agente, es aquí donde se tiene la definición de la parte del lenguaje que puede reconocer el agente. Dicha base de conocimiento debe contener la mayor parte de las palabras que se involucran en el contexto que se esté manejando, y necesariamente las palabras más comunes que se manejan dentro dicho contexto. Además, el agente debe saber como reaccionar a cada frase del usuario, lo cual también debe estar integrado en su base de conocimiento.

Dicha base comúnmente es un conjunto de archivos que contienen los datos necesarios organizados de cierta forma para que el sistema los pueda entender. El conjunto de dichos archivos puede ser considerado como una base de datos, donde el sistema contiene el administrador de esta, obteniendo los datos necesarios en el momento indicado, desglosándolos para cada módulo en particular. Como consecuencia, a dicho conjunto de archivos se les conoce como base de datos de conocimiento del agente.

Al tener el PLN del agente conversacional, pueden añadirse ciertos procesos o características para aumentar el realismo del sistema, como es el caso de la síntesis de voz. Dicha síntesis está dividida en dos partes fundamentales: el reconocimiento de voz y la síntesis de texto a voz. La primera se encarga de transformar el sonido analógico de la voz del usuario para poderla tener en medios digitales, con la cual se realizan ciertos procesos donde se pasa dicha onda de sonido digitalizada a una cadena de texto también en medios digitales. Esta última es la que se da al PLN, ya que es la misma frase del usuario pero en medios digitales.

Al generar una respuesta verbal, el módulo del PLN produce también una cadena de texto la cual obviamente está en medios digitales, por lo que se

debe hacer el proceso inverso al reconocimiento de voz. El módulo que realiza este proceso es la síntesis de texto a voz, y como su nombre lo indica, toma como entrada una cadena de texto, la procesa, y obtiene la frase en un sonido digitalizado (comúnmente un archivo de audio), este debe ser reproducido por el sistema como un sonido analógico para que el usuario lo entienda, y será el sistema mismo quien se encargue de sincronizar dicho audio con los movimientos del avatar.

Estos movimientos mencionados son manejados por los módulos de animación del agente, pero su animación no es tan simple como se puede pensar. En el caso de que la geometría del avatar asemeje la forma humana, su animación puede dividirse en tres partes tomando en cuenta el objetivo que persigue cada una. La animación primordial se enfoca al movimiento de la boca, esta debe tomar en cuenta la entonación de cada una de las frases que puede llegar a responder el agente. Para este caso se tienen diversos métodos que se utilizan para la resolución de dicho problema, estos pueden enfocarse al movimiento de la boca por medio de posiciones pre-calculadas, o con base en un diseño muscular, donde se le aplica a un músculo un cierto grado de fuerza el cual genera el movimiento de una parte de la geometría influenciada por dicho músculo.

También pueden ser considerados otros factores como la forma en que se deforma la boca al decir una letra, sílaba o frase [1]. En esta parte debe considerarse que el movimiento se vea de manera más natural pero ocupa un espacio mayor en memoria, o tener una animación entrecortada y por tanto que se vea muy artificial pero sin ocupar tanto espacio en memoria. Y aunque la animación de la boca es la parte primordial del movimiento del avatar, también pueden ser animadas otras regiones de la geometría de la cara agregando mayor realismo al avatar.

Al mover estas otras regiones de la cara se pueden crear gestos, lo que aumenta el realismo del avatar y genera otro tipo de posibilidades para el sistema, como el poner estado de ánimo al agente, como se puede ver en [1] o [8]. Para poder modificar la geometría de forma que muestre distintos gestos, se pueden utilizar los mismos métodos mencionados para la animación de la boca, pero las regiones de influencia se extienden a toda la cara.

Al aplicar modificaciones a la cara para que realice algún gesto, se debe tomar en cuenta que puede llegar a afectar directamente en el movimiento de la boca, por lo que es muy recomendable utilizar el mismo modelo para los dos módulos de animación. De esta manera se asegura que la geometría no tendrá comportamientos extraños al realizar las dos animaciones al mismo tiempo, lo cual abre una gran gama de posibilidades donde el agente podrá decir las distintas respuestas verbales con gestos diferentes.

El último tipo de animación que puede ser agregado al avatar es en el cuerpo, ya que si se presenta al avatar con una parte de su cuerpo, se ve muy raro que la cara tenga animación pero el resto del cuerpo permanezca estático. En una conversación, los humanos no dejan su cuerpo estático, al contrario, se ayudan de su cuerpo para enriquecer la comunicación, como ya se mencionó,

a esto se le llama lenguaje no verbal, pero no es exclusivo para los humanos, también puede ser añadido este tipo de acciones al avatar por medio de la animación corporal.

Para realizar este tipo de animación se tienen distintas técnicas, donde una de las más concurridas es por medio de la definición de un esqueleto, el cual tiene asociado una cierta jerarquía, y cada una de las partes de dicha jerarquía representa un hueso, donde cada uno de estos tiene asociado una parte de la geometría del avatar la cual se moverá dependiendo del movimiento que realice dicho hueso.

Existen más técnicas para la animación corporal de un avatar, pero todas se concentran en mover cierta parte de la geometría que lo representa de una forma adecuada a la respuesta que está generando el agente. Como ya se mencionó, dicha animación agrega información a la comunicación entre el agente y el avatar, pero además aumenta el realismo de este último.

Pero la comunicación no verbal en un sistema de realidad virtual con agentes conversacionales no queda sólo de parte del agente, también el usuario puede aportar este tipo de comunicación. Las técnicas utilizadas en este caso son de visión computacional, y pueden hacerse en varios niveles, ya sea que se enfoque a los movimientos corporales del usuario por medio de cámaras o captura de movimiento, o que se enfoque al rostro del usuario (donde también se cuenta con varias técnicas). Al digitalizar los movimientos corporales y/o gestuales del usuario, el sistema debe poder interpretarlos para agregar dicha información a la conversación verbal que están entablando, y aunque este tipo de interacción no es estrictamente necesario para tener un agente conversacional, si aumenta el realismo y la naturalidad con que el usuario puede llegar a interactuar con el sistema.

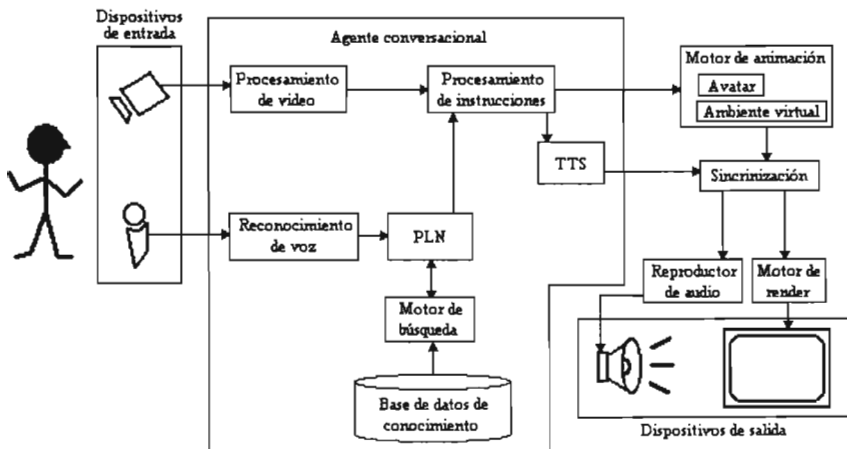


Figura 2.2. Arquitectura de los agentes conversacionales. TTS = Text To Speech, o síntesis de texto a voz, PLN = Procesamiento de lenguaje natural.

Con este conjunto de módulos, se puede obtener un diagrama que muestre la parte y la forma en que están involucrados, así como el flujo de la

información y el procesamiento que va por el sistema a través de dichos módulos. El diagrama se muestra en la figura 2.2., en esta se puede observar las distintas partes que puede integrar un agente conversacional.

Hay que recordar que no todos los módulos expuestos en la figura 2.2. son estrictamente necesarios para integrar un agente conversacional, como por ejemplo la parte de visión computacional es simplemente para aumentar el realismo y ampliar la forma de interacción del usuario hacia el sistema. Pero no solamente se muestran módulos que podrían eliminarse, sino que hay otros que se pueden llegar a reemplazar por otro parecido sin alterar la condición de agente conversacional, por ejemplo se mencionó que un agente puede prescindir de geometría representativa y en cambio mostrar su respuesta con algo físico como un robot que calcule una ruta dentro de un cuarto con obstáculos, en este caso se reemplazaría el dispositivo de despliegue por el robot, dando así otro tipo de salida a un sistema que también maneje un agente conversacional.

En el diagrama se pueden observar también otros módulos que no se habían contemplado anteriormente, estos simplemente corresponden al manejo de la salida para que los medios en que se presente pueda interpretarlos correctamente. Un módulo importante que se debe tomar en cuenta cuando se tienen más de una forma de presentar la salida es el de la sincronización, este se encarga de que las varias salidas se muestren de forma tal que ninguna de estas salidas se vea desfasada con respecto de las otras.

Para la implementación de estos agentes, se pueden utilizar distintas herramientas y/o métodos que resuelven cada módulo por separado. Algunos de estos son descritos en los siguientes dos capítulos, los cuales se utilizaron o modificaron para el diseño de la metodología descrita en este trabajo.

CAPÍTULO 3

PROCESAMIENTO DEL LENGUAJE NATURAL

Una parte importante para los agentes conversacionales es su interfaz basada en dialogo, la cual consta de un conjunto de módulos para procesar las frases del usuario y entregar una respuesta hablada por parte del agente. Dentro de ellos se encuentra el procesador de lenguaje natural, el cual se encarga de darle una interpretación lógica para la aplicación a las frases del usuario, y entregar una respuesta adecuada a cada una de estas. Este procesamiento es automático, por lo que la aplicación realiza internamente la interpretación de las frases.

Se debe notar que dicho procesamiento trabaja sobre la frase previamente digitalizada, es decir, sobre una cadena de caracteres, además de que su salida son un conjunto de instrucciones que debe interpretar la misma aplicación, y una cadena de caracteres, que será la respuesta que debe dar el agente. Para digitalizar la voz del usuario y generar la voz del agente, se debe contar con programas de síntesis de voz que realicen dichos procesos. Estos se revisan en un capítulo posterior.

Dejando la síntesis de voz a otro tipo de aplicaciones, el procesador se puede concentrar en analizar y comprender las frases del usuario. Para ello cuenta con tres partes fundamentales para el procesamiento de cualquier lenguaje [10]:

- Palabras.
- Sintaxis.
- Semántica.

Estos revisan si las palabras utilizadas por el usuario se encuentran en el conjunto de palabras que tiene el agente en su base de conocimientos; que la estructura de la frase sea la correcta; y el significado de dicha frase, respectivamente. Su resultado debe ser entregado a la aplicación de forma que ésta pueda entenderla, y por consiguiente ejecute de manera correcta lo indicado por el usuario.

Para estos propósitos se cuenta con un conjunto de técnicas computacionales comúnmente utilizadas en el campo de la inteligencia artificial, cada una de ellas puede ser utilizada en las tres partes antes mencionadas, aplicadas de cierta forma dependiendo de que parte se trate. Por tal motivo se trata cada una de dichas partes por separado, mostrando que es lo que abarcan, las técnicas utilizadas en ellas y la forma en que se aplican.

3.1. Palabras.

Dentro del lenguaje natural, las palabras son las unidades léxicas que utilizamos para armar una frase. Estas están compuestas de una o más letras pertenecientes a nuestro alfabeto (conjunto de símbolos que conocemos), y son las que conforman nuestro vocabulario. Este último forma parte de la base de conocimiento del agente, por lo tanto, entre más palabras contenga, mayor alcance tendrá la interacción con el agente.

Como se puede observar existe un elemento que no tiene un padre, por lo que es el único que se encuentra en el primer nivel, a este se le conoce como raíz el cual es único dentro de este tipo de estructuras, este sirve como referencia para realizar las operaciones. Los otros elementos que no tienen hijos son conocidos como hojas, y no necesariamente están en el último nivel, amenos de que las reglas para el árbol lo exijan.

Existen varios tipos de árboles, cada uno con su propio conjunto de reglas que los hacen distintos a los demás, y les dan ciertas propiedades que mejoran su desempeño en algunas de las operaciones como la búsqueda de algún elemento. De entre los distintos tipos de árboles se encuentran el binario, rojo-negro, equilibrado, AVL, entre otros.

La forma más básica de este tipo de estructuras es el árbol binario, donde su característica primordial es que cada elemento tiene un máximo de dos hijos. De esta forma se simplifica su implementación, ya que cada uno de los elementos del árbol debe tener un número finito de hijos o simplemente no tener hijos. Esta forma de implementar los árboles también simplifica el modo en que se recorren todos sus elementos, además que disminuye el tiempo promedio que consume la búsqueda de un elemento en caso de que estén ordenados.

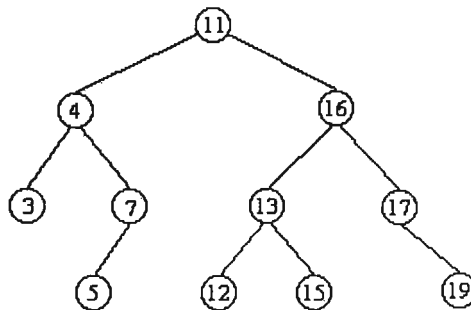


Figura 3.2. Árbol binario de datos numéricos.

Al tener un orden aleatorio, se tiene el mismo problema que con las listas, donde se tienen que ir revisando todos los elementos hasta llegar al que se quiere localizar o terminar sin encontrarlo. Pero si la estructura está ordenada, la búsqueda simplemente recorrerá un camino sin pasar innecesariamente por varios elementos del conjunto. Por ejemplo, en la figura 3.2 se muestra un árbol de números, todos están ordenados del menor al mayor, por lo que al buscar un número simplemente se verifica si es mayor o menor al elemento actual, pasando al elemento de la derecha o de la izquierda según sea el caso, por lo que al buscar el número 14 únicamente se revisarán cuatro números de once que comprenden todo el árbol, estos son el 11, 16, 13 y 15, y ya que ninguno de estos es el 14 y no hay más elementos que cumplan con la condición necesaria (que el 15 tenga un hijo a la izquierda debido a que el 14 es menor), se puede decir con certeza que dicho número no existe en el conjunto, en cambio, si se busca el número 7, se revisan los números 11, 4 y 7,

y como este último es el requerido, se finaliza la búsqueda con éxito, aun cuando dicho elemento no sea una hoja.

Otra técnica utilizada para el procesamiento de palabras son las expresiones regulares. Estas realizan una búsqueda dentro de un conjunto de datos en base a una instrucción con las indicaciones necesarias utilizando un conjunto de símbolos, cada uno de estos proporciona un significado distinto a las búsquedas. A continuación se listan algunos de los símbolos utilizados en las expresiones regulares:

- *: Significa que pueden localizarse ninguno, uno o varios caracteres en la parte que se utilice.
- +: Pueden estar uno o varios caracteres donde se utilice.
- []: Encierran un carácter o un conjunto de caracteres para indicar que puede ir cualquiera de ellos.
- (): Se utilizan para agrupar una expresión regular o un conjunto de caracteres.
- |: Se utiliza para buscar más de un patrón a la vez, es utilizado como el operador binario OR.
- -: Sirve para indicar una serie que inicia con el carácter anterior al símbolo y termina con el posterior.
- ^: Es utilizado para negar algún carácter.

Con esto, si se tiene el siguiente conjunto de palabras:

Hola	Dulce	Camión	Reloj
Hongo	Ventana	Termodinámica	Refresco
Esdrújula	Coladera	Película	Tornillo
Vaso	Comadreja	Colorante	Martillo
Caricatura	Murciélago	Pasillo	Mover
Vídeo	Elefante	Lechuga	Donde
Caramelo	Mostrar	Licuada	Encontrar

Tabla 3.1: Conjunto de palabras de ejemplo.

Se pueden realizar varios tipos de búsquedas utilizando las expresiones regulares, por ejemplo, para encontrar todas las palabras que comiencen con "L", la instrucción deberá ser:

L*

Con esto le indicamos que la primer letra o carácter es la "L", y que deberá estar seguida de ninguna, una o más letras distintas entre sí, con lo que el resultado será:

- Lechuga
- Licuado

En este caso el resultado es muy corto, pero en el caso de un conjunto de datos mayor pudo haber incrementado el número de palabras significativamente, por lo cual se puede refinar la búsqueda, ya sea utilizando

otro símbolo, o reutilizando el mismo junto con otra palabra. Por ejemplo, si se quieren todas las palabras que empiecen con M y que contenga una g en cualquier lugar de la palabra, la instrucción debe ser:

M*g*

Esta instrucción realizará una búsqueda de todas las palabras que comiencen con M, y contengan al menos una g en cualquier parte de ella, debido a los dos asteriscos colocados antes y después de dicha palabra, lo cual le da la propiedad de tener cualquier número de letras entre la M y ella, lo mismo que después de ella. La palabra que concuerda con dicha búsqueda es:

- Murciélago

Como se puede observar, en esta ocasión se realizó una depuración mayor del conjunto de palabras, debido a que hay más palabras que comienzan con M (como Mover), pero solo esta contiene una g.

Como se menciona, una instrucción puede contener varios tipos de símbolos para realizar una búsqueda más robusta. Por ejemplo, para buscar palabras que comiencen con T o E en una sola instrucción, se deberá poner:

[ET]*

En esta instrucción se utilizan tres símbolos, los paréntesis para agrupar la E y la T, haciendo que el operador OR se aplique únicamente a estas dos letras, ya que sin ellos se realizaría la búsqueda sobre la E y la T*, con lo que se estaría buscando las palabras que sean solo una E o que comiencen con T. Como ya se vio el operador binario OR le pide a la búsqueda que le entregue las dos opciones, y el asterisco afuera permite que las palabras cumplan sólo con la condición de comenzar con cualquiera de esas dos letras. El resultado de dicha búsqueda es:

- Esdrújula
- Elefante
- Termodinámica
- Tornillo
- Encontrar

También pueden utilizarse expresiones regulares dentro de otras más grandes, por ejemplo, al realizar una búsqueda de las palabras que inicien con E y que no terminen en vocal, se puede utilizar la siguiente instrucción:

E*^[aeiou]

El símbolo ^ niega el carácter que le precede, pero como es otro símbolo, negará lo que represente, que indica cualquier vocal por lo que al unirlos no permite que exista una vocal en esa parte. Además de la unión de estos símbolos que ya forman por sí solos una expresión regular, se tiene el resto de la instrucción que es otra expresión regular, y que busca las palabras

que comiencen con E. En conjunto, dicha instrucción arrojará el siguiente resultado:

- Encontrar

El resto de las palabras que inician con E no aparecen debido a que la negación no permite que terminen en vocal, y aunque esta palabra contiene vocales, su última letra es una consonante.

Como puede observar, las expresiones regulares pueden llegar a ser muy útiles para un tipo de aplicaciones donde se necesita encontrar un conjunto de palabras con ciertas características. Por esta razón son utilizadas en varios sistemas, como por ejemplo para realizar operaciones con los comandos del sistema operativo UNIX o Linux, también están presentes en lenguajes como PERL, que las utiliza para realizar búsquedas dentro de bases de datos o en un documento.

3.2. Sintaxis.

La búsqueda con expresiones regulares ayuda a obtener algunas palabras con cierta característica dentro de un conjunto mayor, pero para el PLN no es suficiente, ya que el tener sólo un conjunto de palabras no es de mucha utilidad para saber qué se está tratando de decir. Pero antes de descifrar la frase del usuario, se debe ver si la forma en que la dijo es la correcta, ya que en el lenguaje se tienen estructuras que deben seguirse para que las frases tengan congruencia.

Un método para revisar la sintaxis es mostrado en [2] y [10], el cual es realizarlo por medio de árboles, donde cada palabra se toma como una hoja de estos, y la raíz es la referencia principal de la frase. Para utilizar este método se dividen las palabras de acuerdo a su tipo o la relación que tengan en su contexto, los tipos que generalmente se utilizan son verbos, adjetivos, objetos, entre otros, aunque pueden variar el número y los tipos utilizados dependiendo del contexto de la aplicación que se esté realizando.

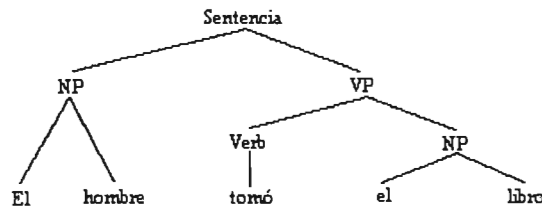


Figura 3.3. Árbol de gramáticas libres de contexto. NP = Sujeto, VP = Predicado, Verb = Verbo.

En la figura 3.3. se puede observar como se desglosa una frase que puede corresponder a un contexto dado en un árbol, este se va construyendo desglosando la frase en sus distintas partes que contiene, y estas a su vez se desglosan en las partes que las componen, hasta llegar a tener las palabras que contiene la frase. También se puede observar que cada uno de los nodos

del árbol que no es una palabra, corresponde a un tipo de palabra de acuerdo con su descripción.

Las descripciones de las palabras están basadas en la forma en que se utilizan dentro de las frases, esto depende directamente del lenguaje que se utilice, ya que no todos los idiomas formulan las mismas estructuras ni el mismo orden de las palabras para formar los enunciados. En el ejemplo se ve una frase en inglés la cual consta de dos partes principales, el sujeto y el predicado, el sujeto está compuesto por "el hombre", y es simbolizado por el tipo NP. El predicado en cambio se conforma de dos partes más, el verbo y otro sujeto, denotados con Verb y NP respectivamente, el verbo es "tomó" y el otro sujeto se refiere a "el libro" (ejemplo traducido de [10]).

Con dicha estructura se tiene que las gramáticas que se pueden generar están definidas como se muestra a continuación:

Sentencia -> NP VP
NP -> artículo objeto
 artículo persona
VP -> Verb NP
Verb -> verbo

Estas reglas pueden definir a una gramática libre del contexto, la cual se puede llevar a un autómata de estados finitos. En el ejemplo se puede ver que el árbol puede generarse siguiendo la raíz de este y caminando a través de las producciones que tiene definidas, esta es una forma de poder revisar la sintaxis de las frases, ya que se tomarán como válidas únicamente las que puedan concordar con los árboles que se puedan generar. Hay que notar que el árbol presentado es muy general y simple, pero se puede tener una gramática más elaborada en donde las producciones de las sentencias contengan más de un resultado, donde cada uno puede ser otra sentencia.

Como ya se mencionó, las gramáticas regulares pueden llegar a describir un autómata de estados finitos, de hecho son estos últimos los que se utilizan como método para programarlas. Sus transiciones pueden ser descritas de la misma forma, y aunque la sintaxis puede ser revisada analizando un árbol que describa una gramática libre de contexto ya sea en orden ascendente o descendente, los autómatas pueden mejorar dicha revisión, ya que por cada variación en la estructura se tiene un árbol, pero todos pueden ser descritos en un mismo autómata.



Figura 3.4. Relación entre autómatas finitos, expresiones regulares y lenguajes regulares [10].

Existe una relación entre las expresiones regulares las cuales pueden tener un mayor potencial en la búsqueda sobre documentos, pero también

pueden describir autómatas de estados finitos y viceversa [10]. Además de dicha relación, cualquiera es utilizado para la descripción de lenguajes regulares (que son un tipo de lenguajes formales). De aquí se tiene una correlación entre ellos, la cual está descrita en la figura 3.4.

Aunque las expresiones regulares trabajan en búsquedas sobre palabras y están relacionadas directamente con los autómatas finitos, estos últimos pueden trabajar con otros tipos de entradas, no se encierran en la búsqueda sobre palabras. Un autómata se compone de un conjunto de elementos relacionados entre sí para realizar transiciones de tal forma que cumpla con el objetivo para el cual fue diseñado. Los elementos que comúnmente conforman un autómata son:

- Alfabeto: Es el conjunto de símbolos que puede reconocer el autómata, estos pueden ser letras del alfabeto, o incluso puede tomarse una palabra completa como un símbolo.
- Estados: Los estados del autómata son los puntos donde inicia y termina una transición, cada estado está conectado a otro u otros de acuerdo las reglas de transición.
- Estado inicial: Es un estado como cualquier otro, con la diferencia de que es donde inicia el sistema, por lo que siempre debe existir. Pueden encontrarse transiciones que lleguen a este estado, y necesariamente debe existir al menos una transición que salga de este estado. Comúnmente se tiene un sólo estado inicial, pero hay autómatas que pueden tener más.
- Estados finales o de aceptación: Son los estados donde debe terminar el sistema, si el autómata termina en un estado que no esté definido dentro de este conjunto, entonces la entrada no es aceptada y se dice que no pertenece al lenguaje. En general se tienen uno o más estados finales, entre los cuales se permite tener al estado inicial. Este tipo de estados debe tener al menos una transición que llegue a ellos, y opcionalmente pueden salir transiciones de estos hacia otros estados.
- Reglas de transición: Son las reglas que describen las transiciones entre estados, tomando en cuenta el símbolo de entrada. Estas pueden ser descritas por medio de expresiones regulares o con una tabla de transiciones. La simbología de las expresiones regulares dentro de los autómatas es un poco distinta a la descrita anteriormente, donde los símbolos más utilizados son el * que indica ninguna o más ocurrencias del carácter anterior, y el + que indica una o más ocurrencias del carácter anterior.

Estos componentes generales se encuentran en todos los autómatas, donde la diferencia entre cada uno es lo que permiten o no permiten hacer. En el caso de los autómatas de estados finitos, estos deben tener un conjunto finito de estados y únicamente permiten un estado inicial, además de que cada estado tiene una sola transición por cada símbolo de entrada.

Además de la gráfica, hay otra forma para describir las transiciones del autómata, esta es por medio de una tabla de transiciones como la mostrada en la tabla 3.2, la cual describe al autómata de la figura 3.5.

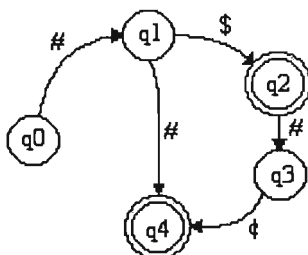


Figura 3.5. Autómata de estados finitos.

Estados	#	\$	c
q0	q1	0	0
q1	q4	q2	0
q2	q3	0	0
q3	0	0	q4
q4	0	0	0

Tabla 3.2: Tabla de transiciones del autómata de la figura 3.5.

Claramente se puede ver que para cada entrada existe solo una transición, que en ocasiones terminan en estados vacíos o inexistentes (denotados con 0) que significan que dicha entrada no es aceptada por el autómata. La única forma en que una entrada sea aceptada es que se llague al estado final (denotado con un doble círculo), y que ya no se tengan símbolos de entrada.

En el ejemplo de la figura 3.5. se muestra un autómata de estados finitos, el cual reconoce el conjunto de frases que dicen una cantidad de dinero, cada uno de los componentes que lo integran se explican a continuación, considerando que con el símbolo “#” se denota una cantidad numérica, con “\$” se denota “peso” o “pesos”, y con “c” se denota “centavo” o “centavos”.

- Alfabeto: {#, \$, c}
- Estados: {q0, q1, q2, q3, q4}
- Estado inicial: {q0}
- Estados finales: {q2, q4}

Las reglas de transición están definidas por la tabla 3.2., con lo que el autómata está completamente definido. Hay que observar que los símbolos de entrada son un conjunto de palabras, pero que el reconocimiento total del autómata es una frase completa, y más en específico la sintaxis de dicha frase. De esta forma se pueden obtener autómatas más elaborados que puedan analizar la sintaxis de un mayor número de frases que puede generar otro contexto.

3.3. Semántica.

Finalmente, después de revisar que los símbolos que se están utilizando se encuentran en nuestro alfabeto (las palabras), y ver que la forma en que están acomodados es la correcta (sintaxis), se puede continuar con sacar la información de la frase para descifrar el mensaje que esta trae. Esto se realiza por medio de un análisis semántico, para el cual se pueden utilizar métodos descritos en los dos análisis anteriores, ya sea para realizar dicho análisis o como apoyo a otros métodos exclusivos para esta parte.

Una de las formas en que se realiza este análisis es por medio de los árboles que describen gramáticas libres de contexto, en los cuales se va pasando de palabra en palabra para obtener la información de cada una de ellas y así mezclarla para obtener el significado de la frase, como lo realizan en [2] y [5]. En la figura 3.6. se puede ver el análisis semántico de una frase, junto con la resolución de la ambigüedad que puede producir el lenguaje.

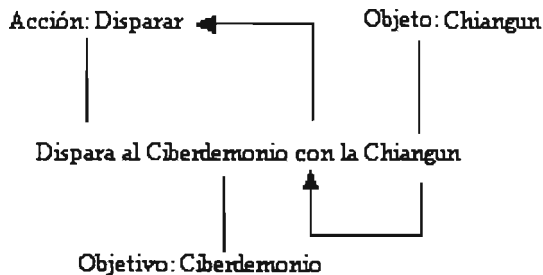


Figura 3.6. Análisis semántico de una frase [2].

Aquí podemos ver que la palabra "con" provoca una ambigüedad dentro de la frase, ya que se puede interpretar de dos formas, una diciendo que se dispare con el arma llamada Chiangun al Ciberdemonio, y la otra diciendo que se dispare al Ciberdemonio que tiene el arma. Dicha ambigüedad se resuelve al analizar las otras partes de la frase y tomando en cuenta el contexto (que en este caso es un videojuego) se deduce que el agente debe utilizar la Chiangun para disparar al Ciberdemonio [2].

Utilizando el algoritmo del árbol antes mencionado, el agente puede llegar a una conclusión errónea de la frase, por tanto es muy importante contar con el contexto que se está manejando, y en este caso en particular también ayuda la información que se tenga del ambiente virtual. La forma en que se analiza es pasando de palabra en palabra tomando la información que éstas puedan generar, al último se recopila dicha información, y se mezcla comparándola entre sí y con el contexto para obtener el significado de la frase.

Otra forma en que puede realizarse el análisis semántico es por medio de reglas de asociación utilizando la lógica de conjuntos. Esta forma abre la posibilidad de enlazar más de una frase en una sola sentencia, esto se logra utilizando las conjunciones que se manejan en dicha teoría. Para esto se debe

tomar en cuenta la tabla de verdad que se produce con estos operadores, ésta se presenta en la tabla 3.3.

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$
F	F	V	F	F	V
F	V	V	F	V	V
V	F	F	F	V	F
V	V	F	V	V	V

Tabla 3.3. Tabla de verdad de sentencias lógicas.

Aquí se consideran dos proposiciones P y Q, estas pueden considerarse como falsas o verdaderas (denotado con F y V respectivamente), y al ser operadas por alguna de las conjunciones mostradas, presentan un resultado también falso o verdadero. Las conjunciones presentadas son la negación, la conjunción, la disyunción y la implicación en dicho orden, aunque no son todas las que existen pero sí son las conjunciones más útiles para operaciones entre frases. En el siguiente ejemplo se muestra el uso de dicha técnica donde se ve las implicaciones que producen las dos frases.

Si es restaurante \Rightarrow sirve comida

Si es restaurante \wedge es vegetariano \Rightarrow sirve comida \wedge comida vegetariana

Estas se conocen como reglas de asociación, las cuales utilizan una base de conocimiento donde se dice si la proposición es verdadera o falsa. Esta base es la referida como base de conocimiento del agente, y aunque en los otros métodos no indiquen si las proposiciones son verdaderas o falsas, si deben indicar a que se refiere cada una de las partes que componen las frase.

Este método no es muy útil para conocer el significado de una frase, sin embargo, puede utilizarse para complementar otros y aumentar las capacidades de análisis a más de una frase por sentencia, como se mencionó anteriormente.

CAPÍTULO 4

HERRAMIENTAS DE SÍNTESIS DE VOZ

Existen muchos métodos para realizar el reconocimiento de voz, al igual que para la síntesis de texto a voz, pero estos son temas tan extensos que se han realizado trabajos de tesis para poder resolverlos. Por tal motivo, y para poder concentrar los esfuerzos de este trabajo sobre el diseño de interfaces basadas en diálogo, se eligió utilizar API's (Interfaz de Programa de Aplicación, o Application Program Interface) que facilitaran el uso de estas características.

4.1 Reconocimiento de voz.

Para el reconocimiento de voz se buscaron algunas herramientas que facilitaran su uso, de las cuales se encontraron de dos tipos, unas que brindaban dichas capacidades a programas ya existentes, lo que no ayuda al objetivo del presente trabajo, ya que no se tiene el control en el reconocimiento, además depende de las aplicaciones con que sean compatibles.

El otro tipo es el más útil para este tipo de sistemas, este consta de un API que ayuda a agregar la característica del reconocimiento de voz en los programas directamente. De estas API's se encontraron el "Via Voice" de IBM, el cual tiene capacidades de reconocimiento para varios idiomas, entre los cuales se tiene el español, que es de particular interés para el desarrollo del presente trabajo, pero tiene el inconveniente de ser muy caro. Otro que se encontró fue el API de Microsoft llamado "Speech", la cual es libre y se puede obtener de su sitio web, anteriormente tenía el inconveniente de que estaba especializado en el idioma inglés, puede ser entrenado para poder reconocer el idioma español de una forma un poco forzada, aunque ya se cuenta con su versión en español del "Speech Server". El problema de utilizar esta API es que la aplicación se encierra a trabajar únicamente en el sistema operativo Windows.

Otro API para el reconocimiento de voz es el SONIC, de la Universidad de Colorado [13], además de ser libre para el ambiente académico, puede reconocer varios idiomas, entre ellos el español, y aunque está desarrollado para trabajar en el sistema operativo Linux, también puede ser utilizado en el sistema operativo Windows, aunque con algunos requerimientos para su correcta compilación. En su distribución dan las bibliotecas ya compiladas junto con documentación y algunos ejemplos, lo de las bibliotecas llega a ser una desventaja debido a que dificulta la integración de éstas en los sistemas, y como está enfocada a Linux no tiene soporte completo para Windows.

Se revisa ésta última herramienta para conocer el entorno del reconocimiento de voz. Esta herramienta se compone de cinco módulos para realizar el reconocimiento de voz: **la representación de características, el modelo acústico, el modelo del lenguaje, el conjunto de símbolos fonéticos, y el mecanismo de búsqueda.** Está basada en la tecnología de los modelos ocultos de Markov con densidad continua. Sus modelos acústicos son árboles de decisión de estados conectados con un modelo de Markov oculto (HMM: Hidden Markov Model) con una gama que asocia funciones de densidad de probabilidad a un modelo de estados de duraciones. El reconocedor implementa una estrategia de búsqueda en dos pasadas. La primer pasada

consiste en una búsqueda sobre un árbol léxico. Durante la segunda pasada la cadena de palabras resultante es convertida en una palabra gráfica.

La **representación de características** utiliza la representación de coeficientes Cepstral de frecuencias Mel (MFCC: Mel Frequency Cepstral Coeficientes) de la señal del habla, utilizando un muestreo de 10ms y una ventana de audio de 20ms. El extractor de características procesa 12 coeficientes MFCC más una muestra normalizada y añade la primera y segunda derivada de dichas características para construir un vector característico de 39 dimensiones (100 vectores característicos por cada segundo de audio),

Su **modelo acústico** tiene una topología ubicada en tres estados, cada estado puede estar modelado con un número variable de distribuciones Gausianas mezcladas multivariadas. El entrenador del modelo acústico utiliza el algoritmo de Viterbi como modelo de estimación, esto reduce substancialmente el esfuerzo del CPU necesitado para entrenar modelos acústicos comparado con los modelos de entrenamiento adelante-atrás (forward-backward).

El soporte para el **modelo del lenguaje** está provisto por modelos de lenguaje basados en palabras y en clases estándar, modelos actuales de una, dos, tres o cuatro gramáticas pueden ser aplicados durante la primer pasada del reconocedor. SONIC puede procesar modelos de lenguaje que puedan ser procesados por la herramienta de modelado de lenguaje estático de Cambridge [15] y la herramienta de modelado de lenguaje de SRI¹ [13]. El soporte de gramáticas de estado finito basadas en el reconocimiento de voz también puede ser provisto.

El **conjunto de símbolos fonéticos** que puede utilizar la herramienta es arbitrario, donde el silencio siempre tendrá ligado el símbolo SIL. Actualmente para el idioma inglés utilizan un conjunto de símbolos de 55 fonemas, adoptado por las últimas versiones del reconocedor de voz Sphinx-II², dicho conjunto se muestra en la tabla 4.1.

Fonema	Ejemplo	Fonema	Ejemplo	Fonema	Ejemplo	Fonema	Ejemplo
AA	Father	DX	Butter	KD	Talk	GD	Mug
AE	Mad	DH	Them	JH	Jerry	SH	Show
AH	But	EH	Bed	K	Kitten	T	Tot
AO	For	ER	Bird	L	Listen	TH	Thread
AW	Frown	EY	State	M	Manager	UH	Hood
AX	Alone	F	Frend	N	Nancy	UW	Moon
AXR	Butter	G	Grown	NG	Fishing	V	Very
AY	Hire	HH	Had	OW	Cone	W	Weather
B	Bob	IH	Bitter	OY	Boy	Y	Yellow
CH	Church	IX	Roses	P	Pop	Z	Bees
D	Don't	IY	Beat	R	Red	ZH	Measure
PD	Top	BD	Tab	S	Sonic	SIL	Silence

¹ <http://www.speech.sri.com/projects/srilm/> página oficial de la herramienta.

² De la universidad Carnegie Mellon (CMU) <http://www.speech.cs.cmu.edu>

Td	Lot	DD	Had	TS	Bits	Br	Breathe
Ls	Lipsmack	Lg	laughter	ga	garbage		

Tabla 4.1. Fonemas definidos para el inglés [13].

El archivo de la configuración de los fonemas contiene la definición de los fonemas utilizados por el reconocedor, su archivo de configuración de estos inicia con una cabecera en ASCII, posteriormente se listan los símbolos de los fonemas válidos utilizados por el reconocedor. Este archivo tiene el formato mostrado a continuación.

```

<numphones> 55
<phonelist>
AA
AE
AH
...
</phonelist>

```

Su **mecanismo de búsqueda** está basado en el modelo de paso de token para el reconocimiento de voz, el reconocedor soporta el reconocimiento por palabras clave de una gramática, y en un diálogo continuo de vocabulario amplio. En su implementación, los tokens están propagados a través de un árbol léxico estático, los tokens mismos contienen información histórica de las palabras, así que los modelos de amplio espacio de tiempo y de n gramáticas pueden ser utilizados.

En conjunto estos módulos realizan el reconocimiento de las frases dichas por el usuario. En la figura 4.1 se muestra el esquema general de la forma en que la herramienta utiliza las distintas partes ya explicadas para realizar su reconocimiento, de aquí se puede notar que se utilizan una serie de archivos en donde se almacena la configuración de cada una de las partes. Dichos archivos deben tener un formato en especial, de estos los más relevantes para la aplicación son los que se describen a continuación.

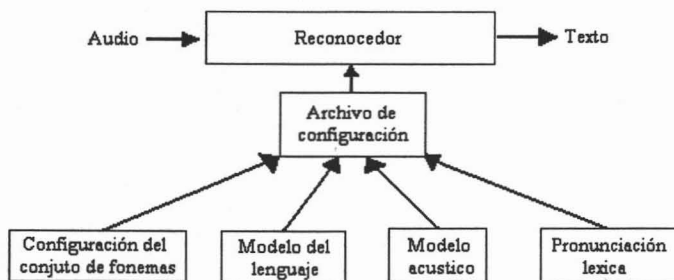


Figura 4.1. Esquema general del reconocedor de voz.

Los archivos de audio deben estar en formato PCM (Pulse Code Modulation) lineal de 16 bits sin cabecera, donde sus bytes deben estar acomodados de acuerdo al formato nativo de la máquina en que se compila la biblioteca.

El léxico contiene la pronunciación de las palabras contenidas en el vocabulario reconocido, las palabras son enlistadas seguidas por su secuencia de fonemas. Las pronunciaciones alternas son denotadas utilizando un número entero de alternativa entre paréntesis. Opcionalmente pueden ser asignadas probabilidades normalizadas a la pronunciación de las variantes utilizando los paréntesis cuadrados.

```
[0.00000] ACCIDENTAL    AE K S AX D EH N AX L
[-0.22185] ACCIDENTAL  AE K S AX D EH N T AX L
```

Para poder utilizar el API dentro de una aplicación, primero se debe llamar a la función de inicialización, dando como argumento el nombre del archivo de configuración del API.

Esta puede ser ejecutada de dos formas, en modo estático (modo batch) donde el reconocimiento se realiza sobre un audio ingresado al sistema a la hora de ejecutarlo, o en modo en vivo, donde una cadena de audio es conectado a la aplicación de manera que sobre este se realiza el reconocimiento, el cual puede cambiar en tiempo de ejecución. En general se tiene mayor efectividad al utilizar el modo estático, pero se pierde la característica del reconocimiento en tiempo de ejecución.

Para realizar el reconocimiento en el modo en vivo, se necesita definir el clásico ciclo definido en este tipo de aplicaciones. Este modo consiste en iniciar el *utt* (*utterance*: representación de pedazos de texto dentro de la frase **[14]**), poner la cadena secuencial de los datos en formato raw, determinar la hipótesis y finalizar el *utt*, esto se muestra en el siguiente pedazo de código.

```
char hip[10000];

decoder_begin_utt();

while( [muestras_enviadas_al_sistema] )
{ decoder_rawdata(chunk_of_rawdata, small_chunk_of_samples);
  decoder_get_partial_hypothesis(config, hip);
  printf("La hipótesis actual es [%s]\n", hip);
}

word_hypothesis = decoder_lattice_bestpath(); /* Imprime la hipótesis final*/
decoder_free_word_hypothesis(word_hypothesis); decoder_end_utt();
```

Aquí se puede notar que las hipótesis obtenidas dentro del ciclo no son la respuesta final del API, simplemente marcan un camino y estas son las posibles alternativas que va analizando. Para obtener la cadena reconocida es necesario llamar a la función "*decoder_lattice_bestpath()*", la cual contiene el reconocimiento final de la frase de entrada.

Otra de las características que puede manejar la herramienta es la entonación, para lo cual se cuenta con diferentes módulos de entonaciones disponibles con varios niveles de control. En general la entonación es generada en dos pasos.

1. Predicción de acentos (y/o tonos finales) en una por sílaba base.
2. Predicción de los valores de los objetivos, esto puede ser realizado después de tener la duración.

Reflejando esta una división hay dos principales módulos de entonación, esos llaman a sub-módulos dependiendo de los métodos de entonación requeridos. La entonación y sus módulos son definidos en Lisp, y llaman a los sub-modulos los cuales están en C++.

Se cuenta con una entonación por defecto, la cual es una forma simple de entonación y ofrece los módulos *Intonación_Default* y *Intonation_Target_Default*. El primero de ellos actualmente no hace nada. El segundo simplemente crea un objetivo en el inicio de la expresión y otro al final. Sus valores por defecto son 130Hz. Y 110Hz, sin embargo, estos valores pueden ser puestos a través del parámetro *duffint_params*, por ejemplo el siguiente generara un monótono a 150 Hz.

```
(set! duffint_params '((start 150) (end 150)))  
(Parameter.set 'Int_Method 'DuffInt)  
(Parameter.set 'Int_Target_Method Int_Targets_Default)
```

La entonación simple utiliza el árbol CART para predecir si cada sílaba es acentuada o no. Un valor predeterminado de nulo en los recursos de no acentuación es generalmente por la correspondencia de la función *INIT_Target_Simple*.

Hay que notar que este modelo no es soportado y puede ser complejo o inconveniente, pero con esta propuesta es muy rápido y sencillo generar algunas otras. Alguno similar a éste puede ser para español, de cualquier modo esto no está desarrollado como un módulo serio de entonación.

El árbol de entonación es un modelo muy flexible, dos árboles de clasificación y regresión distintos (conocidos como árboles CART) pueden ser utilizados para predecir acentos y entonaciones. Sin embargo actualmente este modulo es utilizado para una implementación del sistema de etiquetado de entonación ToBi³, el cual puede ser utilizado por distintos sistemas de entonación.

El objetivo de este módulo es utilizar el modelo de regresión lineal para predecir el inicio de media vocal y los objetivos finales para cada sílaba utilizando arbitrariamente características específicas. Este sigue el trabajo descrito en [15].

³ www.ling.ohio-state.edu/~tobi/ página oficial del proyecto.

El módulo de entonación general ayuda las especificaciones externas de cada una de las reglas por una extensa clase de teorías de entonación. Esta es designada para ser multi-lenguaje y ofrece en camino rápido reglas preexistentes frecuentes dentro de Festival sin escribir nuevo código en C++.

CAPÍTULO 5

ARQUITECTURA DE UNA INTERFAZ BASADA EN DIÁLOGO EN ESPAÑOL

5.1. Arquitectura.

Como ya se ha mencionado en capítulos anteriores, este trabajo enfoca sus esfuerzos en obtener el módulo de conversación verbal para agentes conversacionales. Más en específico se diseñó una metodología⁴ lo más general posible para el desarrollo de interfaces verbales en un contexto dado, la cual sigue una serie de pasos donde uno por uno se van obteniendo los distintos algoritmos con su respectivo enfoque al problema analizado. La generalización de esta metodología permite obtener interfaces verbales para distintos contextos, ya sea un agente que muestre qué cosas están en un cuarto en específico, o un tutor virtual, u otro tipo de aplicación que requiera un agente conversacional dentro de un contexto dado.

De las primeras consideraciones que se deben tener en cuenta es precisamente el contexto en el que se va a desenvolver el agente, de aquí se obtiene el sub-lenguaje que utilizará el agente y formará parte de la base de datos de conocimiento del mismo. Dicho sub-lenguaje deberá contener al menos la mayoría de las palabras utilizadas en el contexto dado, y necesariamente las palabras más comunes de este.

Para la interfaz se tomo una arquitectura que engloba los componentes que se consideraron necesarios para el tipo de aplicaciones a las que se quería llegar, aunque ya se ha mencionado que existen otras propuestas de arquitecturas [7], algunos de estos componentes son estrictamente necesarios para todo tipo de agente conversacional (como la base de conocimiento), aunque otros, como ya se ha mencionado en capítulos anteriores, no son del todo necesarios, estos estarán incluidos o no se tomarán en cuenta de acuerdo a la naturaleza del problema a resolver.

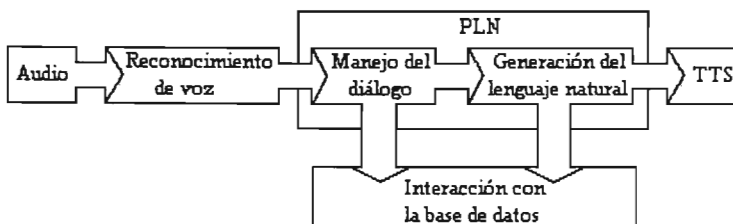


Figura 5.1. Flujo de datos de la arquitectura para la interfaz basada en diálogo.

En dicha arquitectura se muestran siete componentes que siguen el flujo de los datos desde la entrada verbal del usuario, hasta la salida también verbal del sistema. Hay que tomar en cuenta que dicha salida puede estar acompañada de la animación de un avatar que represente al agente. Los componentes de la arquitectura están definidos como sigue:

1. Audio.
2. Reconocimiento de voz.
3. Procesamiento del Lenguaje Natural.

⁴ Esta metodología fue publicada en [16] en forma resumida como parte de los planes para el presente trabajo.

4. Manejo del Dialogo.
5. Generación de Lenguaje Natural.
6. Interacción con la Base de Datos.
7. Síntesis de Texto-a-Voz.

Donde el componente del audio se encargará de recibir la señal analógica de la voz del usuario por medio de un micrófono, esta debe ser digitalizada para poderla procesar en la computadora donde estará en sistema con el agente conversacional. Después de ser digitalizado, debe ser enviado al componente de reconocimiento de voz, este flujo se muestra en la figura 5.1.

En el reconocimiento de voz toma el sonido de la voz previamente digitalizado por el componente de audio, este se procesa dichas señales digitales para obtener el conjunto de palabras de la frase que dijo el usuario, es decir, se obtiene la frase que dijo el usuario en una cadena de texto. Esta última debe ser enviada al procesador del lenguaje natural, como se muestra en la figura 5.1.

En el procesador del lenguaje natural toma la cadena de texto que contiene la frase que formulo el usuario, a esta la pasa por sus analizadores, desde el léxico pasando por el sintáctico y terminando con el semántico. Con ellos se procesa el diálogo del usuario para ver si es congruente, y si pasa los filtros mencionados, el PLN genere una respuesta adecuada a la frase analizada. Esta parte está integrada por dos componentes: el manejador del diálogo y la generación del lenguaje natural, como se puede observar en la figura 5.1., donde también se ve que este componente en cada una de sus partes requieren acceso a la base de conocimiento para poder realizar los análisis satisfactoriamente.

El manejo del diálogo se encarga de realizar los análisis mencionados apoyándose en la base de datos de conocimiento del agente. Este desglosa la frase en sus distintas partes para facilitar los métodos utilizados en los análisis realizados, dentro de estos métodos se va recolectando información que posteriormente se utilizará para poder generar una respuesta adecuada.

Los analizadores que lo componen son el léxico, el cual se encarga de ver si las palabras utilizadas por el usuario están contenidas en el sub-lenguaje que maneja el agente, además extrae el tipo de cada una, siendo esto muy importante para el resto de los análisis realizados. El otro analizador que contiene es el sintáctico, el cual revisa que el orden de las palabras dentro de la frase, es decir, su estructura sintáctica, sea la correcta. Al realizar estos análisis se va obteniendo información acerca del significado de la frase, dicha información es pasada al componente de generación de lenguaje natural.

En este último componente, se toma la información mencionada para realizar un análisis semántico de la frase y poder generar una respuesta adecuada que corresponda con el diálogo del usuario. Aunque este ya contiene información de la frase gracias a los análisis anteriores, es necesario acceder a la base de datos de conocimiento para complementar dicho análisis, además

de que es ahí donde se encuentra toda la información que el agente conoce y por consecuencia todas las respuestas que puede generar.

La salida de este módulo no simplemente es una cadena de texto que contiene el diálogo de respuesta por parte del avatar, sino que también puede contener una serie de instrucciones dirigidas al avatar para realizar alguna animación, ya sea de la boca, gestos, o que genere algún movimiento corporal. Estas salidas deben enviarse a sus respectivos módulos, donde la cadena de texto que contiene la respuesta del agente es enviada al componente de síntesis de texto a voz.

En la interacción con la base de datos se tiene el manejador de dicha base el cual se encarga de procesar las distintas peticiones realizadas por los componentes de manejo del diálogo y la generación del lenguaje natural, es decir los componentes que contiene el PLN. Este manejador procesa dichas peticiones y regresa los datos solicitados de forma que el componente que pidió esa información lo pueda entender. De los distintos datos que puede contener esta base se encuentran las palabras que conforman el sub-lenguaje definido para el agente, además también se pueden encontrar datos del ambiente o el contexto que está manejando el sistema, ya que es de aquí de donde se toman para poder armar la respuesta del agente. Además de estos datos, también se puede llegar a tener la forma de interpretar cada una de las palabras que conforman las frases, para ayudar al analizador semántico a sacar la información de estas, pero es este analizador quien con esta información debe ver la conexión entre las palabras para resolver ambigüedades donde una palabra puede dar un significado distinto según su posición dentro de la frase.

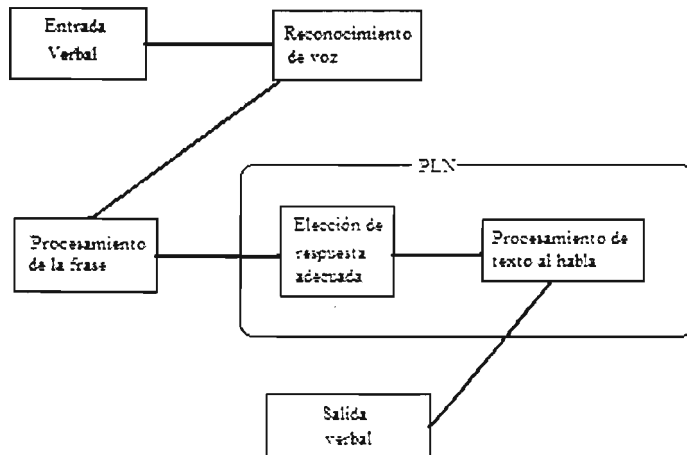


Figura. 5.2. Arquitectura del Sistema de Diálogo

El último componente es la síntesis de texto a voz (proceso de Text To Speech), esta toma la cadena de texto que le da el generador de lenguaje natural como se muestra en la figura 5.1. De esta manera Se realiza el proceso

en la cadena de texto para obtener un archivo de audio que contiene la frase dicha auditivamente. Así como con el reconocimiento de voz, para el TTS se emplea un API que ayudara a realizarlo y evitar el hacerlo desde el inicio, permitiendo enfocar los esfuerzos del presente trabajo al núcleo del problema que es el PLN.

El esquema general considerado para el procesamiento del diálogo se muestra en la figura 5.2, aquí podemos ver todos los bloques involucrados en dicha interfaz. Como ya se mencionó, para el reconocimiento de voz, así como el de procesamiento de TTS, se utilizan herramientas previamente desarrolladas.

5.2. Definición de la metodología para el PLN.

Entonces lo primero que se tomó en cuenta fue el idioma que requería que reconociera el agente, ya que este proporciona el tipo de estructura de las frases utilizadas dentro del diálogo. En este caso el idioma considerado fue el español, ya que es el oficial en este país, además no se encontraron aplicaciones de este tipo en dicho idioma lo que abrió la posibilidad de definir la metodología que aquí se presenta. Tomando como base dicho idioma, y definiendo el contexto donde se desenvuelve el agente, se realizó un análisis de las estructuras sintácticas que se podían generar dentro de dicho contexto dado por la naturaleza del problema, siendo este uno de los primeros pasos a seguir dentro de la metodología definida en el presente trabajo.

Con dicho análisis se obtiene el sub-lenguaje que debe manejar el agente, el cual como ya se ha mencionado, debe contener la mayor parte de las palabras utilizadas dentro del contexto y forzosamente las más comunes, de esta forma se comienza a obtener la base de datos de conocimiento del agente.

El mismo análisis ayuda a obtener el conjunto de estructuras sintácticas que utiliza el agente para el análisis de las frases, estas estructuras pueden ser plasmadas en forma de un árbol donde cada una de las ramas de este son las estructuras sintácticas obtenidas del análisis antes mencionado, estas pueden ser transformadas en un autómata para poder programar este analizador.

Pero estas estructuras están ligadas directamente con los tipos de las palabras que contienen, ya que cada nodo de dicho árbol es un tipo y no una palabra en sí. Estos tipos se obtienen en el análisis del lenguaje, y se pueden identificar en las frases obtenidas del contexto como puede verse en el siguiente ejemplo.

- Donde está la computadora.
- Como funciona el teléfono.

En estos dos casos se tienen un número distinto de palabras, pero unas comparten su tipo, las primeras se ve que son preguntas ya que estas están definidas en el lenguaje como tales, por lo que se puede decir que son del tipo pregunta. Las segundas ("está" y "funciona") pueden ser tomadas como

conceptos dentro del contexto del sistema, por lo que así es como se le denomina a su tipo. La tercer palabra de las frases son considerados como artículos en el idioma, por lo que se les da dicho tipo para el agente. Por último, las palabras restantes (“computadora” y “teléfono”) se refieren a objetos existentes en el ambiente que se está manejando ya sea virtual o real, por tal motivo pueden ser considerados del tipo objeto, y son estos últimos a los que casi siempre se hace referencia en este tipo de sistemas. Como se puede ver, son de estos últimos de los que el usuario puede solicitar información o realizar una petición para poder interactuar con ellos, por tanto la base de conocimiento del agente debe contener datos acerca de ellos.

Como se ha visto, los tipos de palabras se toman directamente de la forma en que los define el lenguaje, aunque no todos pueden ser considerados como tales, ya que se hace muy general esta clasificación lo que provoca que puedan existir ambigüedades tan grandes que los sistemas computacionales no puedan resolver correctamente. Por tal motivo se debe realizar un reclasificación de dichos tipos partiendo conjuntos generales que puedan provocar ambigüedades en conjuntos más pequeños que ayuden a disolverlas. Entre los tipos de palabras que fueron considerados se tienen los siguientes:

- *Acciones*: Las acciones se refieren a lo que puede hacer el agente como ir hacia un lugar en específico (denotado con la palabra “ve”) o mencionar algún concepto sobre un objeto (denotado con la palabra “dime”), en general son acciones directas que va a realizar el agente.
- *Preguntas*: Como preguntas se tienen todas las posibles preguntas que puede hacer el usuario desde el cómo hasta el porqué. Dentro de las preguntas se esconden una o más acciones que debe realizar el agente y que pueden quedar implícitas dentro de la frase, por ejemplo con la pregunta “donde” se le pide al agente que realice dos acciones, una es decir las características del entorno donde está ubicado el objeto en cuestión (como “sobre esa mesa”), la otra es señalar hacia la ubicación del objeto.
- *Conceptos*: Los conceptos se manejan como todas aquellas formas que tiene el ambiente de interactuar con el agente, por ejemplo el prender una maquina, tomar un objeto, cambiarlo de lugar, etc., por lo que son acciones del usuario pero relacionadas directamente con el ambiente.
- *Objetos*: Las cosas que se toman en cuenta como objetos son aquellas que existen en el ambiente virtual que tienen una interacción avanzada con el agente, no solo la colisión como el caso de las paredes o la detección de proximidad para abrir las puertas, sino que debe de realizar una acción para iniciar la interacción como oprimir un botón para encender una maquina o mover el brazo para activar una palanca. Pero como objetos no solo pueden ser tomados en cuenta las cosas que existen en el laboratorio virtual, sino que en ciertos contextos puede ser el tema abordado en la conversación, ya sea que se explique algún concepto utilizado dentro del laboratorio virtual o se aborde un tema de interés que sí conozca el agente.
- *Enlazadores*. El tipo de palabra denominado como enlazadores nos ayuda a darle el sentido correcto a la frase cuando esta carece de

algunos argumentos necesarios y que pueden ser introducidos por medio de estas palabras tomando en cuenta el contexto de la conversación. De este tipo de palabras se tiene por ejemplo “eso” utilizada cuando alguien quiere referirse al objeto o evento que en ese momento se convierte en el foco de atención, como cuando suena una alarma y alguien pregunta “¿qué es eso?”.

Hay que notar que estos tipos no son todos los que pueden ser considerados dentro de un contexto, ni tampoco es necesario que cada aplicación los maneje, simplemente es una lista que se puede utilizar como referencia ya sea para clasificar a una palabra dentro de alguno de estos tipos o para obtener un nuevo tipo de palabra.

Posteriormente después de tener el autómata para la revisión sintáctica de las frases, se debe tener la forma de sacar la información de estas, en este caso se pueden utilizar alguno de los distintos algoritmos revisados en el capítulo 3 en la parte de la revisión semántica. De aquí se debe obtener la respuesta que debe dar el agente al usuario de acuerdo con la frase de este último.

5.3. Procesador del lenguaje natural.

Con dicha metodología definida como se presentó, se tiene un panorama general para el diseño de este tipo de interfaces, para su desarrollo se divide en dos partes, una que se encargue del pre-procesamiento donde se obtiene el tipo de cada una de las palabras y se organiza la frase para que los siguientes módulos puedan procesarla con mayor facilidad. La segunda parte es precisamente el procesamiento del lenguaje natural sobre la frase desglosada anteriormente junto con la información obtenida en ese paso. El esquema que se siguió para el desarrollo del procesador de lenguaje natural es el que se muestra en la figura 5.3 lo cual corresponde a los bloques de procesamiento de la frase y la elección de la respuesta adecuada del diagrama general.

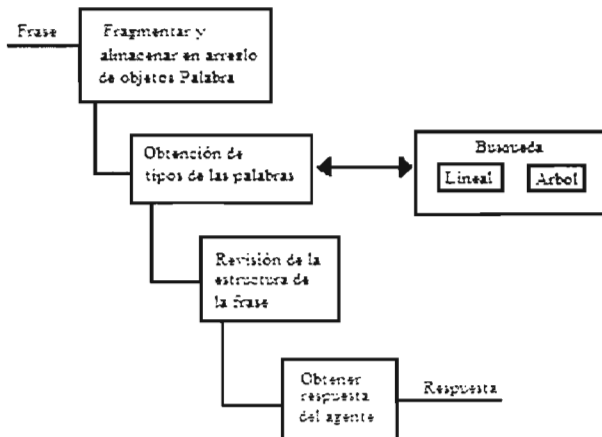


Figura 5.3. Procesador de Lenguaje Natural.

En el diagrama se pueden ver cada una de las partes en que fue dividido el programa, así como la línea de trabajo que sigue ya que va bloque por bloque desde la entrada de la frase hasta la salida de la respuesta en forma secuencial.

5.3.1 Pre-procesamiento del Diálogo.

En el procesamiento del dialogo se manejan los dos primeros bloques de la figura 5.3. Después de adquirir la frase del reconocedor en la forma requerida, ésta debe segmentarse en cada una de sus palabras y almacenarse en una estructura para manejarla posteriormente en su análisis. La estructura utilizada para dicho fin es un objeto llamado "Palabra", donde los datos de mayor interés que almacena son la palabra y su tipo ya que de estos dependen que la frase pase la revisión sintáctica y la respuesta que genere el sistema.

Ya dividida la frase en las distintas palabras que la conforman y almacenadas en un arreglo de objetos de tipo "Palabra", se realiza una búsqueda para obtener el tipo de cada una. Dicha búsqueda se realiza sobre un conjunto de archivos cuyo nombre esta relacionado con su contenido, por ejemplo el archivo "objetos.dat" contendrá los objetos que reconocerá el agente, o el archivo "accione.dat" contiene todas las acciones a realizar por parte del agente.

Estos archivos contienen las diferentes maneras en las cuales las palabras pueden utilizarse dentro de una frase, de esta forma el agente puede comprender las distintas maneras de armar una frase con el mismo significado, por ejemplo, si se le dice al agente:

- "Dime como funciona",

El archivo "accione.dat" debe contener la palabra "Dime", pero se puede decir lo mismo con las frases:

- "Di como funciona"
- "Me dices como funciona"
- "Puedes decir como funciona",

Por lo tanto además de contener la palabra de la acción normal "Decir" también debe contener sus distintas formas en que puede ser utilizada dentro del contexto:

- "Dime"
- "Dices"
- "Di".

La estructura que tienen los archivos es tener primero la palabra seguida de la instrucción o dato. De esta manera se pueden cargar en una estructura ya sea de lista o de árbol para realizar la búsqueda en base a estas. Dichas estructuras son generadas al iniciar el sistema para aumentar su eficiencia y no aumentar el tiempo de respuesta.

5.3.2 Procesamiento del Diálogo.

Después de organizar la frase en el arreglo de objetos "Palabra" y haber obtenido su tipo se puede comenzar a procesarla. La primera parte de su procesamiento es revisar su estructura sintáctica para lo cual nos ayuda una estructura tipo árbol generada en el análisis del problema. Esta contiene las formas sintácticas de las frases correspondientes a un contexto dado. Para un pequeño ejemplo de un agente mostrando objetos dentro de un laboratorio virtual la estructura generada es la siguiente:

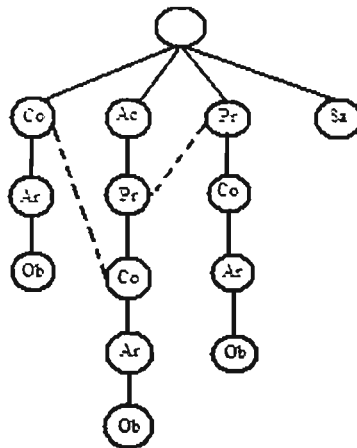


Figura 5.4. Árbol de estructuras de la frase

Los nodos de dicha estructura representan cada una de las palabras que pueden ser utilizadas, las marcas contenidas en los nodos son los diferentes tipos de palabras que se obtuvieron previamente en el análisis, estas son:

- Ac = Acción
- Ar = Artículo
- Co = Concepto
- Ob = Objeto
- Pr = Pregunta
- Sa = Saludo

Se puede observar que la mayoría de los tipos mostrados son los encontrados en el análisis general, a excepción de los artículos que en este caso son utilizados solo como una referencia del estado de la frase. Estos pueden tomarse en cuenta como palabras sin relevancia para el significado de la frase pero existentes en la estructura. Después de los conceptos e incluso al inicio de la frase puede ir un conjunto pequeño de palabras (a lo más tres) que estén en la misma situación de los artículos, por ejemplo en la frase: "puedes decirme como prendo la maquina", se tiene una estructura como sigue:

NI Ac Pr Co Ar Ob

NI Puedes
Ac Decirme
Pr como
Co prendo
Ar la
Ob maquina

Donde NI se toma como una palabra no identificada pero como está al inicio la frase estará correcta ya que son menos de tres y el resto de la estructura corresponde a una línea del árbol. En el ejemplo se ve claramente que el tipo "Artículo" puede ser tomado como una palabra sin identificar con su especificación, aunque en este caso no fue así.

Tomando en cuenta la estructura tipo árbol, se construyó un autómata que es el que revisa su sintaxis. Las consideraciones que se tomaron para pasar del árbol al autómata son las siguientes:

- Todos los nodos del árbol serán estados del autómata y la transición de estados estará dada por la jerarquía del árbol.
- Se toma la raíz como estado de inicio con la consideración ya mencionada de las palabras no identificadas.
- Se buscan ramas que sean un subconjunto de otras mayores para omitirlas al momento del traspaso de los nodos del árbol a los estados del autómata. En la figura 3 están señaladas por líneas punteadas.
- Se pone una transición del estado de inicio a todos aquellos estados que eran nodos conectados a la raíz y que carezcan de transición a dicho estado.
- Se pone una transición de cada estado que fue un nodo hoja hacia un estado final, el cual será el estado de aceptación.

De esta forma el autómata del ejemplo queda como se ve en la figura 5.5. Este será el que revise que la sintaxis de la frase será correcta sólo si se llega al estado final.

Contando con la revisión sintáctica de la frase se toman en cuenta los significados de las palabras. En este punto es donde el concepto de marca tiene mayor relevancia. Hay que notar que las de mayor importancia son las de "Acción" y "Pregunta" ya que estas contienen las instrucciones de lo que debe hacer el agente. En el caso de la acción, se tiene directamente lo que se quiere que realice, ya sea moverse a algún lugar o decir información de alguna cosa. El caso de las preguntas es muy similar, ya que la naturaleza de la pregunta proporciona una o más acciones a ejecutar. Por ejemplo, el preguntar la ubicación de un objeto, lleva dos acciones:

- a) Señalar o moverse hacia el objeto
- b) Dar información sobre su ubicación, como los objetos cercanos.

Las marcas restantes proporcionan información acerca de la acción que se le está pidiendo al agente, por lo que complementan las instrucciones que se generan en la revisión de cada marca, estas son procesadas por el sistema tanto para mandar al agente los movimientos que debe ejecutar como para realizar una búsqueda de la respuesta adecuada dentro de la base de datos de respuestas según sea el caso.

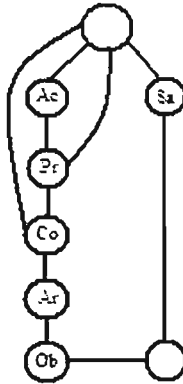


Figura 5.5. Autómata de revisión sintáctica

Entonces la línea que sigue el procesador del lenguaje natural se puede ver, por ejemplo, al decirle al agente la frase: "Cómo funciona la maquina", después de que el reconocedor de voz genera la cadena, el sistema la obtiene y analiza generando cuatro objetos palabra con sus respectivos tipos, los cuales son:

<i>Pregunta</i>	<i>Como</i>
<i>Concepto</i>	<i>funciona</i>
<i>Artículo</i>	<i>la</i>
<i>Objeto</i>	<i>maquina</i>

Al entrar la frase en el autómata este revisa su sintaxis pasando por correcta, ya que podemos ver un camino desde el estado de inicio hasta el estado final sin saltarse en ningún punto. Posteriormente se busca la acción dentro de la oración, al carecer de esta marca se busca la de "Pregunta" la cual genera dos instrucciones, la primera es "Decir" la segunda es "Mostrar". Podemos notar que dichas instrucciones sólo es una parte de todo lo que quiso decir el usuario con la frase y que con esta el agente no puede concretar la acción pedida. La información faltante está contenida en las otras palabras. Siguiendo con el análisis el sistema se encuentra con la palabra "funciona" de tipo "Concepto" lo cual le da el tema a las dos instrucciones, pero aún no se sabe de qué se tiene que hablar, pero aún faltan palabras a analizar. La siguiente es de tipo "Artículo" las cuales no proporcionan ningún tipo de información a la instrucción y por lo mismo están consideradas como candidatas para tomarlas como palabras no identificadas. Por último se tiene la palabra "maquina" de tipo "Objeto" lo que le da a las instrucciones generadas el

contexto sobre el cual se está refiriendo la frase, por lo tanto las dos instrucciones generadas por el sistema quedan como se muestra a continuación:

decir función maquina
mostrar función maquina

Teniendo la síntesis de la frase en estas instrucciones, el sistema ya puede entenderlas y procesarlas correctamente por medio de un parser interno, con lo que al reconocer la instrucción "decir" ejecutará una búsqueda en la base de datos de las respuestas con los datos que contiene para posteriormente mandar la respuesta encontrada al procesador de texto a voz y así generar la salida de audio. Y al encontrarse con la instrucción "mostrar" consultará la base de datos de los movimientos del agente de la misma forma con los datos que contiene para que el agente ejecute la serie de movimientos encontrados en dicha base, referentes a la búsqueda realizada, al mismo tiempo que se escucha el audio generado.

CAPÍTULO 6

DESARROLLO Y RESULTADOS DE LA INTERFAZ PARA EL DIÁLOGO

La metodología descrita en el capítulo anterior para el diseño de interfaces verbales es mostrada por medio de un conjunto de módulos, los cuales se comunican entre sí para poder lograr obtener dicha interfaz. Pero aunque se tienen algoritmos y hasta un autómata que describe una pequeña aplicación, se necesita probar en un caso más elaborado para probar que dicha metodología puede ser utilizada para definir la interfaz verbal no sólo de un caso en particular, sino de varios casos aunque estos no se parezcan entre sí.

6.1. Caso de uso.

Para probar la metodología descrita en este trabajo, se requirió diseñar una aplicación que necesitara el uso de un agente conversacional. El contexto elegido fue un supermercado en donde el agente proporcione información de los productos como su ubicación y su precio. Lógicamente se utilizaron los métodos modificados descritos en el capítulo anterior para que la interfaz fuera en español.

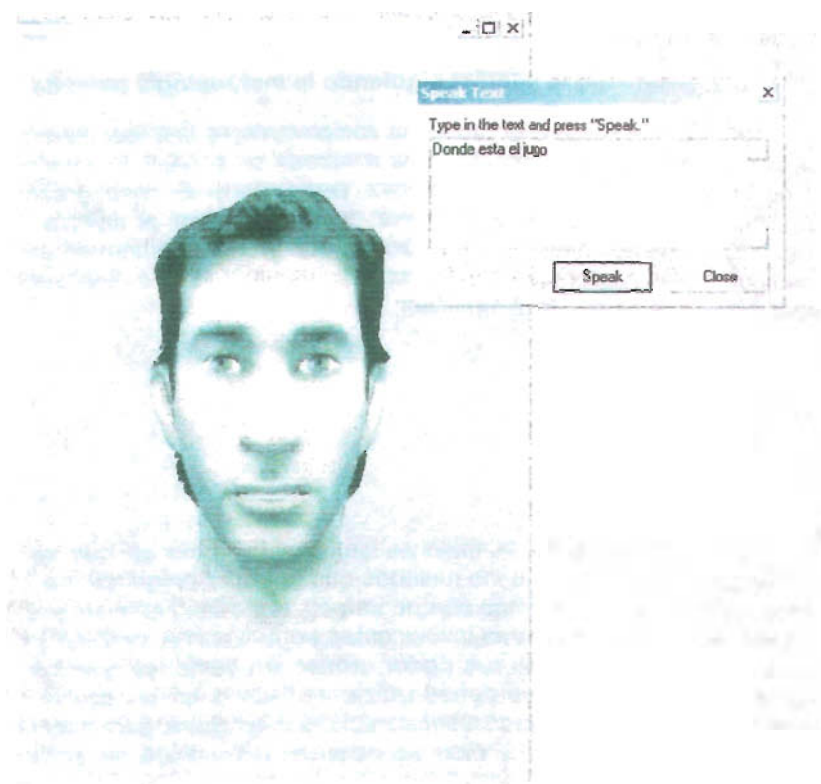
Como el problema, al que da respuesta el presente trabajo, surgió de la falta de una interfaz verbal de una aplicación anteriormente desarrollada, se ingresó el contexto mencionado a ésta. Dicha aplicación presenta un modelo de una cara en 3D, donde para iniciar a utilizar el módulo que engloba éste trabajo se elige su opción desde el menú, como respuesta mostrará una ventana de diálogo donde ya está iniciado y activado el reconocimiento de voz, toda palabra reconocida se mostrará en dicha ventana de diálogo, y al pulsar su botón "OK" se iniciará el procesamiento de la frase que se encuentre en la ventana. La respuesta del agente será de forma verbal, realizando el modelo los movimientos de la cara, y reproduciendo una frase con la respuesta adecuada a cada diálogo del usuario.

De esta forma, al estar definidas las entradas y salidas del sistema, es posible ver los dispositivos que serán necesarios para trabajar con el sistema y que formarán parte de la interfaz en general, la cual se puede considerar como una interfaz inteligente. Como entrada por parte del usuario se tiene su diálogo verbal, por lo mismo se necesita tener un micrófono que pueda captar estos diálogos y los transforme en señales digitales para la computadora, además de un teclado ya que la interfaz gráfica de usuario permite hacer modificaciones al diálogo reconocido.

De las salidas del sistema se tienen dos tipos, una sonora que será el diálogo que genere el agente como respuesta al diálogo del usuario, para lo cual es necesario contar con una bocina por donde se reproduzca el sonido de la voz del agente, y una gráfica que es donde se presenta el modelo de la cara en 3D.

Para este tipo de salida se necesita un dispositivo de despliegue, pero se tienen varias alternativas que muestran el resultado de distintas formas. Una de las alternativas es utilizar un casco de realidad virtual, otra puede ser desplegar con un cañón sobre una superficie grande, pero por la naturaleza de la aplicación no es buena opción, ya que al mostrarse en un área tan grande los usuarios pueden inhibirse frente a otros usuarios quedando restringida a un

número menor del pensado, en cambio si se utiliza un monitor de un tamaño normal (aproximadamente 15 a 17 pulgadas), el uso del programa se hace más personal y por lo mismo permite que un mayor número de personas lo utilice.



6.1. Interfaz utilizada para probar el sistema, ésta presenta al avatar como una cosa que muestra sus facciones al dar una respuesta verbal. El cuadro de dialogo que aparece a la derecha es donde se muestra la frase reconocida, y lo que se utiliza de entrada para el FLN.

Ahora que se tienen definidos los dispositivos de la interfaz, se puede proseguir con la resolución del problema, el cual debe ser acotado ya que un centro de información puede manejar un gran número de datos, pero para demostrar la forma de utilizar la metodología sólo se necesita un pequeño subconjunto de estos, por lo tanto el agente únicamente dará información de las siguientes características de algunos de los productos que pueden existir dentro de una tienda departamental.

- Ubicación.
- Precio.
- Existencia.
- Marcas.
- Descripción.

Ahora falta definir los productos que deben estar en la base de conocimiento del agente, donde cada uno de ellos debe contener los datos expresados en la lista anterior. De esta forma se comienza a abastecer dicha base con la información que se necesitará posteriormente en el sistema para poder mostrarla de una forma coherente a cada una de las peticiones que realice el usuario al sistema. La parte gráfica del sistema se muestra en la figura 6.1.

6.2. Diseño de la interfaz siguiendo la metodología definida.

Ahora que se tiene el problema completamente definido, es momento de comenzar a utilizar la metodología mostrada en el capítulo anterior para poder hacer el diseño de la interfaz verbal para el caso presentado. Siguiendo esta metodología, el primer paso es realizar el análisis de las frases que pueden intervenir en el sistema, con lo que primero se obtienen los tipos de las palabras que van a estar contenidas en el sub-lenguaje, los tipos obtenidos para este ejemplo son:

- Ac: Acción.
- Pr: Pregunta.
- Co: Concepto.
- En: Enlace.
- Pc: Precio.
- Ob: Objeto.

Además de su tipo también se muestra la forma en que se hace referencia a estos dentro de los métodos que utiliza el programa. Se puede observar que, al mismo tiempo en que se está realizando este análisis para obtener el tipo de las palabras involucradas en el sistema, también se está obteniendo el sub-lenguaje que podrá utilizar el agente, ya que los tipos aparecen de las frases que puede utilizar el usuario en las cuales están contenidas las palabras que conforman dicho sub-lenguaje, pero además de obtener esta información, también se obtienen las estructuras sintácticas que intervienen en ese contexto. Estas últimas son acomodadas en un árbol de estructuras sintácticas, donde como se definió anteriormente, cada una de las ramas es una estructura válida para el sistema, dicho árbol se puede ver en la figura 6.2., además a continuación se muestra una lista de cada estructura utilizando las abreviaturas de los tipos de las palabras encontrados y acompañados de una frase que ejemplifica dicha estructura.

- Ac Pr Pc Co (Dime que precio tiene)
- Ac Pr Pc (Muéstrame cuanto cuesta)
- Ac Pr Co (Muestra como llegar)
- Ac Pr Co En Ob (Dime donde está el jugo)
- Ac En Pc En Ob (Dime cuanto cuesta el jugo)
- En Pr Co (En donde está)
- En Pr Co En Ob (Dime donde está el jugo)
- Co Ob (tienen jugo)
- Pr Pc (Cuanto cuesta)

- Pr Pc Co (Qué precio tiene)
- Pr Pc En Ob (Cuanto cuesta el jugo)
- Pr Co (Donde está)
- Pr Co En Ob (Donde está el jugo)

Con esta lista se puede observar la forma de sacar los tipos de las palabras, así como las estructuras sintácticas que serán utilizadas para dicho analizador. También se ve que en dichas frases están contenidas las palabras que se van a utilizar para armar los diálogos que puede reconocer el sistema, además si se siguen extendiendo las distintas posibilidades de las frases que

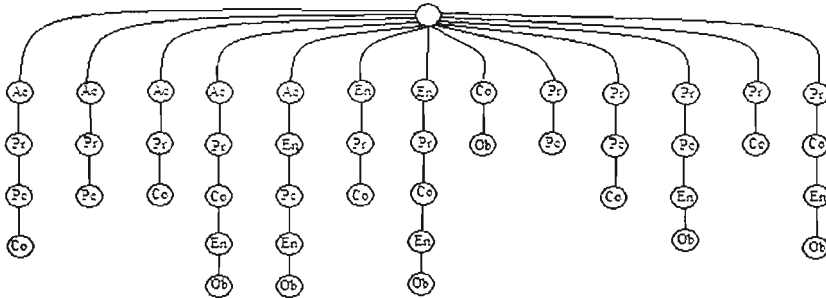


Figura 6.2. Árbol de estructuras jerárquicas.

pueden ser utilizadas por parte del usuario, se tendrá en su totalidad el sub-lenguaje que manejará el agente, además de todos los objetos a los que se hará referencia, y por consecuencia, de los que se necesita tener la información anteriormente descrita.

Ya con estas estructuras se puede tener el árbol de estructuras sintácticas, el cual es mostrado en la figura 6.2. En este se pueden observar todas las estructuras anteriormente mencionadas en la lista de forma que cada una forma una rama del árbol. De este árbol se puede obtener un autómata, simplemente se puede tomar como estado inicial su raíz, y agregar un nuevo nodo al que se conecten todas las hojas, este se utiliza de estado final.

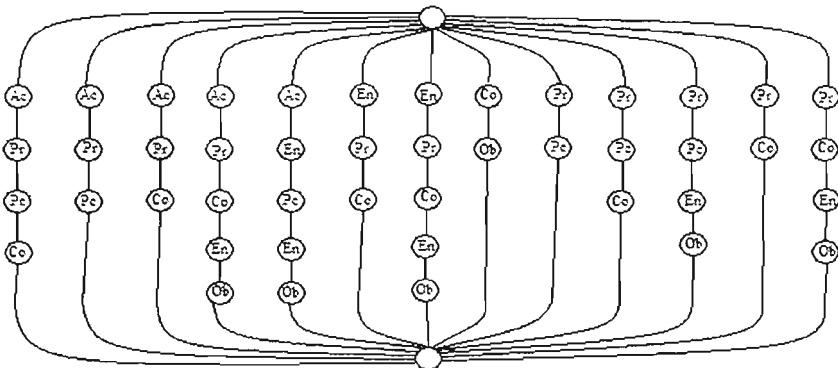


Figura 6.3. Autómata agregando un sólo estado final.

Al realizar una conversión de este tipo se pueden tener autómatas enormes como el mostrado en la figura 6.3, donde cada uno de los estados del autómata corresponde a un nodo dentro del árbol (excluyendo el estado final).

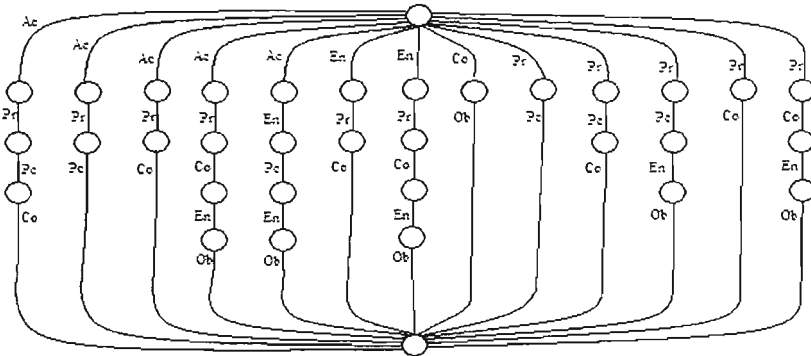


Figura 6.4. Autómata uniendo todas las hojas del árbol en un estado final

Los símbolos de entrada para cada una de las transiciones del autómata son el tipo de la palabra del nodo al que llegan, por lo que los nodos hoja pueden ser integrados en uno mismo que reemplace al estado final, esto es mostrado en la figura 6.4.

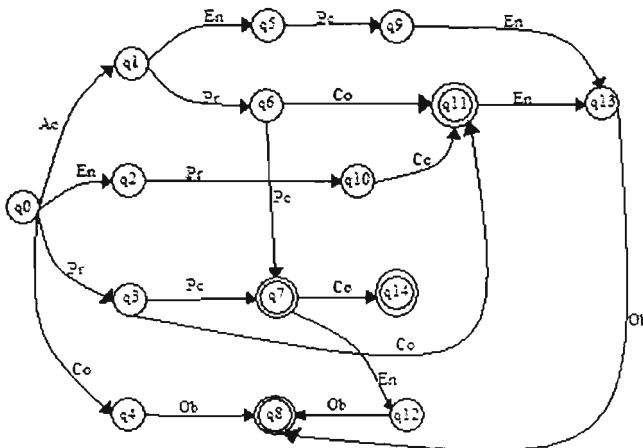


Figura 6.5. Autómata de las estructuras sintácticas del ejemplo.

De esta manera se eliminan un gran número de estados, pero sigue quedando un autómata que puede ser reducido con algún método definido en su teoría. Pero este no es el único problema, además de tener muchos estados, se tiene un autómata no determinístico, ya que en el estado inicial con un mismo símbolo de entrada, tiene la posibilidad de ir a más de un estado. La resolución de este tipo de autómatas es más elaborada que la resolución de los autómatas finitos determinísticos, por lo que se busca transformar el autómata de la figura 6.4 en uno de este estilo, la teoría de autómatas tiene métodos para esta transformación, pero en este tipo de casos se puede utilizar el descrito en la metodología ya que varias de las estructuras son similares entre

sí. De esta manera, varias de las estructuras que inician en estados diferentes, se pueden llegar a encontrar en estados posteriores por lo mismo de que terminan con los mismos tipos de palabras y en el mismo orden.

En la figura 6.5. se muestra un autómata equivalente al mostrado en la figura 6.4. pero con la diferencia de que este sí es determinístico, es decir, en cada estado por cada símbolo de entrada se tiene una sola transición. De esta manera se tiene el autómata definido por sus siguientes componentes.

- Alfabeto = {Ac, Pr, Co, En, Pc, Ob}
- Estados = {q0, q1, q2, ..., q13, q14}
- Estado inicial = {q0}
- Estados finales = {q7, q8, q11, q14}

Donde las transiciones están dadas por la tabla 6.1. que se muestra a continuación.

	Ac	Pr	Co	En	Pc	Ob
q0	q1	q3	q4	q2	0	0
q1	0	q6	0	q5	0	0
q2	0	q10	0	0	0	0
q3	0	0	q11	0	q7	0
q4	0	0	0	0	0	q8
q5	0	0	0	0	q9	0
q6	0	0	q11	0	q7	0
q7	0	0	q14	q12	0	0
q8	0	0	0	07	0	0
q9	0	0	0	q13	0	0
q10	0	0	q11	0	0	0
q11	0	0	0	q13	0	0
q12	0	0	0	0	0	q8
q13	0	0	0	0	0	q8
q14	0	0	0	0	0	0

Tabla 6.1. Tabla de transiciones del autómata

En la tabla de transiciones se puede observar que aún hay estados similares los cuales pueden ser considerados como el mismo, eliminando de esta manera un estado más. Tal es el caso de los estados q3 y q6, donde con un símbolo de entrada Co se realiza una transición al estado q11 y con un símbolo Pc se realiza una transición al estado q7. Esta reducción de estados se omitió en el ejemplo pero se muestra para hacer notar que hay más de una forma de tener un autómata que describa el mismo lenguaje (en este caso el sub-lenguaje definido para el agente).

La programación del autómata puede ser facilitada con su tabla de transiciones donde únicamente hay que ver la forma de ir de un estado a otro revisando si las transiciones son válidas, y cuando se llegue a un estado final ver si no quedan más símbolos de entrada por resolver. Esto puede realizarse con los siguientes pedazos de algoritmo.

Primero se debe definir el estado q_0 como el estado actual para iniciar el análisis, entonces se define un ciclo que continua mientras el estado actual sea válido.

- Repetir mientras act_i sea un estado válido

Después, los estados se dividen en dos categorías: estados normales y estados finales, donde la diferencia entre ellos es que si se acaban los símbolos de entrada cuando se llega a este estado la respuesta varía. En los estados normales la cadena es tomada como no válida por lo tanto falla su análisis, esto puede verse con el siguiente algoritmo que revisa el estado q_6 el cual es de este tipo (hay que tomar en cuenta que dicha revisión se encuentra dentro del ciclo),

- En caso de que $act = q_6$ entonces hacer los siguientes pasos
 - Si el símbolo de entrada es nulo entonces el estado actual es un estado no válido y la cadena no es válida.
 - En caso contrario
 - En caso de que el símbolo de entrada sea C_0 , $act = q_{11}$
 - En caso de que el símbolo de entrada sea P_c , $act = q_7$
 - En otro caso la cadena de entrada no es válida y $act =$ un estado no válido.
 - Pasar al próximo símbolo de entrada.

Además de la forma de tratar el problema de quedarse sin símbolo de entrada, el algoritmo presenta el análisis de los casos en que los símbolos de entrada son válidos, pero también ataca el problema de cuando no tiene transición válida para un símbolo de entrada determinado, resultando en la invalidez de la cadena analizada.

El algoritmo utilizado para revisar estados finales únicamente cambia en el segundo paso, validando la cadena analizada si no se cuenta con más símbolos de entrada. Este algoritmo se muestra ejemplificando el estado q_{11} .

- En caso de que $act = q_{11}$ entonces hacer los siguientes pasos
 - Si el símbolo de entrada es nulo entonces el estado actual es un estado no válido y la cadena si es válida.
 - En caso contrario
 - En caso de que el símbolo de entrada sea E_n , $act = q_{13}$
 - En otro caso la cadena de entrada no es válida y $act =$ un estado no válido.
 - Pasar al próximo símbolo de entrada.

Hay que notar que estos algoritmos van cambiando según el estado del que se trate, ya que cada uno tiene su propio conjunto de símbolos de entrada que son aceptados y por lo tanto generan una transición no válida, y en las ocasiones en que los estados aceptan los mismos símbolos de entrada, no siempre realizan una transición al mismo estado.

Al estar revisando la sintaxis con el autómata y los algoritmos anteriores, se puede estar obteniendo información extra cada vez que se capta un símbolo de entrada válido, en estas ocasiones se manda la palabra en turno para poder ir llenando las instrucciones que deberá ejecutar el agente en cualquiera de sus áreas de dibujado o por medio de la bocina mencionada anteriormente.

Pero sacar la información de esta forma para ir llenando una pila de instrucciones con sus respectivos datos, no es suficiente para poder obtener el significado de la frase. Dentro de la base de datos de conocimiento del agente, se debe tener el tipo de acción que puede generar una palabra dentro de la frase, en este caso las acciones generadas son:

- Decir.
- Mostrar.
- Mover.

Donde la primera se refiere a presentar una respuesta verbal por parte del agente, la cadena de texto que se tendrá que enviar al módulo de TTS es obtenida de la misma base de conocimiento, pero no se trata de obtener cualquiera, sino que se debe buscar exactamente la información requerida por el usuario. Para esto se formulo una estructura que contiene todos los datos que el agente puede presentar, con lo que sólo basta saber que es lo que solicita el usuario, ya que de esta forma se puede acceder dicha información directamente.

Además de esta estructura, se cuenta con un objeto que realiza el manejo de las instrucciones y los datos que puedan necesitar, con lo que es este objeto el que contiene la referencia del contexto interno que se está manejando, que en este caso es cualquier producto por el cual se esté preguntando.

La instrucción mover se refiere a que el agente mueva su geometría asociada, conocida como el avatar, o algún otro avatar a través del mapa que se muestra gráficamente en la interfaz. Esta instrucción debe mandar llamar a los algoritmos especializados en la búsqueda del mejor camino en un cuarto dado (en este caso la tienda departamental) esquivando los obstáculos presentados en él (estantes, cajas y demás elementos que contenga la tienda), pero dichos programas fueron desarrollados en la tesis titulada "Navegación dirigida de agentes virtuales en realidad aumentada".

Con la instrucción mostrar, el agente debe poner en el área de dibujo correspondiente una imagen del producto solicitado, para este fin se agrego a la estructura que contiene la información de los productos una dirección donde puede ser encontrada dicha imagen, de la misma forma que se realiza con las otras dos instrucciones, la referencia del producto que está solicitando el usuario la toma de la estructura mencionada.

Con esto se termina el diseño e implementación de la parte de la interfaz que se encarga del procesamiento del diálogo y la generación de la respuesta adecuada, que como se ha mencionado, conforman al PLN. Lo que resta por

hacer es conectarlo con los otros módulos considerados en la arquitectura del sistema, lo cual es inmediato tomando en cuenta que el reconocedor de voz le entrega la frase en una cadena de texto al PLN, y que este a su vez entrega el conjunto de instrucciones, entre las cuales se encuentra la cadena de texto que contiene el diálogo del agente la cual es enviada al módulo de TTS para obtener la frase auditivamente, las otras instrucciones también salen inmediatas para que el interprete del módulo de animación las procese y realice los cambios necesario en las geometrías del ambiente virtual (mapa de la tienda departamental) o del avatar (la cara del agente).

6.3. Pruebas del sistema.

Para las pruebas del sistema se utilizó el ejemplo de la tienda departamental descrito anteriormente en este mismo capítulo. Las pruebas se dividieron en dos tipos siguiendo cada una el objetivo buscado para este trabajo.

La primer prueba fue medir el tiempo de respuesta del sistema, el cual inicia desde que el sistema capta la finalización de la frase por parte del usuario, donde el sistema da un tiempo determinado (aproximadamente tres segundos) para decidir que el usuario terminó su diálogo. La medida del tiempo finaliza al momento en que el sistema da una respuesta al usuario, no importando que esta sea o no satisfactoria, o que la frase procesada no pase la prueba de alguno de los filtros que tiene el sistema.

Hay que notar que para obtener dichas medidas de tiempo no se tomó en cuenta la parte gráfica del sistema, donde los tiempos medidos se muestran en la tabla 6.2. En ella se consideraron tres partes que restan del programa, estas son el reconocimiento de voz, el PLN y el TTS, para cada uno de los casos que se probaron se tiene la medida de los tiempos con distintas combinaciones de las parte mencionadas.

Frase	Todo	Parte 1 y 2	Parte 2 y 3	Sólo PLN
Donde esta el jugo	1.1	0.7	0.8	0.4
Dime donde esta el jugo	1.3	0.9	0.9	0.5
Cuanto cuesta	1.1	0.7	0.7	0.4

Dichos tiempos están dados en segundos, y muestran una respuesta eficiente para el usuario, ya que un humano puede llegar a esperar al menos tres segundos para que alguien le de una respuesta, aunque hay que notar que en una conversación real, el intercambio de diálogos se realiza en menos de un segundo.

El otro tipo de prueba que se realizo fue el tratar de ver que tan robusto era el sistema para poder identificar las frases que entraban en su contexto. Esta se realizo dejando que un número de usuarios utilizara el sistema con una breve explicación previa del contexto que maneja y del uso de la interfaz, de esta manera los resultados obtenidos por dicha prueba arrojaron las siguientes observaciones.

- Dentro del contexto que maneja el sistema, se reconoció la mayor parte de las frases utilizadas (aproximadamente el 95%).
- Los usuarios formulaban frases dentro del contexto de un módulo de información de una tienda departamental pero fuera del sub-contexto definido para este ejemplo.
- Las frases que formularon los usuarios en el inciso anterior fueron analizadas, teniendo como resultado que seguían las mismas estructuras sintácticas definidas para el ejemplo, por lo cual sólo se necesita agregar las palabras que no se encuentran en el sub-lenguaje y la información requerida por cada una de ellas (por ejemplo las ofertas).

Con lo que se comprobó que el sistema es lo suficientemente robusto si los usuarios no se salen del contexto que maneja, además este último puede ser expandido si así se requiere (como lo fue en este caso).

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

CONCLUSIONES

Como se mencionó al inicio, el objetivo general era obtener una interfaz verbal para un agente conversacional. En la etapa de desarrollo del presente trabajo, se encaminó la solución al diseño de una metodología, la cual es presentada a lo largo de este escrito. Dicha metodología se enfoca al procesamiento del lenguaje natural y la generación de la respuesta adecuada por parte del agente, lo cual se presenta en el quinto capítulo. Como se utilizaron herramientas para realizar el reconocimiento de voz y la síntesis de texto a voz (discutidas en el capítulo cuatro), se pudo enfocar los esfuerzos en la metodología presentada.

El tipo de interfaces verbales que pueden ser diseñadas con la metodología, son las que se buscaba lograr obtener desde el inicio, por lo que se logró el objetivo principal. Pero además de un tipo de interfaz verbal enfocado a una aplicación en específico, se analizó que dicha metodología también ayuda a obtener interfaces verbales para otro tipo de aplicaciones.

La forma de solucionar el problema, obteniendo la metodología mencionada en lugar de una aplicación dedicada, permite que la interfaz pueda ser utilizada en cualquier sistema operativo, y por lo mismo pueda ser programada en el lenguaje requerido. Además, por el mismo motivo y por separar el diseño de la metodología de la elección de las herramientas, se pueden utilizar cualesquiera que cumplan con las características necesarias mencionadas.

La metodología muestra la limitante de realizar el procesamiento solamente dentro de un contexto seleccionado, además como es la conjunción de varias ideas, se tuvo que adecuar cada una para llegar a obtener un equilibrio entre éstas, lo que ocasionó que no encaje con algunas de las ideas surgidas en trabajos anteriores. También el contexto que maneja no es tan amplio, ya que esto provoca que el diseño de la interfaz deba tener más consideraciones, con lo que la metodología, además de incrementar su complejidad, puede acarrear errores al reconocer la frase. Otro de los problemas de la metodología es que está enfocada a un solo contexto, por lo que no pueden convivir más en una sola aplicación.

Una de las consideraciones que no se mostraron en el presente trabajo, fue que existen varias propuestas de expresiones léxicas, las cuales pueden ser utilizadas en el árbol de estructuras sintácticas (discutido en el capítulo cinco), estas propuestas ayudan a ver los tipos de palabras mencionados para dicho árbol, facilitando la clasificación. Ahora que para integrar más de un contexto en las interfaces, se deben analizar otros trabajos desarrollados para integrar esta característica dentro de la metodología.

Lo alcanzado en el presente trabajo cumple con una etapa para obtener un agente conversacional, donde aún faltan detalles que quedan como un trabajo a futuro como mejorar el diseño de las interfaces verbales agregando diversas características, de las cuales se tienen en mente el poder utilizar conjunciones para unir dos o más frases y poder analizarlas como una sola. También se tiene en mente poder ampliar los contextos seleccionados, además de utilizar más de uno.

Aparte de este tipo de problemas, también se tienen en mente el agregar la representación gráfica del agente (ya que la mostrada en el ejemplo no es una definitiva), lo cual implica otros problemas, como el movimiento facial o la interacción del agente en el ambiente virtual. Aunque éstos se están trabajando de manera separada al diseño de la interfaz verbal, por lo que en un futuro también se pretende unir todas estas piezas para obtener un sistema completo.

Literatura citada.

1. Ríos H., Solís A. L., Guerrero L., Peña J, Santamaría A.: Facial Expression Recognition and Modeling for Virtual Intelligent Tutoring Systems. Memorias del Mexican International Conference on Artificial Intelligence, Acapulco México (2000): pp 115-126
2. Cavazza M., Palmer I., Parnell S.: Real-Time Requirements for the Implementation of Speech-Controlled Artificial Actors. Proceedings of the Fourth International Workshop on Modeling and Motion Capture Techniques for Virtual Environments, Londres Inglaterra (1998): pp 187-198
3. Cassell J., Sullivan J., Prevost S., Churchill E.: Embodied Conversational Agents. The MIT Press, Massachusetts USA (2000).
4. Prevost S.: Modeling Contrast in the Generation and Synthesis of Spoken Language. Proceedings of the International Conference on Spoken Language Processing, Massachusetts USA (1996): pp 1349-1352
5. Moser M., Moore J. D.: Investigating Cue Selection and Placement in Tutorial Discourse. Proceedings of the 33rd Conference of Association for Computational Linguistics, Massachusetts USA (1995): pp 130-135
6. Heeman P. A., Byron D., Allen J. F.: Identifying Discourse Markers in Spoken Dialog. Papers from the AAAI Spring Symposium on Applying Machine Learning to Discourse Processing, Stanford USA (1998): <http://www.cse.ogi.edu/~heeman/papers/98.aaais.pdf>
7. Seneff S., Husley E., Lau R., Pao C., Schmid P., Zue V.: Galaxy-II: A Reference Architecture for Conversational System Development. Proceedings of the International Conference on Spoken Language Processing, Sydney Australia pp. 931-924 (1998): pp 931-934
8. Nagao K., Rekimoto J.: Speech Dialogue with Facial Displays: Multimodal Human-Computer Conversation. Proceedings of the 32nd Conference on Association of Computational Linguistics, New Mexico USA (1994): pp.102-109
9. Cassell J.: Embodied Conversational Agents:Representation and Intelligence in User Interface. AI Magazine Volume 22, California USA (2001) : pp 67-83

10. Jurafsky D., Martin J. H.: Speech and Language Processing And Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, New Jersey USA (2000)
11. Contreras G.: Visión artificial (2): Interfaces Inteligentes. Robotiker Volumen 10 (2005) : <http://revista.robotiker.com/articulos/articulo56/pagina1.jsp>
12. Balsas J. R., Díaz M. C., Montejo A., Martínez F., García M., Ureña L.A. : Arquitectura para agentes de interfaz inteligentes: el ordenador sugerente. Memorias del Interacción' 2000, Granada España (2000) : pp 74-79
13. Pellom B., Hacioglu K. : SONIC : The University of Colorado Continuous Speech Recognizer. Reporte Técnico, Colorado USA (2004)
14. Black A. W., Taylor P., Caley R. : The Festival Speech Synthesis System. Documentación del Sistema, Edinburgh Inglaterra (1999)
15. Woodland P. C., Gales M. J. F., Pye D., Young S.J. : The Development of the 1996 HTK Broadcast News Transcription System : Cambridge University Engineering Department, Cambridge Inglaterra (1996)
16. Ramírez R. Solís A. L.: Un Sistema de Diálogo para Agentes Conversacionales en Laboratorios Virtuales. IX Ibero-American Workshops on Artificial Intelligence. Puebla Mexico (2004): pp 426-434