Capítulo 2

CU Communicator

Capítulo 2 CU Communicator

2.1 Introducción a CU Communicator

La Universidad de Colorado desarrolló CU Communicator un sistema de diálogo hablado vía telefónica para poder reservar boletos de avión, hotel, y alquiler de coches. Los usuarios utilizan este sistema usando un teléfono y mediante un lenguaje natural para interactuar con este agente de viajes automatizado. El agente entonces tiene acceso al Internet para ver la disponibilidad de vuelos, hotel, y la información del alquiler de coches [communicator 02].

El sistema del CU Communicator es un ejemplo de la investigación actual que conduce a los sistemas de diálogo avanzados permitiendo una interacción humano–computadora más robusta. La robustez y la portabilidad de los sistemas de diálogo hablados, son dos de las partes más importantes en el CU Communicator. Se utilizan estrategias robustas para el control del análisis y del diálogo para que sean lo más flexibles posible dada la variación del habla y de la manera de interactuar del usuario.

2.1.1 Arquitectura del CU Communicator

El CU Communicator utiliza la arquitectura Galaxy, la cual es una arquitectura modular [Pellom et al 00]. Esto significa que el sistema está compuesto de un número de servidores que interactúan entre ellos a través del sistema de DARPA Hub mostrados en la figura 1.1. Se compone de un Hub y de ocho servidores:

- Servidor de Audio (Audio client): Recibe señales del teléfono y las envía al reconocedor. Envía discurso sintetizado al teléfono.
- Servidor Reconocedor de voz (Speech Recognizer): Toma la señal del servidor de audio y producen una cadena texto.
- Servidor de Análisis Semántico (Parser): La cadena de texto es tomada del reconocedor y se trata de producir la mejor interpretación del texto.
- Servidor de Manejo del Diálogo (Dialogue Manager): Resuelve ambigüedades en la interpretación, estima confianza en la información extraída, Le solicita al usuario más datos si es necesario, construye los queries para la base de datos (SQL), envía datos al generador de lenguaje natural para la presentación al usuario.
- Servidor de Bases de Datos (Database): Recibe las queries del manejador de diálogo, hace las interfaces a la base de datos SQL y devuelve la información solicitada, recupera datos en tiempo real del WEB.
- Servidor Generador del Lenguaje (Natural Language Generator): Genera cadenas de texto las cuales van a ser enviadas al sintetizador de voz.
- Servidor de Síntesis de voz (TTS: Text to Speech): Recibe la cadena de texto del generador de lenguaje natural y la sintetiza para enviársela al servidor audio.

- Servidor de Perfil del usuario: Permite que los usuarios incorporen el perfil de las preferencias que es utilizado por el manejador del diálogo para proveer los valores prefijados para las líneas aéreas, los hoteles, los coches, el etc.

El CU Communicator tiene un control de flujo como se puede ver en la Fig. 2.1.

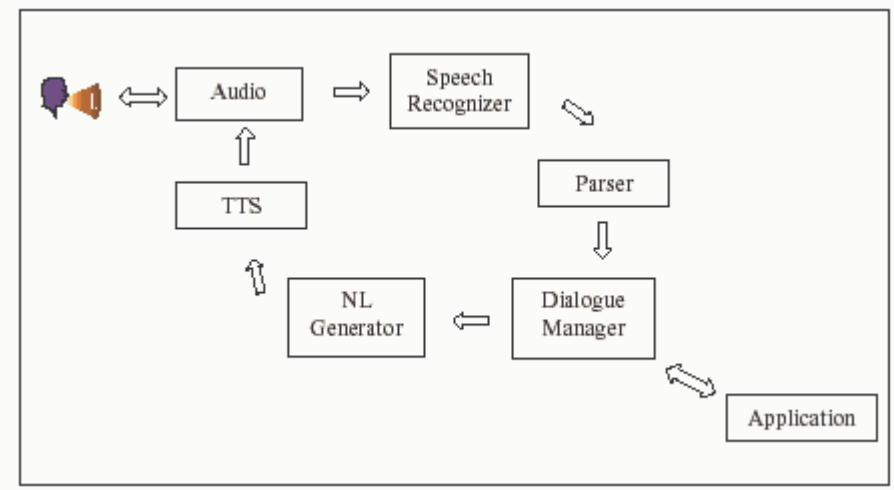

Fig. 2.1 Diagrama de control lógico del sistema Communicator 02

2.2 Servidor de Audio (Audio client)

El servidor de audio es responsable de contestar las llamadas que entran, reproducir y grabar la entrada y salida del audio. El sistema utiliza el servidor de audio MIT/MITRE el cual fue proporcionado por los participantes del programa DARPA (Defense Advanced Research Project Agency) Communicator. El hardware de telefonía consta de un módem serial el cual es conectado a la entrada del micrófono y a la terminal de salida del audio. El audio es grabado y luego es preprocesado por el reconocedor de voz, finalmente es preprocesado por el sintetizador de voz y luego enviado al servidor de audio para ser reproducido.

Recientemente se ha desarrollado un nuevo servidor de audio, el cual tiene la capacidad de soportar "barge-in" utilizando la plataforma del hardware "Dialogic". Este nuevo servidor de audio trabaja bajo un ambiente Linux, implementa la cancelación de la salida de audio, todo basado en software. Este servidor puede trabajar en cualquier plataforma "Dialogic".

La manera en que trabaja el servidor de audio es en el momento que se presiona el botón de iniciar audio que está disponible en la interfaz de audio del proceso de "cliente audio", inicializará el servidor de audio para que empiece a capturar o reproducir el audio. Consecuentemente, el servidor audio informa al "Hub" que el audio está disponible, y el "Hub" informa al reconocedor para comenzar a procesar. Véase Fig. 2.2.

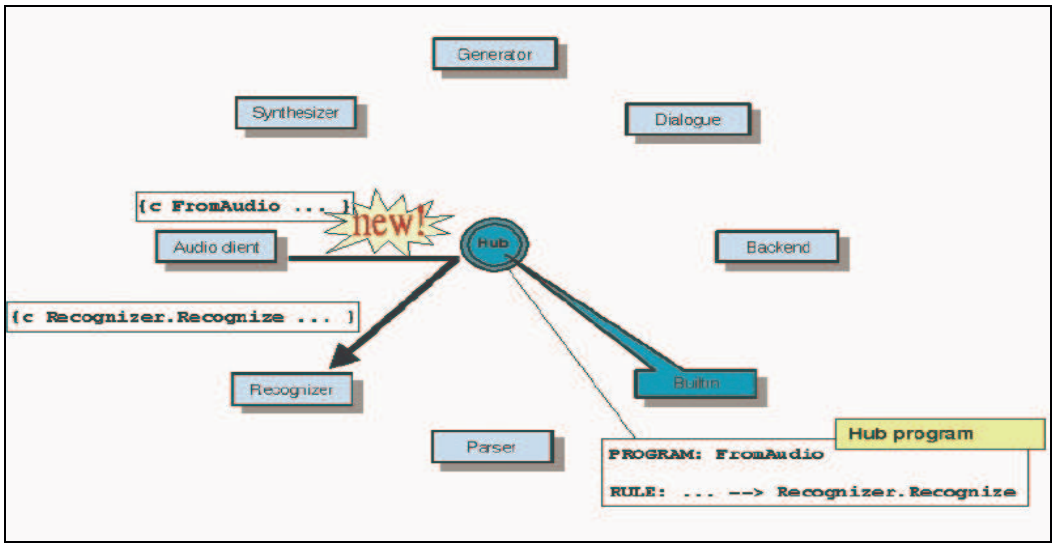

Fig. 2.2 Proceso de comunicación entre el servidor de audio (Audio client) y el reconocedor de voz (Recognizer) mediante el Hub [Darpa Communicator team 02].

Como se puede ver en la figura 2.2 el servidor informa al Hub que ha recibido un nuevo mensaje. El Hub construye un token de este nuevo mensaje. Este nuevo token contiene siete pares de valores:

- El host de la máquina por la cual se conecta
- El puerto del servidor de audio.
- El id que envía el servidor de audio al Hub para que sepa que operación realizar.
- La frecuencia del audio.
- El formato de codificación del audio.
- El id de la secuencia que sigue el Hub.

2.3 Servidor de reconocimiento de voz (Recognizer)

Actualmente el CU Communicator (la versión que está disponible vía Web) utiliza el "SphinxII" desarrollado por la Universidad de Carnegie Mellon para el servidor de reconocimiento de voz [Ravishankar 96]. Este es un reconocedor semi-continuo de Modelos Ocultos de Markov con un modelado del lenguaje de tri-gramas. El reconocedor recibe un vector de entrada enviado del servidor de audio. El servidor de reconocimiento produce una cadena de palabras de donde se elige la mejor hipótesis.

De una forma mas detallada el reconocedor procesa todo el audio. Éste envía un mensaje con los resultados obtenidos en forma de texto. Después se utiliza la función `Builtin.call_program` para invocar un nuevo programa al Hub, la cual unifica el proceso de la entrada de texto, véase Fig. 2.3.

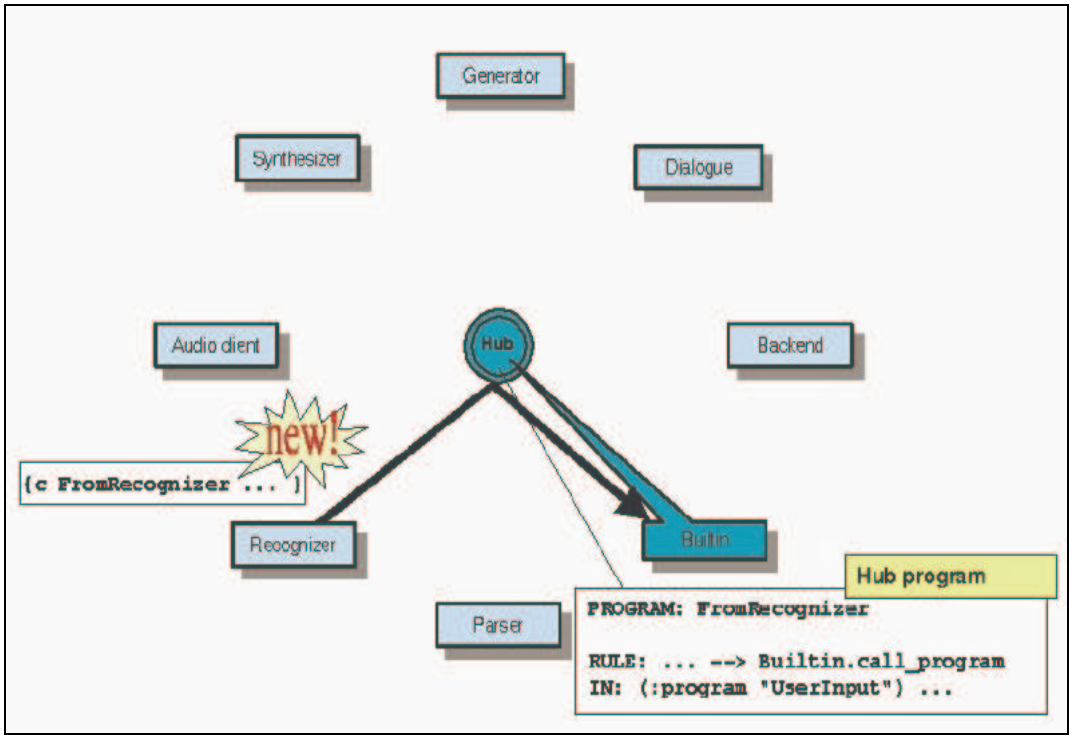

Fig. 2.3 Forma en que interactúa el servidor de reconocimiento de voz (Recognizer) con el Hub [Darpa Communicator team 02].

2.4 Servidor de Confianza (Confidence)

Este servidor tiene como principal función la de considerar la detección y rechazo de errores de reconocimiento a nivel palabra y frases fuera del dominio usando el modelado de lenguaje [San-Segundo et al 00]. Recientemente se ha considerado la detección y rechazo de unidades micro-reconocidas por el nivel de concepto. Debido a que el concepto es utilizado para actualizar el estado del sistema de diálogo, el nivel de confianza del concepto es vital para lograr una exitosa interacción humano-computadora.

2.5 Servidor de Análisis Semántico (Parser)

El CU Communicator utiliza la versión más reciente del Parser Phoenix para mapear la salida del reconocedor en una secuencia de frames semánticos. El parser está diseñado para el desarrollo simple y robusto de interfaces de lenguaje natural a aplicaciones, especialmente en las aplicaciones del lenguaje hablado. Dado que habla espontánea es continuamente malformada y también porque el reconocedor a veces reconoce errores, es necesario que el parser sea robusto a errores en reconocimiento, gramática y fluidez.

La manera en que el parser interactúa con el Hub es cuando el parser recibe un mensaje enviado por Builtin.call_program, llamado UserInput, reenvíando este mensaje al manejador de diálogo (Dialogue Manager) véase Fig. 2.4.

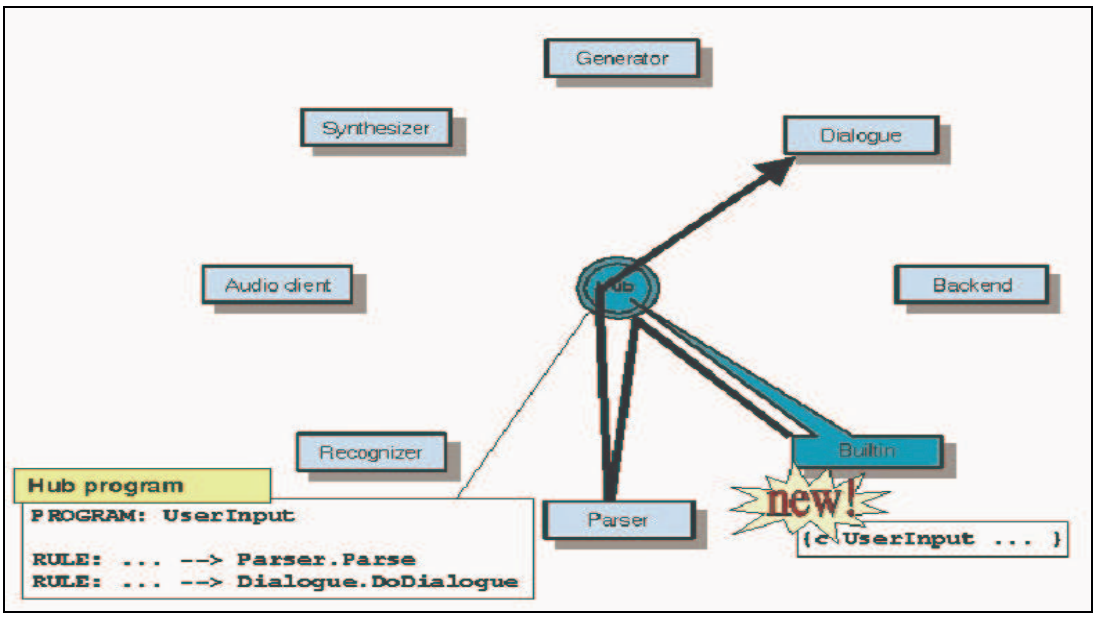

Fig. 2.4 El analizador semántico (Parser) envía un mensaje al manejador de diálogo (Dialogue Manager)

[Darpa Communicator team 02].

En cada punto donde el Hub recibe un mensaje de respuesta imprime el estado de la señal y actualiza su estado, entonces se podrá obtener el seguimiento de la secuencia desde la salida del Hub.

2.6 Servidor de Manejo de Diálogo (Dialogue Manager)

El manejo de diálogo controla la interacción del sistema con el usuario y la aplicación del servidor. Es responsable para decidir que acciones debe realizar el sistema en cada paso de la interacción.

El manejador de diálogo es el encargado de varias funciones:

- Resuelve la ambigüedad en la interpretación de los eventos.
- Estima la confidencia al extraer información.
- Clarifica la interpretación con el usuario si es requerido.
- Integra nuevas entradas con el contexto de diálogo.
- Construye consultas (queries) para la base de datos.
- Envía datos al generador de lenguaje natural para la presentación del usuario.

El manejador de diálogo es flexible para manejar cualquier evento del contexto del sistema. El manejador de diálogo recibe entrada desde el parser y envía una salida al generador de lenguaje natural. Éste envía y recibe información desde la aplicación o la base de datos. El manejador de diálogo toma el paso final de interpretar la expresión e integrar esto dentro de un contexto del diálogo. El manejador de diálogo controla la

interacción del sistema con el usuario. Eso decidirá qué es lo que el sistema debe hacer o lo siguiente a decir.

El manejador de diálogo está configurado para operar como un servidor en la arquitectura del Hub DARPA. La interacción con otros servidores es llevada a cabo por el envío de frames mediante el Hub.

El manejador de diálogo organiza la producción del sistema. Eso es normalmente en un estado inactivo esperando la entrada de un evento. Cuando recibe una entrada, envía algunos frames al Hub, y entonces regresa a un estado inactivo.

El manejador de diálogo cuando interactúa con otros servidores tiene diversas funciones:

- Recibe la cadena de entrada desde el Servidor de análisis semántico

  - Valoración de la elipsis.
  - Clarificación.
  - Actualización del contexto.

- Envía peticiones al generador de lenguaje natural

  - Información de salida para el usuario.

- Genera los queries para la base de datos y los envía al servidor de bases de datos.
- Recibe los resultados de la base de datos desde el servidor de bases de datos.

Los frames son una estructura básica de información para representar un dominio de información. Un frame tiene un nombre y un número fijo de slots. Cada slot es un

concepto jerárquico con el nombre del slot como la raíz como se puede ver en la Fig. 2.5. Existe una librería de funciones para la manipulación de los frames. La información es extraída en pequeñas partes dentro de los frames y es almacenada en los frames directamente por el manejador de diálogo. Idealmente, el concepto de estructura en este frame es igual al producido por el parser. En este caso, la extracción desde el parser al frame del manejador de diálogo es directa.

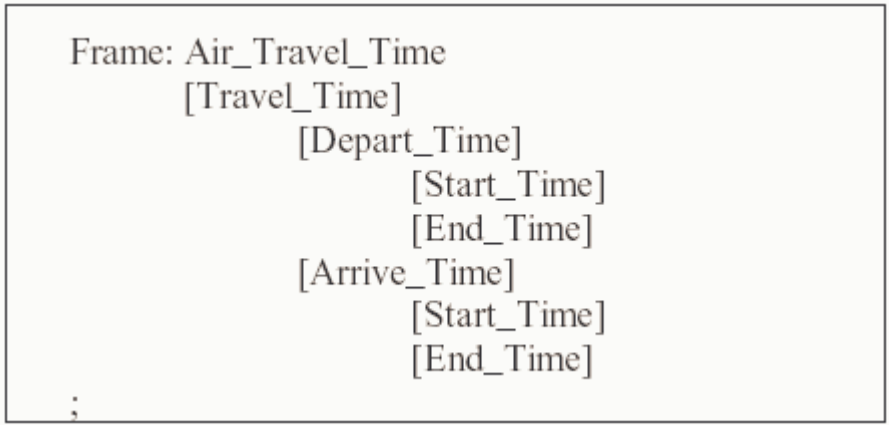

Fig. 2.5. Estructura de un frame donde Air_Travel_Time es el nombre del frame y los slots son todos los demás campos los cuales son llenados con la información proporcionada por el usuario.

2.7 Servidor de Base de Datos

Una de las últimas partes de las interfaces consiste en una base de datos SQL y scripts en PERL que accedan información desde Internet. Durante las operaciones, las peticiones a la base de datos son transmitidas desde el manejador de diálogo hasta el servidor de bases de datos por medio de un frame con formato presentado en la Fig. 2.5.

Esta última parte consiste en componentes de información dinámica y estática. Las tablas estáticas contienen información tal como conversiones entre 3 diferentes listas de aeropuertos, de las ciudades, estados y países. Hasta hace unos años se encontraban más de 8000 aeropuertos en la base de datos, 200 cadenas de hoteles y 50 compañías de renta de autos.

En el momento en que se recibe una petición para la base de datos, se utiliza el comando SQL desde el manejador de diálogo para seleccionar algunos registros desde la memoria local, en caso de no encontrarse los registros dentro de las máquinas se realiza una petición vía Internet para obtener todos los datos, y una vez que fueron obtenidos se insertan en una base de datos local ver Fig. 2.6.

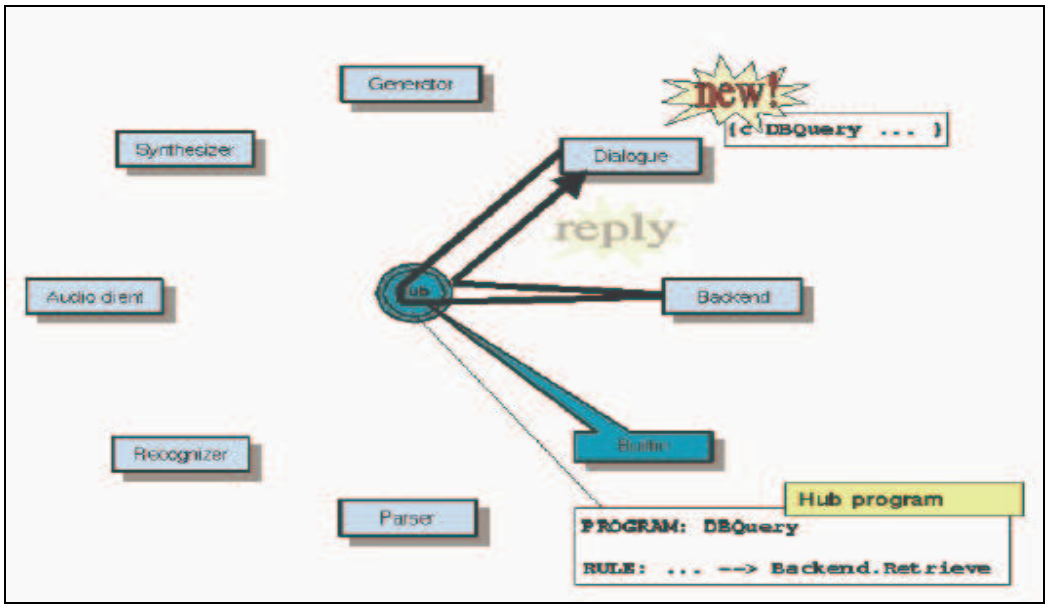

Fig. 2.6 Respuesta del servidor Backend al manejador de diálogo (Dialogue Manager) basándose en un query formulado [Darpa Communicator team 02].

La interfaz para Internet es diseñada mediante scripts en PERL específicos del dominio con extensiones HTML. Estos scripts se encargan de realizar las conexiones con el sitio Web en el que se encuentra toda la información de los viajes. El tiempo de acceso es de 3 a 6 segundos en promedio, este tiempo también dependerá en el día y hora en la que se conecten.

2.8 Servidor de Síntesis de Voz

Para la salida del audio, se ha desarrollado un dominio dependiente concatenado de síntesis de voz. La síntesis concatenada puede adherirse en unidades de orden desde fonemas, palabras o frases. Cada declaración es ortográficamente transcrita y fonéticamente alineada usando el HMM- basado en el reconocedor. Los esfuerzos de investigación para la colección de datos actualmente fija su atención en métodos para reducir la distorsión audible para los segmentos límite y esquemas de optimización para la generación de los límites, así como herramientas para agilizar la corrección de la desalineación de los límites.

El sintetizador de voz recibe un mensaje desde el manejador de diálogo en el momento en que decide que es necesario decirle algo al usuario. Entonces envía un nuevo mensaje al Hub, el cual es enviado al servidor generador que se encarga de volverlo a enviar al sintetizador. Ver Fig. 2.7.

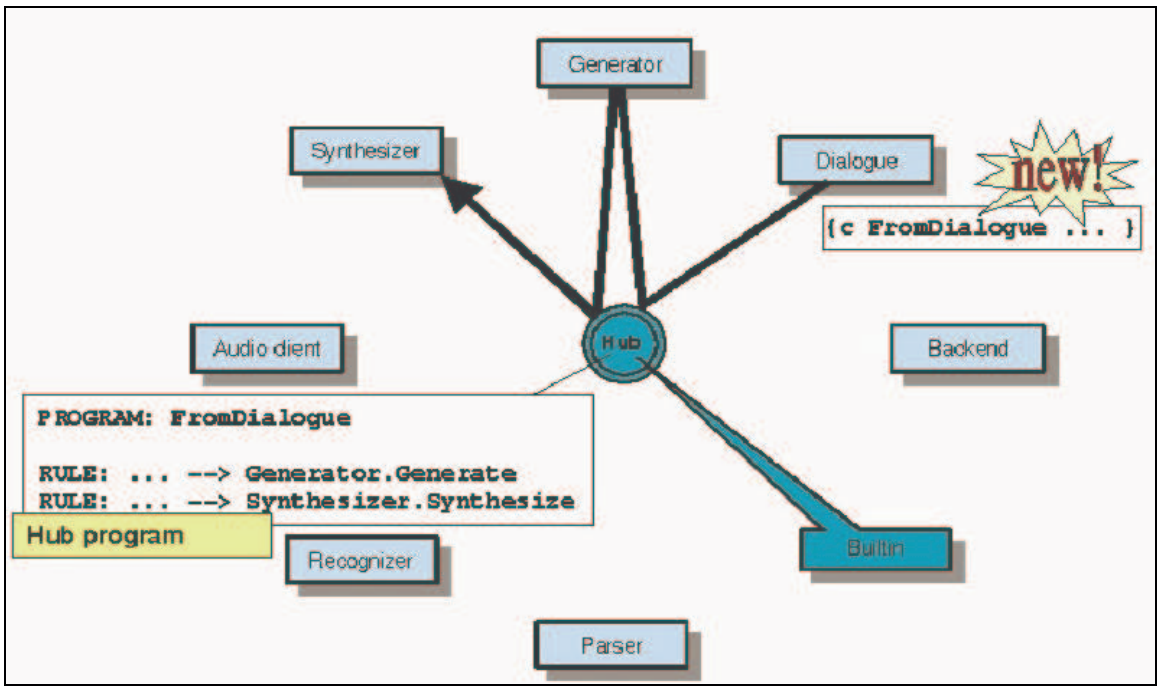

Fig. 2.7 El manejador de diálogo(Manager Dialogue) envía un mensaje al sintetizador (Synthesizer).

[Darpa Communicator team 02].

2.9 Servidor del Perfil del Usuario

Para poder diferenciar los múltiples perfiles de usuario que existen se utiliza una página Web para el manejador de diálogo que es el encargado de la mayor parte de la interacción con el usuario. En esta página le permite al usuario personalizar sus diálogos, o sea la forma en que son ordenados los resultados obtenidos.

2.10 Conclusión

En este capítulo se describió el funcionamiento específico de cada servidor y la forma que interactúan entre ellos bajo la arquitectura Galaxy con el objetivo de analizar los protocolos utilizados entre el Hub y los diferentes servidores para establecer una comunicación entre ellos. De esta manera se obtuvo el tipo de datos que necesitaba recibir el nuevo servidor implementado “conexión”, para la transmisión del texto y el audio, los protocolos usados para tener comunicación con el Hub y con otros servidores.