

## 6.1 Introducción

El gestor de diálogo es, sin lugar a dudas, el módulo más importante de un Servidor Vocal Interactivo. Este módulo es el encargado de gestionar los recursos ofrecidos por el resto de módulos para dirigir la interacción del sistema con el usuario. La importancia de este módulo se debe a que el diálogo es la ventana a través de la cual los usuarios perciben el comportamiento y la aparente “inteligencia” del sistema.

En este tema describiremos brevemente los modelos de diálogo utilizados en la literatura, comentando sus ventajas e inconvenientes, para pasar después a detallar la metodología de diseño de gestores de diálogo propuesta en esta tesis doctoral. Esta metodología se presenta mediante su aplicación al diseño de un gestor de diálogo para un servicio de información y reserva de billetes de tren. La metodología propuesta está formada por 5 etapas. La primera etapa es el Análisis de la Base de Datos en la que se describe, mediante un diagrama Entidad-Relación (E-R), el contenido de la base de datos utilizada para dar el servicio. En la etapa de diseño por Intuición se realiza un “brain-storming” sobre el diagrama E-R para proponer diferentes alternativas de diálogo. El tercer paso es el diseño por observación, donde se evalúan las transcripciones de diálogos usuario–operador. El siguiente paso consiste en simular el sistema mediante un Mago de Oz con el fin de aprender las características específicas de las interacciones entre un sistema automático y el usuario. El quinto y último paso es el diseño por Mejora Iterativa en el que el sistema entra en un proceso de prueba (por usuarios reales) y mejora constante. El servicio implementado en este trabajo se ha realizado sobre el entorno de desarrollo TADE (ver apéndice A, apartado A.2). Aunque en este entorno se dispone únicamente de reconocimiento de voz para palabras o expresiones pronunciadas de forma aislada, la metodología que se va a presentar es más general y puede ser aplicada a servicios con reconocimiento de habla continua y comprensión. La metodología propuesta es similar al Modelo Life-Cycle descrito por Bernsen (Bernsen, 1998; [www.disc2.dk](http://www.disc2.dk)). En nuestro caso incorporamos un nuevo paso de diseño por observación donde se analizan conversaciones usuario–operador, y se presentan medidas para evaluar las diferentes alternativas de diseño en cada paso de la metodología. En la bibliografía (Brown, 1998; IBM, 1999; López, 2001) podemos encontrar gran cantidad de metodologías propuestas para el diseño de software en general como son el diseño en cascada, formado por varias fases seguidas linealmente, o el diseño por prototipado en el que se implementa un prototipo inicial sobre el que se definen los requisitos del sistema a desarrollar. Algunas de estas ideas pueden ser reutilizadas para el diseño de Servidores Vocales Interactivos, pero estas metodologías deben ser ampliadas puesto que en el desarrollo de estos sistemas aparecen aspectos nuevos que se deben tener en cuenta:

- Para el funcionamiento de un Servidor Vocal Interactivo se debe hacer uso de módulos de reconocimiento y comprensión que tienen un comportamiento muy diferente a cualquier sistema software convencional. En primer lugar sus tasas de error son mucho mayores, lo que supone una limitación importante de funcionamiento. Y además, los errores que comenten estos módulos casi nunca

son predecibles ni se conocen con exactitud, lo que imposibilita que se pueda devolver un código de error en caso de mal funcionamiento.

- La limitación tan fuerte impuesta por el funcionamiento de estos módulos obliga a que el diseño del servicio requiera de mucho esfuerzo encaminado a evitar que estos módulos, con tasas de funcionamiento no muy elevadas, funcionen razonablemente bien aun en condiciones adversas. Este esfuerzo llega a ser tan importante como el llevado a cabo para que el servicio se ajuste a las necesidades reales de los usuarios. En muchos casos estas limitaciones obligan a reducir la funcionalidad del servicio no pudiendo cubrir más que alguna parte concreta.

En este tema también se describen dos aspectos importantes en la gestión del diálogo: la incorporación de medidas de confianza (obtenidas del módulo de reconocimiento) para la gestión eficiente de las confirmaciones de datos, y la implementación de técnicas de modelado de usuario para adaptar el diálogo a su destreza en la interacción con el sistema.

A lo largo de este capítulo también se harán comentarios sobre el diseño de las frases a pronunciar por el sistema. Si bien esta parte corresponde al módulo de generación de respuesta, su estrecha relación con el diálogo hace inevitable describir su diseño de forma paralela. En nuestro caso, el generador de respuesta utilizado está basado en plantillas o estructuras de subfrases genéricas en las que se definen partes variables que se completan dependiendo del estado e historia del diálogo. Para generar las frases de salida se completan las partes variables de cada una de las subfrases a utilizar, uniéndolas posteriormente para generar la frase final.

### 6.1.1 Definiciones

En este subapartado comentaremos las definiciones de algunos conceptos importantes para poder entender mejor el desarrollo del capítulo. La razón de incluir estas definiciones es que muchos de estos términos pueden ser ambiguos puesto que tienen interpretaciones diferentes en otros entornos de trabajo. Los conceptos son los siguientes:

- **Interacción usuario–sistema:** proceso a través del cual el usuario aporta información al sistema o viceversa. En el caso de aportación del usuario al sistema, es necesario disponer de sistemas de reconocimiento y comprensión de lenguaje natural para entender lo que dice el usuario, y en el sentido contrario, el sistema debe disponer de módulos para la generación de respuesta y la síntesis de voz.
- **Turno de diálogo:** par de interacciones: pregunta del sistema–respuesta del usuario, para el caso de iniciativa del sistema, o par: pregunta del usuario–respuesta del sistema para los casos en los que la iniciativa la lleve el usuario.
- **Diálogo:** conjunto de turnos necesarios para que el SVI ofrezca un determinado servicio al usuario.

- **Subdiálogo:** subconjunto de turnos que forman parte del diálogo. Un subdiálogo suele agrupar turnos de diálogos relacionados entre sí: por ejemplo los turnos para la petición de un dato y su posterior confirmación, o los turnos necesarios para que el sistema informe de las opciones posibles de viaje.
- **Iniciativa del diálogo:** acción de dirigir la progresión del diálogo. Una clasificación característica de los gestores de diálogo (capítulo 2) se basa en las posibilidades que dicho gestor ofrece para que sea uno u otro agente, usuario o sistema, el que pueda dirigir dicha progresión.
- **Objetivo:** cada funcionalidad independiente ofrecida por el Servidor Vocal Interactivo (SVI). El usuario del sistema podría acceder a dicha funcionalidad sin conocer la existencia del resto de posibilidades. Ejemplos de objetivos en un servicio de información ferroviaria podrían ser: la consulta de los horarios de tren o la realización de la reserva del viaje. Aunque dos objetivos sean independientes pueden tener turnos de diálogos comunes: tanto para pedir información de horarios como para hacer una reserva es necesario especificar las ciudades origen y destino del viaje. Dado que los objetivos son independientes, la finalización (o satisfacción) de un objetivo suele llevar consigo la consulta del sistema a una base de datos para obtener (información de horarios) o insertar (hacer una reserva) cierta información.
- **Subobjetivo o paso del diálogo:** subconjunto de turnos de diálogo dentro de los necesarios para satisfacer un objetivo. Con este subconjunto de turnos no es posible ofrecer una funcionalidad completa e independiente, es necesario completar el objetivo en su totalidad para ofrecerla. La finalidad de esta agrupación es organizar conceptualmente los turnos de diálogo cuando el objetivo considerado sea muy complejo. De esta forma, como veremos más adelante, es posible definir zonas de descanso entre los diferentes pasos del diálogo que pueden ser aprovechadas por el sistema para hacer un resumen de la interacción, ofrecer cierta ayuda al usuario o hacer confirmaciones de varios datos simultáneamente. Por ejemplo para el objetivo de hacer una reserva de tren, si queremos reconocer la fecha del viaje con un reconocedor de habla aislada, son necesarias varias interacciones. Este conjunto de interacciones pueden ser agrupadas en un mismo subobjetivo o paso del diálogo.

Este concepto está íntimamente relacionado con el concepto subdiálogo de forma que la agrupación conceptual de turnos dentro de un objetivo (subobjetivo) se implementa con un conjunto de turnos que forman un subdiálogo. La diferencia entre ambos conceptos es que el subobjetivo está formado por un conjunto de turnos pertenecientes al mismo objetivo, mientras que el concepto subdiálogo es más general y puede estar formado por turnos que incluyen varios objetivos.

- **Dato del usuario:** cada elemento de información que el usuario debe facilitar al sistema para que éste pueda satisfacer el objetivo solicitado, es decir, pueda ofrecer la funcionalidad requerida por el usuario.

- **Dato del sistema:** cada elemento de información que el sistema ofrece al usuario. Esta información puede formar parte del proceso de satisfacción del objetivo: dar los horarios de tren o el código de la reserva del viaje, o se ofrece al usuario para informarle de alguna circunstancia de la interacción: confirmación de datos y posibilidades de corrección o ayuda.
- **Servicio:** es el conjunto de funcionalidades (objetivos) ofrecidos por el SVI al usuario. Cuando utilicemos la palabra servicio estaremos haciendo referencia al conjunto de todos los objetivos que el sistema puede satisfacer.
- **Negociación:** conjunto de turnos de diálogo a través de los cuales el usuario acepta o rechaza las propuestas del sistema, añadiendo y/o cambiando algunas restricciones (valores de los datos del usuario). Esta negociación se produce cuando existen varias opciones que se ajustan a los datos especificados por el usuario, o cuando no existe ninguna opción posible. Por ejemplo, cuando el sistema obtiene varias opciones de viaje entre dos ciudades en una fecha determinada y el usuario debe elegir una para poder hacer la reserva, o cuando no existe ninguna y el usuario debe cambiar o relajar alguna de sus condiciones con el fin de obtener alguna.
- **Estrategia de diálogo:** reglas generales que rigen el diseño y el funcionamiento del gestor del diálogo. Estas reglas se derivan de los aspectos ergonómicos considerados en la interacción y de las funcionalidades a ofrecer por el servicio que se quiere desarrollar. En el apéndice F, se presentan algunas recomendaciones a tener en cuenta en el diseño del diálogo. Algunas de estas recomendaciones pueden no ser factibles simultáneamente, por ello, el considerar de forma prioritaria unas en lugar de otras, define la estrategia de diálogo utilizada.
- **Modelo de diálogo:** representación formal del proceso de diálogo entre usuario y sistema. Esta representación está formada por tres elementos:
  - El primero es una estructura de datos en la que se presentan todos los posibles turnos de diálogo. Pueden existir diferentes tipos de turnos de diálogo: turnos en los que el sistema pregunta algún dato al usuario, le pide confirmación de algún dato, responde a alguna pregunta del usuario, o hace una consulta a la base de datos para obtener cierta información. Estos turnos pueden estar prefijados inicialmente o se pueden ir generando a medida que avanza el diálogo. Para cada turno se debe guardar en una estructura cierta información característica que dependerá del tipo de turno representado. Por ejemplo, la pregunta a formular y el diccionario de reconocimiento activo en el caso de un turno para preguntar un dato, las plantillas para la generación de la respuesta en el caso en el que el sistema quiera ofrecer cierta información al usuario, el dato a confirmar en el caso de un turno de confirmación, etc...
  - El segundo elemento necesario es una estructura de datos en la que se refleje el estado actual y la historia del diálogo: objetivo o funcionalidad solicitada

por el usuario, secuencia de turnos realizados hasta ahora, conjunto de datos obtenidos del usuario, conjunto de datos confirmados, etc...

- El tercer elemento es el algoritmo que define la evolución entre los diferentes turnos del diálogo. Este algoritmo debe utilizar la información contenida en la estructura de datos que define la historia y el estado actual del diálogo, para decidir el siguiente turno a ejecutar. En el caso de que estos turnos se vayan generando dinámicamente, este algoritmo debe definir este proceso de generación.

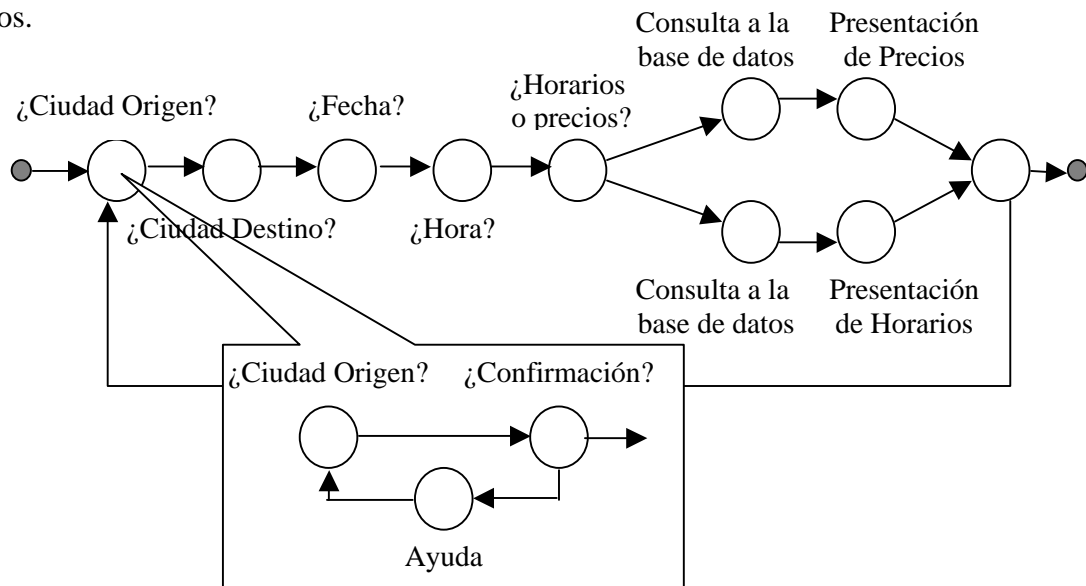
En el apartado 6.2 se describen diferentes alternativas para la representación formal del diálogo (modelos de diálogo).

## 6.2 Modelos de diálogo

En este apartado vamos a describir los modelos de diálogo más habituales para el desarrollo e implementación de gestores de diálogo en SVIs. En primer lugar describiremos el modelo basado en un grafo o autómata de estados finito y después comentaremos los modelos basados en una representación semántica de la tarea que ofrecen mayores prestaciones.

### 6.2.1 Modelo basado en un autómata de estados finito

En este caso el modelo de diálogo se basa en una representación del diálogo como secuencia de estados formando un grafo dirigido. En la figura 6-1, podemos ver el grafo o autómata para el caso de un servicio con dos objetivos: consulta de información de horarios y precios para viajes en tren. En esta figura se representan los estados principales sin entrar en demasiado detalle. Para cada uno de estos estados se podría hacer una descripción más extensa que implicaría la definición de un mayor número de estados.



**Figura 6-1:** Ejemplo de grafo para un servicio de información de horarios y precios de tren.

Veamos cómo se definen para este caso cada una de las tres partes que forman un modelo de diálogo:

- 1) **Representación de los turnos de diálogo.** En este caso, cada turno del diálogo está asociado a un estado o nodo del autómata. En cada nodo se guarda una estructura de datos con información sobre el turno de diálogo asociado:
  - Tipo de turno: obtención de un dato del usuario, confirmación de algún dato, acceso a la base de datos, presentación de la información al usuario,...
  - Frase a pronunciar por el sistema: pregunta a realizar al usuario, frase de confirmación del dato, consulta SQL a realizar a la base de datos,...
  - Datos adicionales: dato a preguntar, dato a confirmar, información a presentar,...

En este punto conviene comentar que existen flujos de información entre los diferentes estados del diálogo. Por ejemplo cuando se pregunta un dato, el resultado de lo reconocido debe ser guardado en un estado posterior donde se realizará la confirmación de ese dato. De igual forma, cuando se confirme un dato deberá guardarse en un estado posterior para hacer la consulta SQL a la base de datos. Estos flujos de información se pueden realizar a través de estructuras de datos comunes a todos los nodos del grafo.

La representación de un autómata se suele realizar con una lista de estructuras de datos (para representar los diferentes nodos) interconectada mediante punteros, de forma que cada nodo queda identificado con un número entero. La representación de los nodos en una lista facilita su recorrido.

- 2) **Representación de la historia y del estado del diálogo.** En este caso el estado actual del diálogo queda determinado conociendo el último nodo ejecutado (o simplemente conociendo su identificador). La historia del diálogo se puede representar mediante una lista con la secuencia de nodos ejecutados. En algunos casos, como el representado en la figura 6-1, simplemente es necesario conocer si la información solicitada es de precios u horarios de tren para que la secuencia de nodos ejecutados quede definida sin ambigüedad.
- 3) **Algoritmo para decidir la secuencia de turnos.** El algoritmo simplemente debe seguir las especificaciones que le va dando el grafo definido. Conocido el nodo ejecutado, el grafo define el siguiente nodo (turno de diálogo) a ejecutar. Pueden ocurrir situaciones en las que dado un nodo, existan varias transiciones posibles. En este caso, el siguiente nodo a ejecutar dependerá del valor de un dato fácil de comprobar.

La principal ventaja de este modelo es su sencillez, una vez definido el flujo del diálogo mediante el diseño del autómata, la gestión del diálogo es muy sencilla. Simplemente hay que ir ejecutando los turnos de diálogo tal y como fueron especificados.

El mayor inconveniente es su rigidez. Los sistemas desarrollados con este modelo suelen ser gestores que permiten exclusivamente iniciativa del sistema. Si quisiéramos incorporar cierta iniciativa por parte del usuario, tendríamos que realizar las siguientes acciones dependiendo del tipo de iniciativa considerada:

- Para permitir que el usuario pudiera cambiar de objetivo en cualquier punto habría que definir transiciones desde cualquier estado de un objetivo (o funcionalidad) a los estados iniciales del resto de objetivos considerados en el servicio. Este hecho produciría un grafo con casi interconexión total muy complicado de gestionar. En algunos casos, como en el ejemplo presentado en la figura 6-1, existen varios nodos comunes a dos objetivos con lo que esos nodos no se volverían a ejecutar en el caso de cambio de un objetivo a otro.
- En el caso de que la iniciativa del usuario se quedase reducida a la posibilidad de cambiar la secuencia de datos a especificar, se podrían utilizar las estructuras comunes a todos los nodos para ofrecer esta funcionalidad. De forma que si en un mismo turno se recogen varios datos, que estos datos se almacenen en estas estructuras para que se tengan en cuenta en turnos siguientes y no se vuelvan a preguntar. Por ejemplo si en la primera pregunta (figura 6-1) el usuario nos da la ciudad origen y la de destino en la misma interacción, el segundo nodo nos le debemos saltar y no ejecutarle. En este caso estamos dando un protagonismo especial a las estructuras comunes, que como veremos más adelante, pasarán a ser piezas muy importantes en la descripción de los siguientes modelos.

Algunos ejemplos de este tipo de modelo de diálogo son los siguientes: (Sutton et al, 1998; San-Segundo et al, 1999; Baggia et al, 2000; Córdoba et al, 2000; San-Segundo et al, 2000c; Córdoba et al, 2001; San-Segundo et al, 2001c; San-Segundo et al, 2001e).

## 6.2.2 Modelo basado en plantillas

Este modelo del diálogo está basado en una representación del servicio mediante plantillas de datos. Cada uno de los objetivos del servicio se asocia a una plantilla de datos diferente. Esta plantilla contiene las estructuras en las que se almacena la información sobre los datos que necesitamos preguntar al usuario para satisfacer el objetivo solicitado.

En la figura 6-2 se representan las dos plantillas correspondientes a los dos objetivos definidos en el ejemplo de la figura 6-1: consulta de precios y horarios de tren. También hemos representado un ejemplo de estructura de datos correspondiente al dato Ciudad\_Origen. En esta estructura se representan los siguientes elementos:

- NOMBRE: corresponde con el identificador que utilizará el sistema para referirse a ese dato. Por ejemplo la representación del concepto que hace alusión a ese dato.
- VALOR: será un campo vacío a rellenar por la aplicación cuando el usuario aporte ese dato.

**OBJETIVO PRECIOS**

Ciudad Origen	Ciudad Destino	Fecha	Hora	Tipo Consulta: PRECIOS
---------------	----------------	-------	------	------------------------

**OBJETIVO HORARIOS**

Ciudad Origen	Ciudad Destino	Fecha	Hora	Tipo Consulta: HORARIOS
---------------	----------------	-------	------	-------------------------

**NOMBRE:** Ciudad\_Origen  
**VALOR:** MADRID  
**PREGUNTA:** Diga la ciudad origen de su viaje.  
**PREG. CONFIRMACIÓN:** ¿Ha dicho \$Ciudad\_Origen ?  
**¿CONFIRMADO?:** Sí  
**SQL:** "where CIUDAD\_ORIGEN == \$Ciudad\_Origen"

**Figura 6-2:** plantillas de los objetivos del diálogo presentado en la figura 6-1.

- **PREGUNTA:** corresponde con la frase que debe utilizar el sistema para solicitar del usuario el dato correspondiente. En este caso hemos representado una única opción pero podríamos considerar varios tipos de frases con el fin de utilizar una u otra, según la destreza del usuario (modelado de usuario, ver apartado 6.4.5.2).
- **PREG. CONFIRMACIÓN:** frase utilizada por el sistema para pedir al usuario confirmación del dato reconocido. En esta frase habría que insertar el valor del dato anteriormente reconocido que se representa con la expresión "\$Ciudad\_Origen". En este caso hemos considerado una única frase que hace referencia a una confirmación explícita pero se podrían considerar varios tipos de frases para el caso de que se permitiesen diferentes confirmaciones.
- **¿CONFIRMADO?:** campo conteniendo el valor NO hasta que el usuario confirme positivamente el dato pasando a valer SÍ.
- **SQL:** en este campo guardaríamos la parte correspondiente a este dato, de la consulta en SQL necesaria para obtener la información que satisfaga el objetivo solicitado. La consulta SQL final se calculará mediante la concatenación de los campos SQL de cada uno de los datos que forman la plantilla.

Los campos descritos no son más que un ejemplo del tipo de información que se podría considerar en la estructura, pero se puede incluir más información según el diseño realizado: diccionario de reconocimiento, modelo de lenguaje utilizado, medida de confianza del dato reconocido...

Veamos cómo se definen cada una de las tres partes que forman el modelo:

- 1) Representación de los turnos de diálogo.** En este caso, los turnos del diálogo se van generando de forma dinámica por el gestor de diálogo. El algoritmo que decide la secuencia de turnos (descrito posteriormente en el punto 3), va generando las estructuras asociadas a cada uno de ellos. Estas estructuras varían



dependiendo del tipo de turno: petición de un dato, confirmación de un dato, acceso a la base de datos, presentación de información al usuario,... El contenido de estas estructuras se rellena con la información almacenada en las estructuras de datos de las plantillas o con información obtenida de la base de datos.

- 2) **Representación de la historia y del estado del diálogo.** En este caso el estado actual del diálogo queda determinado por el contenido de las plantillas: datos obtenidos, datos confirmados,... Además se pueden considerar variables globales adicionales que ayuden a complementar aún más la descripción del estado del diálogo como puede ser el objetivo activo en cada momento, el último dato preguntado o el nivel de destreza del usuario (utilizado en el modelado del usuario, apartado 6.4.5.2).

En cuanto a la historia del diálogo se puede ir almacenando en una estructura dinámica el conjunto de turnos que va generando el gestor en su interacción con el usuario incluyendo cierta información asociada: tipo de turno, dato conseguido o confirmado, pregunta formulada,...

- 3) **Algoritmo para decidir la secuencia de turnos.** En este caso el algoritmo se complica porque debe ir generando dinámicamente los turnos a partir de la interpretación que va haciendo del estado y de la historia del diálogo. En primer lugar el sistema debe detectar el objetivo (parte del servicio) deseado por el usuario. En este caso se puede formular una pregunta en la que se ofrezcan las diferentes alternativas para que el usuario elija una de ellas o se puede realizar una pregunta abierta que permita recoger el objetivo solicitado con alguna información adicional. En la bibliografía podemos ver algunos trabajos relacionados con la clasificación de objetivos (Wright et al, 1997; Popovici et al, 1998). A partir de este punto el gestor de diálogo va generando los turnos según las acciones en el orden de prioridad siguiente (Ward y Pellom, 1999):

- **Clarificación o confirmación del último dato introducido.** Antes de continuar con el diálogo debemos confirmar el último dato reconocido o pedirle al usuario que clarifique en el caso de obtener información ambigua.
- **Mensaje de despedida.** Cuando se ha satisfecho el objetivo solicitado.
- **Consulta a la base de datos y presentación de la información al usuario.** En el caso de que dispongamos de datos suficientes para realizar una consulta (plantilla completa) debemos realizar dicha consulta para obtener la información necesaria, seleccionar dicha información (elegir las mejores opciones de viaje) y presentarla al usuario.
- **Preguntar al usuario cierta información.** Cuando no se dispone de información suficiente para hacer un acceso a la base de datos (no se dispone de una plantilla completa), debemos ir preguntando al usuario los datos que faltan. Para resolver el caso en el que tengamos varios datos incompletos, debemos definir un orden de prioridad entre ellos, de forma que se pregunten primero los de mayor prioridad. Este orden, que debe ser especificado por el

desarrollador, es el que define el flujo de diálogo por defecto que se va ejecutando para el caso en el que la iniciativa sea completamente del sistema.

Aunque ahora la implementación y el ajuste del gestor es más complicado, este modelo tiene la ventaja de que nos permite una mayor flexibilidad. En este caso, podemos implementar con relativa sencillez las diferentes opciones de iniciativa del usuario: cambio en la secuencia de objetivos, y cambio en la secuencia de expresión de los datos.

- El cambio de uno a otro objetivo del diálogo tiene ciertos problemas que debemos considerar antes de introducir esta funcionalidad. En primer lugar esta funcionalidad requiere mantener constantemente activado el módulo para la detección de cambio de objetivo. Este módulo, lejos de ser perfecto, tendrá ciertos errores de detección que debemos valorar. En el caso de que el usuario solicitase un cambio de funcionalidad a la mitad de un objetivo y el sistema no lo detectase, sería un error menor pues podría ser atribuido a una restricción del sistema. Pero si el sistema detecta un cambio sin ser solicitado, la desorientación producida sobre el usuario sería enorme, produciendo un abandono de la llamada casi seguro.

Otro aspecto que debemos tener en cuenta, es la reutilización de los datos. Generalmente los servicios ofrecidos por los SVIs están formados por varios objetivos que comparten gran cantidad de datos. Los dos objetivos utilizados en los ejemplos anteriores comparten los datos; ciudad origen, ciudad destino, fecha y hora del viaje. En el caso de un cambio de objetivo sería muy útil reutilizar los datos ya conseguidos del usuario anteriormente. Por ejemplo, al pasar de solicitar información de horarios a información de precios, podemos no volver a preguntar las ciudades origen y destino y suponer que son las mismas. Este hecho está claro en este ejemplo, pero si el servicio incluyese la posibilidad de hacer la reserva de un coche de alquiler, a lo mejor no sabríamos si proponer la reserva del coche en la ciudad origen o en la ciudad destino del viaje.

- En cuanto a la secuencia de datos, el usuario tendría la libertad de contestar a varios datos en la misma pregunta o contestar a otro dato diferente del preguntado. En este caso el gestor no tiene más que completar la plantilla con el dato obtenido, detectar la información que falta y generar los turnos de diálogo necesarios para obtenerla.

En este punto podemos comentar la posibilidad de considerar datos opcionales en la definición de una plantilla. Por ejemplo, en el caso del objetivo de información de precios podríamos prever que el usuario, por su propia iniciativa, especificase el tipo de tren en el que quiere viajar. El tratamiento de este tipo de datos es diferente al resto. Para estos casos, el gestor no considera necesario disponer de ese dato para definir que una plantilla está completa y se puede realizar el acceso a la base de datos. Por esta razón, nunca se le preguntará ese dato al usuario directamente; si el usuario los especifica por su propia iniciativa, el sistema lo recoge pudiendo hacer búsquedas más ajustadas. En la figura 6-2, no se presentan los datos opcionales, aunque se podrían haber mostrado mediante cajas separadas de las plantillas que forman los objetivos.

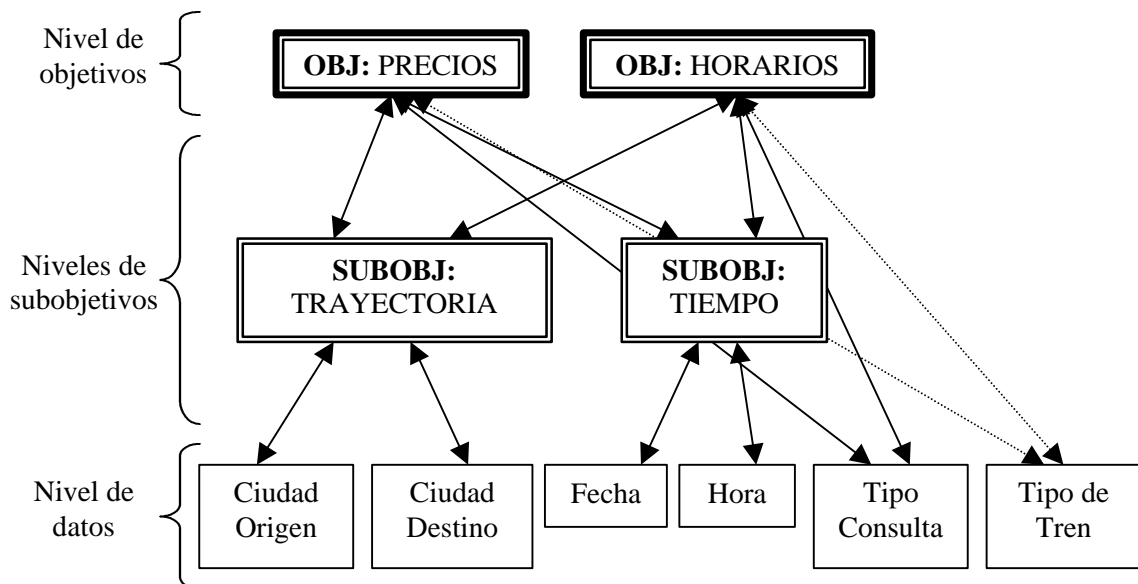
Aunque este modelado es mucho más potente que el descrito en el apartado anterior, no debemos tener la idea de que son técnicas excluyentes. Es habitual, que tomando como base un modelado basado en plantillas, se haga uso de grafos o autómatas finitos para definir partes concretas del diálogo. En determinadas ocasiones, cuando el sistema está teniendo problemas de reconocimiento o cuando un dato resulta difícil de especificar (fecha, número de tarjeta de crédito,...), el gestor puede generar un conjunto de turnos de diálogos organizados en un grafo en lugar de un único turno. De esta forma se define un subdiálogo totalmente guiado por el sistema restringiendo la iniciativa del usuario a favor de una mayor probabilidad de resolución de la tarea.

Algunos ejemplos de este tipo de modelo de diálogo son los siguientes: (Aust et al, 1995; Goddeau et al, 1996; Bennacef et al, 1996; Constantinides et al, 1998).

### 6.2.3 Modelo basado en árboles de objetivos

El modelo basado en árboles de objetivos es similar al de plantillas. Las diferencias residen en la estructura utilizada para representar el servicio ofrecido. En este caso utilizaremos un árbol en el que se muestran los objetivos ofrecidos por el servicio, los subobjetivos en los que se desglosan y los datos utilizados. Veamos el árbol de objetivos para el ejemplo de información de horarios y precios de tren, figura 6-3.

En el nivel superior se representan los objetivos y en el nivel inferior se muestran los datos considerando estructuras análogas a las descritas en el apartado anterior para cada uno de ellos. Entre ambos niveles se ofrece la posibilidad de definir niveles intermedios (subobjetivos) que actúan a modo de bisagra permitiendo una mayor flexibilidad en la definición de las relaciones entre objetivos y datos. Esta estructura nos permite representar la funcionalidad del servicio de una manera más ordenada.



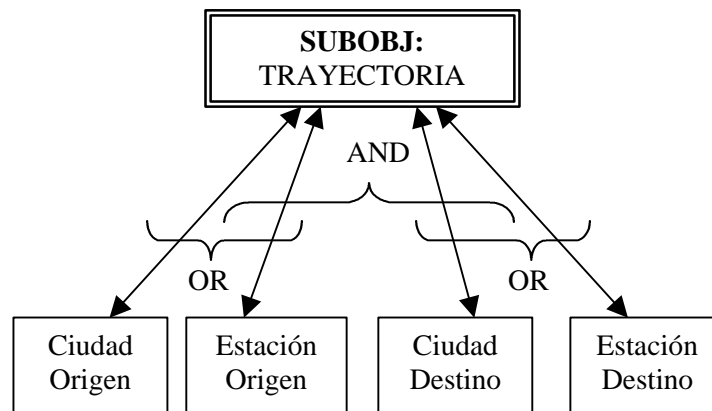
**Figura 6-3:** Arbol de objetivos para el ejemplo de información de horarios y precios de tren.

Las principales características de los subobjetivos son las siguientes:

- Permiten agrupar datos de forma que se pueden definir subdiálogos o pasos del diálogo. Un subobjetivo podría ser el conjunto de turnos necesarios para obtener del usuario los datos correspondientes (preguntas y confirmaciones), los turnos necesarios para negociar la opción de un viaje, o las interacciones necesarias para dar la información de precios. Esta definición de subdiálogos permite definir un ritmo del diálogo con ciertas zonas de descanso en las que se puede hacer un resumen de la interacción o dar algún mensaje de ayuda.
- Otra posibilidad que ofrecen es la de encapsular la complejidad que pueda requerir la obtención de un conjunto de datos determinado. Por ejemplo, a la hora de representar los datos Ciudad Origen y Ciudad Destino, podemos considerarlos como datos independientes de forma que se pregunten al usuario y se confirmen de forma separada. Pero en otras situaciones en las que se disponga de reconocedores más potentes (de habla continua), se podría realizar una única pregunta para solicitar los dos datos simultáneamente. Desde el punto de vista del objetivo la estructura no cambiaría.
- De lo visto en el punto anterior se deduce que los subobjetivos pueden disponer a su vez, de estructuras análogas a las utilizadas para los datos de forma que se pueda almacenar información relacionada con tales subobjetivos: pregunta para pedir todos los datos correspondientes de forma simultánea, frase para confirmar todos los datos, flag de confirmación, flag de subobjetivo completo que se debe activar cuando todos los datos han sido especificados,... Por extensión también podríamos prever estructuras al nivel de objetivo, aunque en este caso predominarían los campos de tipo flag que tengan en cuenta eventos ocurridos: si el objetivo ha sido solicitado, si se han completado todos los datos y se puede acceder a la base de datos, etc.

Una exigencia de diseño importante que debemos realizar en sistemas basados en este modelo es definir reglas de interconexión entre los campos de las estructuras definidas al nivel de datos, subobjetivos y objetivos. Por ejemplo, un subobjetivo lo debemos etiquetar como confirmado cuando se confirmen todos los datos asociados a ese subobjetivo y viceversa, cuando se confirme un subobjetivo (conjunto de datos) de forma conjunta, se deben etiquetar como confirmados todos los datos que existen por debajo. Estas reglas son las que definen la integridad del estado del diálogo representado por el árbol de objetivos. Cada vez que se actualice el estado del diálogo se deben ejecutar todas las reglas de integridad definidas para actualizar todos los campos.

En este modelo también se ofrece la posibilidad de disponer de datos opcionales y la hemos representado con línea discontinua (ver figura 6-3, dato Tipo de Tren). Considerando las reglas de integridad sobre el estado del diálogo que debemos definir, podemos ampliar el concepto de opcionalidad de un dato. Estas reglas pueden definir funciones lógicas complicadas de forma que esta opcionalidad quede a criterio del usuario de la aplicación ofreciendo un punto más de flexibilidad. Veamos el siguiente ejemplo de la figura 6-4.



**Figura 6-4:** Ejemplo de regla de integridad para el subobjetivo TRAYECTORIA.

En este ejemplo podemos ver cómo el subobjetivo TRAYECTORIA quedará especificado cuando el usuario ofrezca un lugar de origen del viaje y un lugar de destino. La especificación del lugar la puede hacer bien mediante la ciudad o bien mediante la estación de tren<sup>1</sup>. En esta situación, la opcionalidad/obligatoriedad de uno u otro dato queda a elección del usuario lo que ofrece una mayor flexibilidad.

Otra ventaja de incorporar el nivel de subobjetivos es que nos permite representar y formalizar de manera más clara la compartición de datos por parte de los objetivos cuando el usuario quiera cambiar de uno a otro. En la figura 6-3 podemos observar como los subobjetivos TRAYECTORIA y TIEMPO son comunes a ambos objetivos (horarios y precios), y por tanto, los datos asociados también.

Por lo demás, en cuanto a los tres puntos característicos de un modelo de diálogo, los comentarios son los mismos a los realizados para el caso del modelo basado en plantillas. Por un lado, la representación de los turnos se va realizando de forma dinámica y el estado del diálogo queda representado por la estructura y contenido del árbol. La secuencia del diálogo se decide en base a un guión por defecto definido por las prioridades de los datos de un subobjetivo y las de los subobjetivos dentro de un objetivo, y donde el usuario puede tomar la iniciativa cambiando la secuencia de datos o el objetivo solicitado.

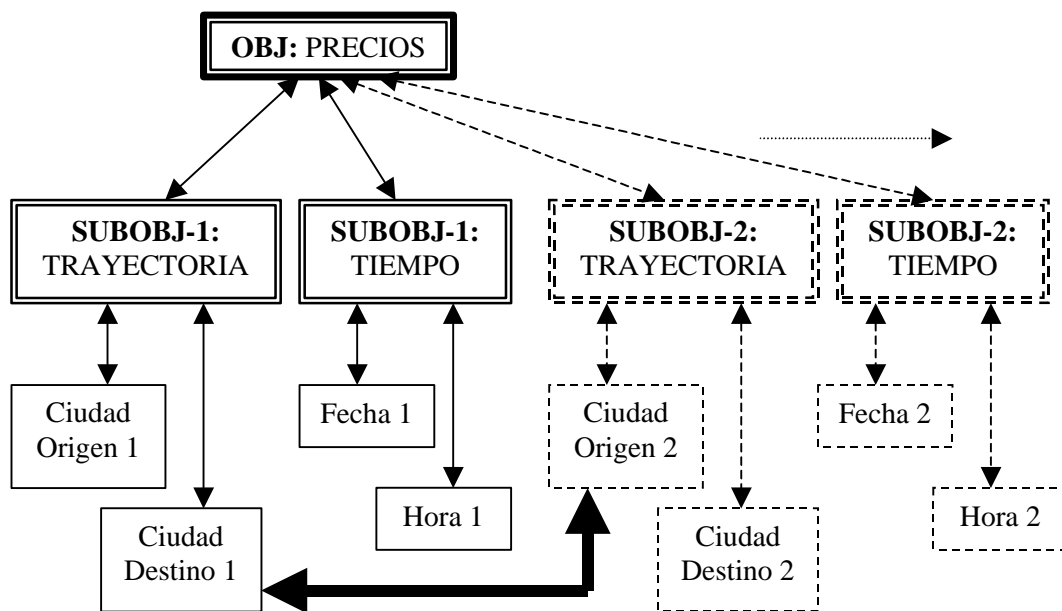
Ejemplos de este modelo de dialogo son los desarrollados en el proyecto Communicator (Ward y Pellom, 1999; Pellom et al, 2000; Rudnicky et al, 2000).

## 6.2.4 Modelo basado en árboles de objetivos dinámicos

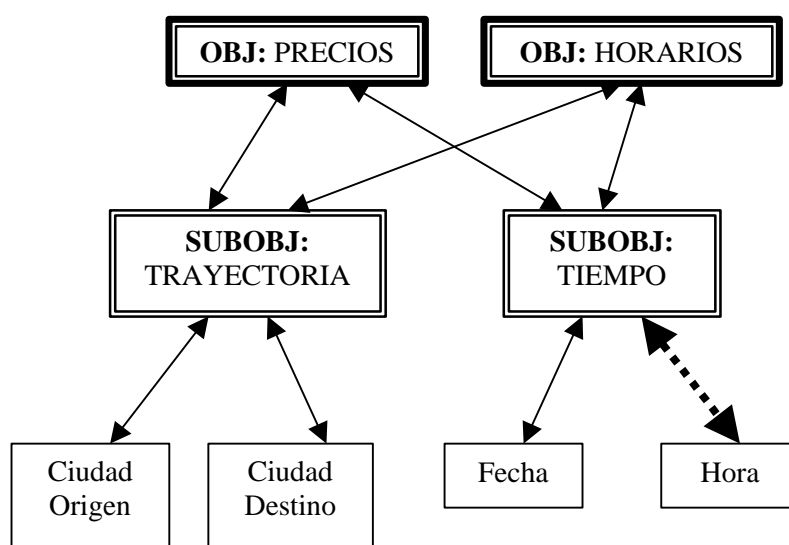
Este modelo es igual al descrito en el apartado anterior con la posibilidad de que la estructura del árbol pueda cambiar de forma dinámica en el transcurso del diálogo. Este dinamismo se puede representar ampliando o reduciendo el árbol de objetivos:

<sup>1</sup> En el caso de que una ciudad tenga varias estaciones de tren, el sistema debe ofrecer la información de viajes de tren para todas las estaciones localizadas en la ciudad especificada.

- Por un lado, a medida que el diálogo avanza, pueden aparecer nuevos datos o subobjetivos, necesarios para completar un objetivo determinado: se pueden definir datos, o subobjetivos enteros, que son inicialmente opcionales para la consecución de una parte del servicio y que pasan a ser obligatorios bajo determinadas circunstancias. En la figura 6-5, podemos observar la posibilidad de que un usuario pueda solicitar información de precios para una ruta en tren formada por varios viajes. De esta forma, el gestor amplía el árbol considerando cada tramo del viaje como una parte adicional del objetivo e imponiendo restricciones entre ellas de forma que la Ciudad Destino de un tramo pasa a ser la Ciudad Origen del siguiente.



**Figura 6-5:** Ejemplo de ampliación del árbol de objetivos según va evolucionando el diálogo.



**Figura 6-6:** Ejemplo de reducción de objetivos según va evolucionando el diálogo.

- Por otro lado, el sistema puede ir haciendo búsquedas parciales en la base de datos de forma que para algunos casos no sea necesario especificar un dato concreto ya que el resultado de la búsqueda ha quedado definido con los datos anteriores: por ejemplo si queremos viajar en tren de Madrid a El Ferrol un lunes, sólo disponemos de un tren directo, y por tanto no hace falta preguntarle al usuario la hora de salida puesto que sólo existe esa posibilidad. Este dato pasaría a ser opcional como se puede ver en la figura 6-6 (línea discontinua).

Algunos sistemas que utilizan este tipo de modelo de diálogo son las nuevas versiones de los gestores desarrollados en el proyecto DARPA Communicator (Ward y Pellom, 1999; Pellom et al, 2000; Rudnický et al, 2000).

En esta tesis no disponemos de reconocimiento de habla continua y comprensión por lo que en el sistema de información y reserva de billetes de tren desarrollado consideraremos una estrategia de diálogo con iniciativa del sistema. Por otro lado, aunque la metodología de diseño que se va a desarrollar en los apartados siguientes se ha demostrado utilizando un modelo basado en autómatas finitos (entorno TADE), se han tenido en cuenta conceptos característicos de los modelos más avanzados: objetivos, subobjetivos o pasos del diálogo, datos opcionales y obligatorios, etc... De esta forma, las conclusiones obtenidas de este estudio serán aplicables al diseño del diálogo en sistemas con iniciativa del sistema y para el diseño del diálogo por defecto (definido mediante las prioridades entre subobjetivos y datos de un modelo) que debe existir en cualquier gestor que incorpore iniciativa del usuario. Este diálogo por defecto requiere de un buen análisis puesto que es la secuencia de turnos que con mayor frecuencia se producirá sobretodo en los primeros usos del sistema, cuando los usuarios tienen poca agilidad.

## 6.3 El problema de la evaluación de flujos de diálogo

Un diálogo sistema–usuario se puede representar como una secuencia de estados (o pasos del diálogo). Estos pasos pueden estar prefijados, o pueden ir generándose dinámicamente como hemos visto. Para cada uno de estos estados se deben tener en cuenta dos parámetros de evaluación fundamentales:

- Tiempo medio que se tarda en ejecutar ese estado.  $T_{M.EST}$ . Es el tiempo consumido en media para que la interacción usuario–sistema ejecute el estado correspondiente.
- Probabilidad de fracaso del diálogo en ese estado.  $P_{ERROR}$ . Es la probabilidad de que el usuario no sea capaz de completar ese estado y acabe colgando.

De esta forma consideraremos que un estado del diálogo es mejor cuanto menor sea el tiempo para ejecutarse y menor la probabilidad de fracaso.

Generalmente el concepto de estado o paso del diálogo se asigna a cada uno de los turnos de diálogo entre el sistema y el usuario. En la figura 6-7 podemos ver cómo se

calculan los parámetros  $T_{M,EST}$  y  $P_{ERROR}$  para el caso de un turno de diálogo en el que el sistema le pide al usuario el nombre de la ciudad origen del viaje.

<b>S:</b>	Diga el nombre de la ciudad origen del viaje	<b>(4,3 segundos)</b>
<b>U:</b>	Sevilla	<b>(2,1 segundos)</b>

**Figura 6-7:** Cálculo de  $T_{M,EST}$  y  $P_{ERROR}$  para un estado del diálogo formado por un sólo turno en el que se pide al usuario la ciudad de origen del viaje.

Para calcular el tiempo del estado debemos sumar el tiempo de la pregunta y el tiempo de respuesta por parte del usuario. En este caso, el tiempo de la pregunta es constante pero el tiempo de respuesta puede variar ligeramente por lo que debemos considerar el tiempo medio. Suponiendo que los 2,1 segundos corresponden con un tiempo medio podemos obtener  $T_{M,EST} = 6,4$  segundos para este estado. Por otro lado, la probabilidad de fracaso depende directamente de la tasa de reconocimiento. Considerando una tasa del 85,3%, podemos decir que la  $P_{ERROR} = 0,147$  (14,7%).

La primera deducción que debemos sacar es que estos dos parámetros se deben calcular realizando pruebas del sistema con usuarios reales, bien para calcular el tiempo medio de respuesta, bien para calcular la tasa real del reconocedor en ese entorno. Si queremos hacer un estudio del diálogo, anterior al período de pruebas, estos valores deben ser estimados. Esta estimación en algunas situaciones puede ser sencilla como en el ejemplo comentado, pero no siempre es así como veremos más adelante. En este caso para calcular un valor aproximado del tiempo de respuesta del usuario podríamos considerar la velocidad de elocución media de los hablantes y la longitud media de los nombres de las ciudades contenidas en el servicio. Por otro lado para calcular la probabilidad de error podríamos estimar la tasa de reconocimiento en condiciones parecidas o con vocabularios de tamaño similar.

A la hora de definir los estados o pasos del diálogo podemos utilizar distintos niveles de granularidad. En el ejemplo anterior hemos considerado un único turno del diálogo pero podríamos considerar varios turnos dentro del mismo estado o paso del diálogo. En este caso la estimación de los parámetros se complica considerablemente. Veamos el siguiente ejemplo (ver figura 6-8) en el que hemos considerado dentro de un mismo estado la pregunta de la ciudad origen del viaje y la confirmación del dato reconocido. Para estimar el  $T_{M,EST}$  y  $P_{ERROR}$  del estado a partir de los datos presentados en la figura 6-8, debemos considerar un número máximo de veces  $N$  que el usuario estaría dispuesto a repetir el dato para que el sistema lo reconociese. En función de este valor, podemos calcular los parámetros del paso completo con las fórmulas siguientes (6-1 y 6-2).

$$T_{M,EST} = \sum_{n=1}^{n=N} \left\{ (1 - P_{REC\ 1})^{n-1} P_{REC\ 1} [n(T_1 + T_2) + (n-1)(T_1 + T_2 + T_A)] \right\} \quad (6-1)$$

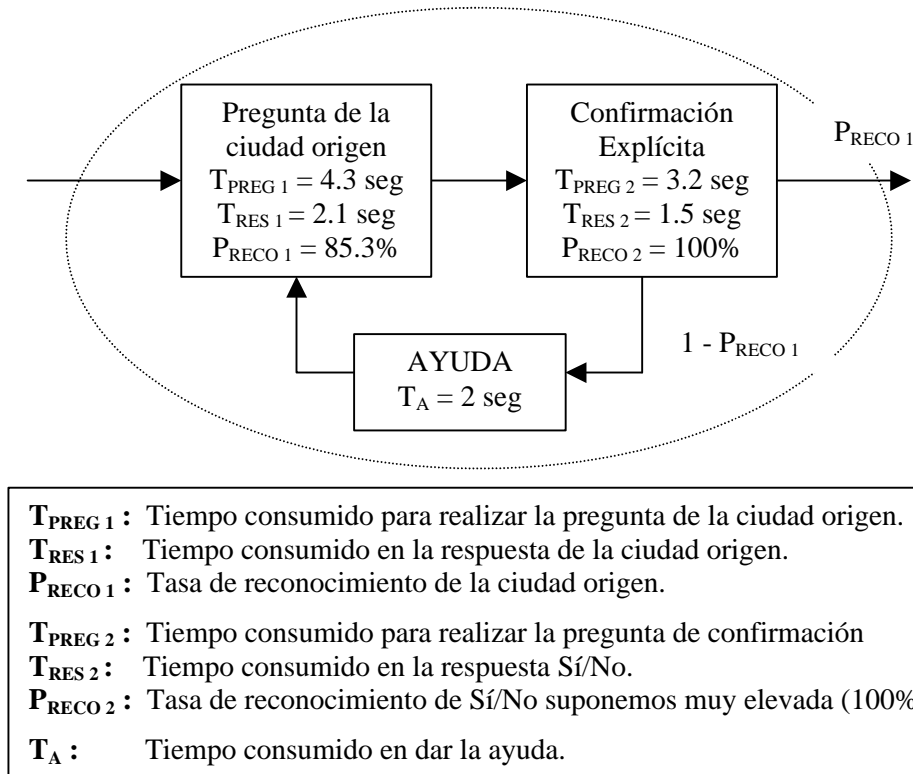
$$P_{ERROR} = 1 - \sum_{n=1}^{n=N} \left\{ (1 - P_{REC\ 1})^{n-1} P_{REC\ 1} \right\} \quad (6-2)$$

donde

$$T_1 = T_{PREG\ 1} + T_{RES\ 1}$$

$$T_2 = T_{PREG\ 2} + T_{RES\ 2}$$





**Figura 6-8:** Cálculo de  $T_{\text{MEST}}$  y  $P_{\text{ERROR}}$  para un estado del diálogo formado por más de un turno de diálogo: pregunta y confirmación del dato.

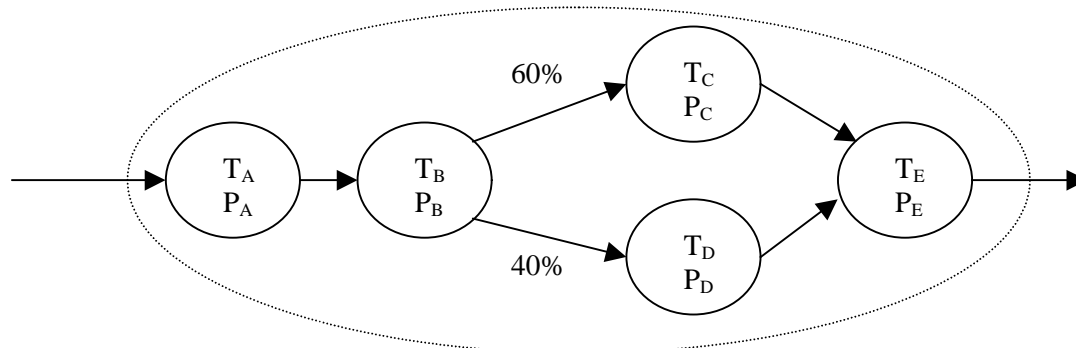
Como podemos ver, a medida que consideramos estados del diálogo más complicados, las fórmulas para la estimación de los parámetros se complican a pesar de haber introducido algunas simplificaciones (en este caso hemos considerado que la tasa de reconocimiento para las palabras Sí/No es del 100%). Además estas fórmulas dependen de ciertos comportamientos de los usuarios: como por ejemplo el valor de  $N$  (número máximo de intentos antes de colgar), que no podemos conocer si no se realizan pruebas con usuarios reales.

Por otro lado, una característica muy importante de los estados es que sus parámetros ( $T_{\text{MEST}}$  y  $P_{\text{ERROR}}$ ) dependen de su posición en la secuencia de estados/pasos que forman el diálogo. Por ejemplo, si consideramos un paso del diálogo en el que el sistema hace un acceso a la base de datos para obtener los viajes de tren y su posterior presentación al usuario. Este paso tendrá diferentes características si el acceso se realiza después de preguntar una hora aproximada de viaje, o si se realiza sin preguntar al usuario ese dato. El volumen de información obtenido en el segundo caso es mucho mayor que en el primero, lo que repercute en que la presentación de las opciones al usuario sea más lenta y por tanto, el tiempo consumido en el estado sea mayor.

De lo anteriormente comentado, podemos concluir que la estimación de los parámetros de evaluación de los estados es una tarea muy complicada, y en algunos casos es inabordable. Por ello, se deben realizar pruebas de campo con usuarios reales

que nos permitan medir los tiempos medios consumidos en cada uno de los estados, así como la tasa de fracaso para cada uno de ellos.

Por otro lado podemos definir pasos del diálogo que a su vez engloben a otro pasos o estados del diálogo más pequeños. En estos casos podemos calcular los parámetros  $T_{M. EST}$  y  $P_{ERROR}$  totales a partir de los parámetros de cada uno de los estados agrupados. Para ello necesitamos conocer la estructura de los estados pequeños que forman el total. Veamos el siguiente ejemplo (figura 6-9).



$$T_{M. EST \text{ TOTAL}} = T_A + T_B + 0.6T_C + 0.4T_D + T_E$$

$$P_{ERROR \text{ TOTAL}} = 1 - [(1 - P_A)(1 - P_B)\{0.6(1 - P_C) + 0.4(1 - P_D)\}(1 - P_E)]$$

**Figura 6-9:** Agrupación de estados para formar un estado mayor. También se muestra el cálculo de  $T_{M. EST}$  y  $P_{ERROR}$  totales a partir de los parámetros de cada estado.

Extendiendo el razonamiento anterior, una vez calculados los parámetros de los estados que forman una parte del diálogo, podríamos obtener el tiempo medio total consumido en ese subdiálogo así como su tasa de fracaso. De igual forma, dando un paso más, el tiempo medio de llamada total así como el porcentaje de consultas no servidas automáticamente se podrían deducir a partir de los parámetros parciales calculados para cada uno de los subdiálogos que forman la aplicación. Estos parámetros nos permiten evaluar diferentes posibilidades de diseño y elegir la mejor. A la hora de seleccionar la mejor opción debemos considerar la que tenga menor tiempo consumido y menor tasa de fracaso. En el caso de tener la misma tasa de fracaso se elegirá la solución con menor tiempo consumido, y en caso de tener igual tiempo, se elegirá la opción con menor tasa de error. El problema surge cuando tenemos opciones en las que los parámetros tienen comportamientos opuestos, es decir, aquella opción que tiene una menor tasa de fracaso supone un mayor tiempo y viceversa. En este caso es necesario combinar ambos parámetros para obtener una única medida de evaluación que nos permita comparar las opciones. Levin y Pieraccini (Levin et al, 2000) proponen la utilización de una combinación lineal con ciertos pesos ponderando cada uno de los parámetros. En este caso, el problema se traslada a cómo definir los pesos de ponderación utilizados. La definición de estos pesos no puede ser única puesto que depende del tipo de servicio utilizado y de la funcionalidad esperada de tal servicio. Supongamos dos casos de diseño en los que tenemos tiempos de llamada de 4 y 8 minutos, con tasas de fracaso del 40% y 20% respectivamente. Para el caso de un servicio en el que hay un apoyo de operadores humanos para atender las llamadas con problemas, se podría preferir un menor tiempo de llamada aun a costa de perder tasa de

resolución mientras que en otro servicio en el que no exista tal apoyo se puede primar la tasa de llamadas satisfechas frente al tiempo consumido. Por tanto, esta decisión la debe tomar el diseñador del servicio teniendo muy en cuenta las necesidades del cliente. Este problema obliga a que las técnicas de evaluación se basen fundamentalmente en el análisis de llamadas con usuarios reales (Suhm y Peterson, 2002)

## 6.4 Metodología de diseño

En los subapartados siguientes describiremos los pasos que constituyen la metodología de diseño de diálogo que se propone en la presente tesis doctoral. En los últimos subapartados describiremos también los ajustes finales realizados sobre el sistema y los resultados obtenidos de la evaluación de campo realizada. Si bien los ajustes son específicos del servicio desarrollado, las ideas mostradas pueden servir de inspiración para el desarrollo de otros servicios. Los detalles del servicio concreto se pueden consultar en el apéndice A.

### 6.4.1 Análisis de la base de datos

Lo primero que debemos dejar claro es que la base de datos a analizar no se refiere a una base de datos de voz ni de diálogos etiquetados. Se refiere a la base de datos que contiene la información que queremos proveer en nuestro servicio. Por ejemplo, en el caso del servicio de información y reserva de billetes de tren, la base de datos contiene la información de todos los posibles viajes en tren a lo largo de la geografía española; tanto la información de horarios y precios como de las reservas de los usuarios. Este análisis consiste en una descripción a alto nivel de la base de datos. La mejor forma de describir una base de datos es representando su diagrama Entidad-Relación (E-R). Este diagrama no tiene por qué representar fielmente el contenido completo de la base de datos. El objetivo de esta representación es mostrar las relaciones existentes entre los datos *que serán útiles para el desarrollo del servicio*. Lo que quiere decir que en la base de datos puede haber información adicional a la que utilizaremos en nuestro servicio que no debiera ser representada en este diagrama. Por ejemplo, la base de datos puede contener información acerca de la distancia kilométrica entre las ciudades, pero como no queremos ofrecer ese tipo de información en nuestro sistema no debíamos incluirlo en nuestro diagrama Entidad-Relación (E-R). Por otro lado, el almacenamiento de los datos puede no corresponder a una base de datos relacional, como es nuestro caso, en el que se utiliza un motor de búsqueda complejo que actúa sobre una lista de ficheros de textos (Ledesma, 2000), por lo que el diagrama debe ocultar también las peculiaridades de la implementación concreta.

En los siguientes apartados veremos un resumen de la descripción formal para un diagrama E-R y su aplicación al ejemplo de desarrollo propuesto.

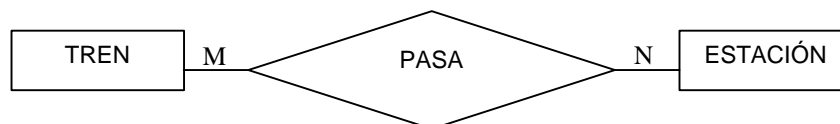
#### 6.4.1.1 Descripción formal

El diagrama E-R está basado en las ideas propuestas por Chen (Chen, 1976; Costilla, 1995). Las principales características de esta representación son:

- Su simplicidad y claridad para representar la información semántica.
- Cada concepto se representa con un conjunto matemático.
- El diseño consiste básicamente en la representación del diagrama E-R.

Veamos resumidamente los principales elementos que componen el diagrama:

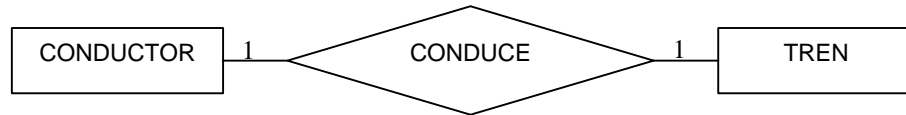
- **Conjuntos Entidad:** conjuntos con todas las entidades similares. Ejemplo: Conjunto TREN que contiene todos los trenes posibles: AVE-127, TALGO-200,... No puede haber unidades repetidas en el conjunto. Estos conjuntos serán los principales conceptos de nuestra aplicación y los candidatos a ser los objetivos para el gestor de diálogo. Cada entidad debe tener un nombre único y se representa con un rectángulo. La selección de los conjuntos entidad es el primer paso en la representación del diagrama.
- **Atributos:** son cada una de las características del conjunto entidad. Por ejemplo para el caso del conjunto TREN podríamos tener: velocidad máxima del tren, capacidad de pasajeros, etc... Cada característica debe tener un nombre único y se representa con una elipse. La representación de estos atributos es el segundo paso en la construcción del diagrama.
- **Clave:** la clave de un conjunto entidad está formada por el conjunto de atributos que definen unívocamente cada elemento del conjunto (ejemplo: código de tren). Se denomina clave compuesta a aquella formada por más de un atributo y clave simple a la formada por un único atributo. Los atributos que forman la clave se representan subrayando su nombre.
- **Conjunto Asociación:** sus elementos son pares de entidades. Este conjunto establece una relación entre dos entidades. Por ejemplo, podríamos establecer un conjunto asociación UTILIZA entre dos entidades como TREN y TRAMO: un TRAMO de viaje UTILIZA un TREN. Estos conjuntos se representan mediante un rombo y pueden tener sus propios atributos y clave. La clave por defecto en estos casos es la unión de las claves de los conjuntos entidad asociados. Veamos con un ejemplo los tipos de conjuntos asociación según su funcionalidad:
  - *Funcionalidad M a N:* en este caso podemos ver como un TREN PASA por varias ESTACIONES y una ESTACIÓN es punto de parada de varios TRENES.



- *Funcionalidad 1 a M:* en el ejemplo vemos como un VIAJE en tren COMIENZA en una ESTACIÓN. Pero esta misma ESTACIÓN puede ser punto de comienzo de varios VIAJES.



- *Funcionalidad 1 a 1*: en este caso un único CONDUCTOR CONDUCE un único TREN y un TREN es conducido por un único CONDUCTOR.

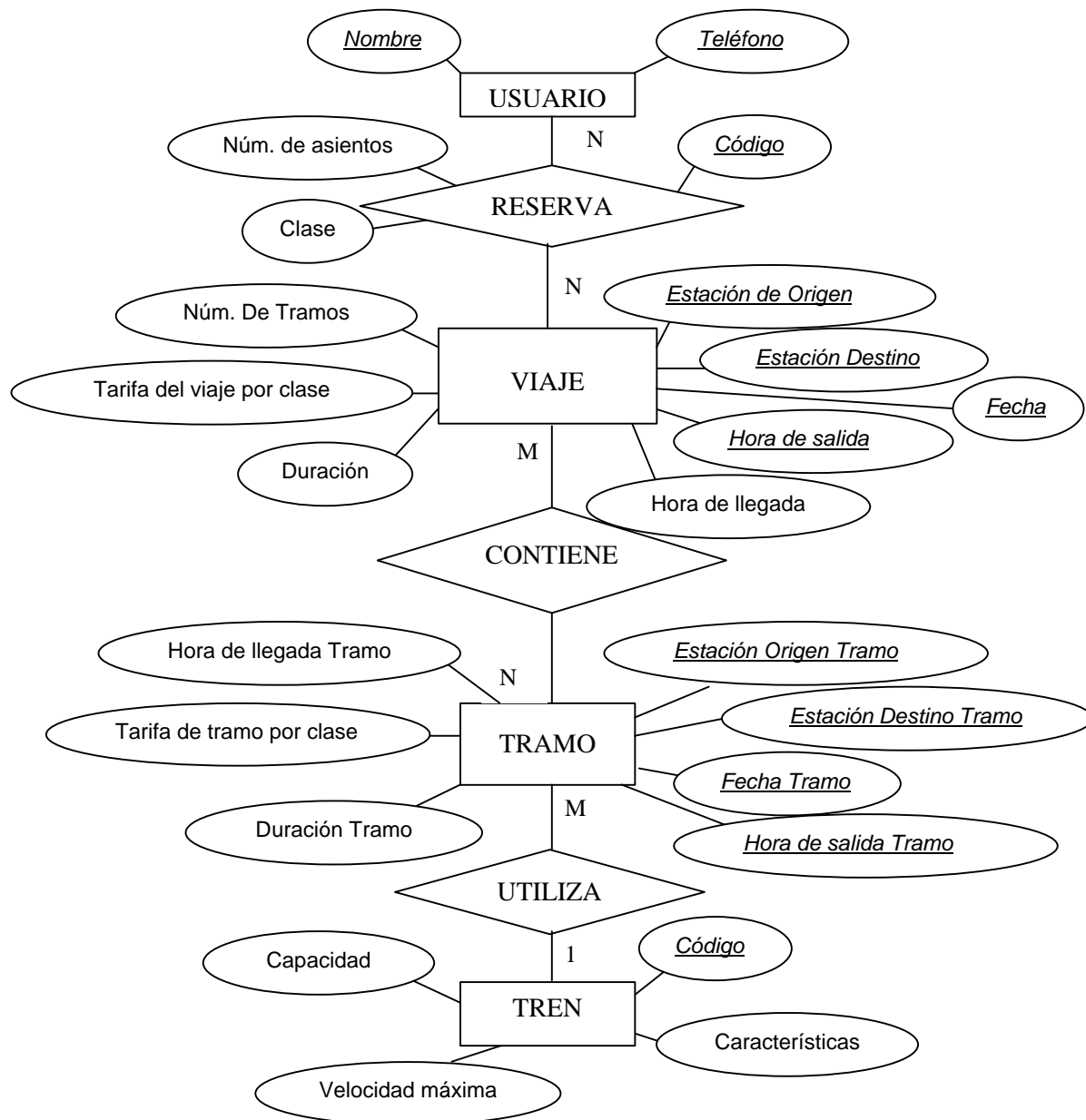


### 6.4.1.2 Aplicación al caso de ejemplo

En este apartado describiremos la construcción del diagrama E-R para nuestro dominio. Para ello utilizaremos la información de horarios facilitada por la empresa RENFE. Esta información ha sido completada con datos simulados sobre las tarifas de precios y sobre las reservas. En la figura 6-10 podemos ver el diagrama final resultante. Los pasos son los siguientes:

- 1- Selección de *Conjuntos Entidad*: considerando la información disponible definiremos los conjuntos VIAJE, TRAMO, TREN y USUARIO.
- 2- Definición de los principales *Atributos* para los Conjuntos Entidad considerados:
  - VIAJE: los atributos que definirán la clave del conjunto VIAJE serán las estaciones de origen y destino del tren, la fecha del viaje y la hora de salida. Otros atributos adicionales son la hora de llegada, la duración total del viaje, la tarifa del viaje por clase y el número de tramos necesarios para completar el viaje.
  - TRAMO: de forma similar al caso anterior, los atributos que forman la clave de este conjunto son la estación de origen y la estación de destino, la fecha del viaje y la hora de salida. Otros atributos adicionales serían la hora de llegada, la duración del tramo y tarifa base del tramo por clase. La tarifa base de un viaje se calcula sumando las tarifas base de cada uno de los tramos que lo componen.
  - TREN: para el caso del concepto tren, la clave está formada por un código de tren que lo identifica unívocamente. Atributos adicionales son la capacidad en número de pasajeros, la máxima velocidad y otras características como los servicios disponibles: teléfono, camas, bar, etc...
  - USUARIO: el usuario lo caracterizaremos por su nombre completo y su número de teléfono, ambos forman la clave para este conjunto entidad.
- 3- Definición de los *Conjuntos Asociación* entre los conceptos, y de los atributos de estos conjuntos asociación. Las asociaciones definidas en nuestro caso son las siguientes:
  - RESERVA: relaciona las entidades USUARIO y VIAJE. La funcionalidad es de N a M, es decir, un usuario puede reservar varios viajes y un mismo viaje puede ser reservado por varios usuarios. Los atributos adicionales que

incorpora esta relación son el código de la reserva, el número de asientos reservados y la clase en la que se quiere reservar.



**Figura 6-10:** Diagrama Entidad-Relación: información y reserva de viajes en tren.

- CONTIENE: relaciona las entidades VIAJE y TRAMO. La funcionalidad es de N a N, es decir, un viaje está formado por varios tramos y un mismo tramo puede formar parte de varios viajes. En este caso no definimos atributos adicionales.
- UTILIZA: relaciona las entidades TRAMO y TREN. La funcionalidad es de 1 a N, es decir, un tramo de un viaje utiliza un único tren mientras que un tren

puede ser utilizado en varios tramos. Para este conjunto asociación, tampoco se definen atributos adicionales.

En resumen podemos decir que los usuarios de nuestro servicio pueden hacer varias reservas de viajes en tren u obtener información de los viajes disponibles. Estos viajes en tren pueden estar formados por varios tramos y a su vez cada tramo está relacionado con el tren que lo cubre.

El diagrama E-R no tiene una solución única y el resultado final depende del desarrollador y del servicio final que queramos diseñar. Por ejemplo en nuestro caso podríamos haber definido un conjunto entidad denominado ESTACIÓN DE TREN pero no lo hicimos porque en nuestro servicio no queríamos proveer información sobre los servicios de las estaciones de tren, luego preferimos considerarlas como atributos tanto de los viajes como de los tramos que los componen.

A la hora de desarrollar este diagrama E-R debemos tener en cuenta los siguientes puntos:

- Hay que prestar especial interés a la definición de los Conjuntos Entidad y Conjuntos Asociación porque la información contenida en ellos será la que defina los posibles objetivos que se pueden ofrecer en nuestro servicio.
- Por otro lado, los atributos que forman la Clave de un Conjunto Entidad o Asociación serán los datos obligatorios que el sistema debe preguntar al usuario para poder ofrecer información de dicho Conjunto Entidad. El resto de atributos serán datos opcionales.

## **6.4.2 Diseño por Intuición**

En este apartado, vamos a describir el segundo paso de la metodología de diseño propuesta, el diseño por intuición. En este paso el objetivo es definir, sobre el diagrama E-R, posibles propuestas sobre el tipo de servicio o conjunto de objetivos que deseamos ofrecer así como posibles flujos de diálogo: tanto la secuencia de objetivos dentro del propio servicio, como las secuencias de preguntas al usuario que nos permiten obtener los datos necesarios para satisfacer dichos objetivos.

### **6.4.2.1 Descripción formal**

Este paso es como un “brain-storming” realizado sobre el diagrama E-R obtenido en la fase anterior. En este punto, es muy importante la experiencia del desarrollador para evitar que la intuición nos pueda llevar por el camino equivocado. El principal objetivo de esta fase es proponer alternativas sobre los objetivos que forman el servicio, los datos obligatorios a preguntar al usuario para satisfacer cada objetivo y la secuencia por defecto para preguntarlos. Estas alternativas serán evaluadas en pasos posteriores de la metodología. Veamos más en detalle el tipo de propuestas a realizar:

- **Objetivos que forman el servicio:** en este caso la idea es proponer varios objetivos que podrían ser resueltos por el sistema a tenor de la información representada en el diagrama E-R. Debemos prestar especial atención sobre los conjuntos entidad y asociación puesto que estos conjuntos representan los conceptos principales de la aplicación. Por esta razón, su contenido es la base del servicio y de los objetivos de diálogo que lo forman. La idea es detectar objetivos que son fáciles de satisfacer y que pudieran cubrir un alto porcentaje de las necesidades de los usuarios.
- **Datos del usuario para cada objetivo:** debemos seleccionar los datos que el sistema necesita para satisfacer un objetivo, es decir, la información que se necesita del usuario para ofrecerle el servicio correctamente. En este punto debemos considerar los atributos que forman la clave de cada uno de los conjuntos entidad o asociación. Estos atributos son los datos obligatorios que se deben especificar para hacer referencia a una unidad dentro del conjunto. Además es necesario plantearse las posibilidades que utilizan los usuarios para especificar cada uno de estos atributos teniendo en mente la tecnología del habla disponible. No es lo mismo plantearse el reconocimiento de una fecha disponiendo de reconocimiento de habla continua, lo que necesitaría de una única pregunta, que utilizar habla aislada que necesitaría de un mayor número de interacciones para obtener el dato.
- **Secuencia por defecto para preguntar los datos:** se deben plantear varias alternativas para la definición del orden en el que se preguntan los datos al usuario.

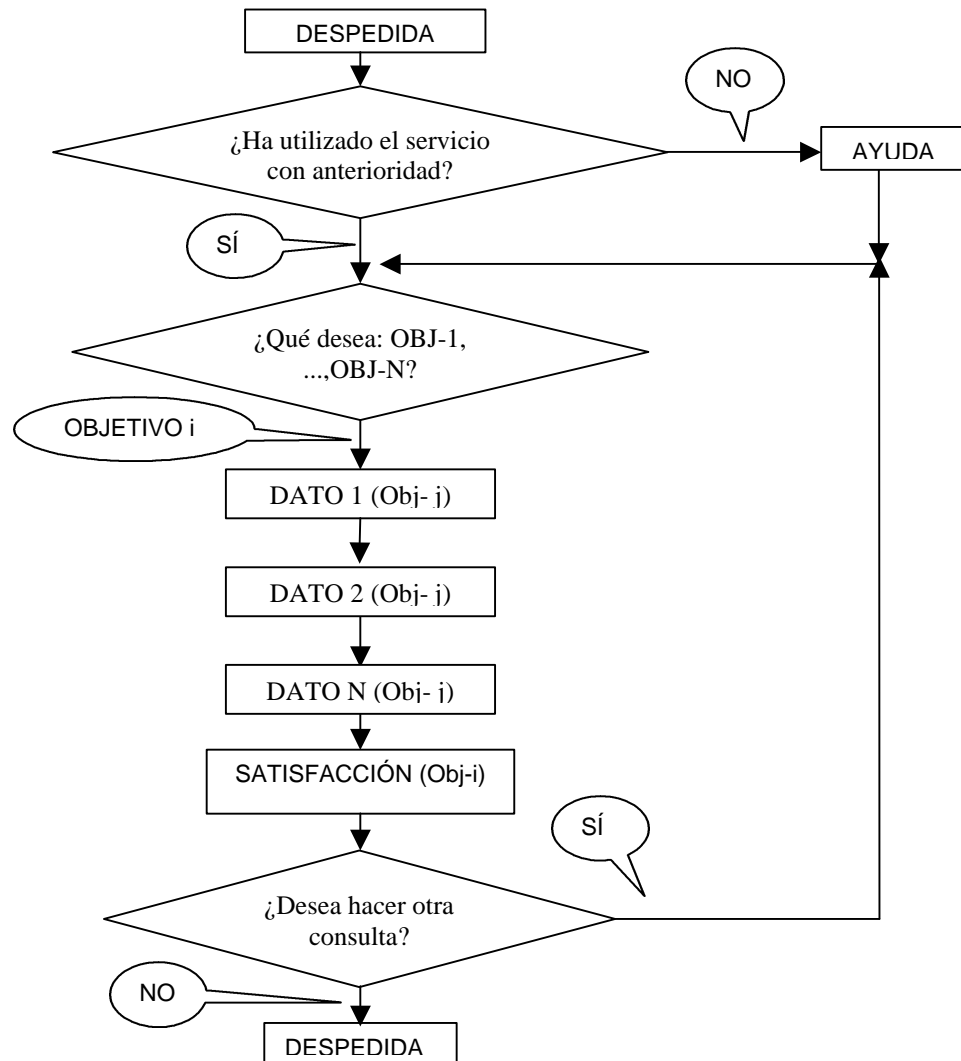
El diálogo que se podría implementar en este punto correspondería con una secuencia intuitiva de interacciones usuario–sistema. En la figura 6-11 representamos una estructura formal del tipo de diálogo que podríamos plantear. Los pasos son los siguientes:

- 1- El sistema ofrece un mensaje de bienvenida lo que permite al usuario identificar que la llamada telefónica ha tenido éxito.
- 2- Posteriormente el sistema ofrece un mensaje de ayuda para los usuarios que utilicen por primera vez el sistema. En este mensaje se ofrecen varias recomendaciones sobre cómo interactuar con el sistema e información del tipo de servicio ofrecido.
- 3- En primer lugar el sistema pregunta al usuario la parte del servicio (objetivo) deseado para pasar posteriormente a preguntarle los datos necesarios para ofrecer tal servicio.
- 4- Una vez conseguidos los datos, el sistema está en condiciones de satisfacer el objetivo solicitado. Después de esto, se ofrece la posibilidad de hacer otra consulta al sistema.
- 5- Para finalizar la llamada, el sistema ofrece un mensaje de despedida.

En cualquier caso, el objetivo de este paso de la metodología no es obtener una primera propuesta de diálogo sino varias. El mejor desarrollo de esta etapa de diseño se



debe evaluar no por la calidad del diálogo que podemos obtener sino por la profundidad del análisis realizado y por el número de alternativas propuestas. Como resultado debemos obtener una hoja o plantilla donde se recojan todas las alternativas. Esta tabla será utilizada en los siguientes pasos de la metodología para anotar la frecuencia de cada alternativa: los objetivos más solicitados, los datos que el usuario facilita al sistema, la secuencia en las que los dice, y la manera en la que los expresa.



**Figura 6-11:** Diagrama de flujo para un diálogo genérico que podríamos implementar después de esta etapa.

Los principales problemas de este análisis son las siguientes:

- La intuición nos puede llevar a tomar decisiones equivocadas especialmente en los casos en los que no se tenga suficiente experiencia.
- Cuando se quiere diseñar un sistema para ofrecer un servicio al gran público, como es el caso que nos ocupa, la intuición de un experto no es suficiente para considerar todos los posibles problemas que pueden aparecer en la interacción usuario–sistema.

Por esta razón no se puede evaluar alternativas sino únicamente proponerlas. Es necesario analizar interacciones reales para poder evaluar las alternativas propuestas, e incluso, proponer nuevas soluciones para problemas no detectados hasta entonces. Este análisis se realizará en los pasos siguientes: diseño por observación y diseño por simulación.

Las principales ventajas que ofrece el diseño por intuición son:

- Las alternativas propuestas son un buen punto de comienzo para conocer más detalles del servicio a implementar, y además, constituyen un primer intento de análisis que nos permitirá afrontar con más garantías los siguientes pasos: permite concretar las alternativas posibles para después centrarnos en su evaluación.
- Mediante la intuición se pueden poner de manifiesto aspectos ergonómicos que debemos tener presente desde el comienzo del diseño.

### 6.4.2.2 Aplicación al caso de ejemplo

Considerando el diagrama Entidad-Relación para nuestro servicio, definido en la figura 6-10, podemos considerar los siguientes objetivos que podríamos ofrecer:

- **Información de viajes de tren:** que el usuario quisiera conocer información de las opciones de viaje disponibles entre dos puntos de la geografía española para una fecha, en un margen de horas determinado: posibles viajes, horarios, duración, tipo de tren utilizado, etc...
- **Información de tramos:** el usuario podría estar interesado en los tramos de los que consta un viaje concreto: para cada uno de los tramos su duración, la estación donde hacer la conexión, la hora de salida del tren, el tiempo entre transbordos, etc...
- **Información de precios:** información sobre el precio de los viajes.
- **Reserva de un viaje:** que el usuario pudiese hacer la reserva del viaje concreto para poder adquirir sus billetes en la estación sin tener que hacer cola de espera. También podríamos ofrecer la posibilidad de compra del billete pero esta funcionalidad tiene ciertos problemas de seguridad al tener que facilitar un número de tarjeta de crédito por teléfono.
- **Información sobre el tren de un viaje:** el usuario podría estar interesado en conocer las características el tren utilizado en el viaje, velocidad máxima, capacidad, disponibilidad de habitáculos para fumadores, etc...

Analicemos ahora los datos necesarios para satisfacer cada uno de los objetivos:

#### 1.- Información de viajes de tren.

En este caso, el sistema necesitará las estaciones origen y destino del viaje, la fecha y la hora de salida:

- Estaciones de Origen y Destino: en este caso, para obtener estos datos nos podemos plantear si es mejor preguntar al usuario por el nombre de la estación o por el nombre de la ciudad donde está localizada la estación. Estas dos propuestas deben ser analizadas y elegir la que más cómoda le resulte al usuario. En el caso de que se elija el nombre de la ciudad, a la hora de ofrecer los viajes habrá que considerar todos los posibles viajes para cualquier estación de tren perteneciente a esa ciudad.

Otro aspecto importante relacionado con estos datos es la secuencia de adquisición. No sabríamos decir si es mejor preguntar por la estación destino y luego por el origen o viceversa. En este caso debemos considerar las dos alternativas como válidas y analizarlas posteriormente.

- Fecha: considerando que en el entorno TADE sólo disponemos de reconocimiento de habla aislada o expresiones cortas, una fecha la debemos considerar como un dato complejo formado por varios datos simples: día, mes y año. Dado que no es habitual prever un viaje con más de un año de antelación, es fácil deducir el año de un viaje a partir del mes del viaje y del año actual. Para obtener el día y el mes podríamos considerar dos interacciones separadas: en la primera preguntaríamos el mes y en la segunda el día del mes. Por otro lado podemos considerar la posibilidad de preguntar al usuario si quiere viajar esta semana y en caso afirmativo, el día de la semana deseado. El concepto semana es una manera natural para el usuario de organizar los días, que nos puede ayudar a diseñar el conjunto de preguntas para obtener una fecha con la tecnología disponible. Estas propuestas deben ser analizadas posteriormente viendo el hábito de los usuarios y su previsión a la hora de organizar un viaje en tren.
- Hora de salida aproximada: al igual que para la fecha, este dato es un dato complejo para la tecnología de reconocimiento utilizada. Este dato está formado por dos datos simples la hora y los minutos. En el caso en el que quisiéramos especificar una franja horaria necesitaríamos dos horas completas (4 datos simples). En este punto debemos considerar el concepto período del día: mañana, tarde y noche, como una manera natural de fraccionar el día. En este caso debemos analizar posteriormente si esta opción le resulta cómoda al usuario y si la limitación horaria que impone nos permite obtener un número de opciones razonables: ni muchas que no se puedan decir todas de golpe, y sea necesario pedir al usuario otro dato, ni pocas que produzca que en la mayoría de las situaciones no se encuentren opciones posibles.

## 2.- Información de tramos.

Los datos requeridos en este caso son los mismos que los comentados en el objetivo anterior, luego son aplicables las mismas ideas. En este caso el usuario debe aceptar alguna de las opciones de viaje propuestas por el sistema, es decir, debe establecer una negociación para seleccionar una opción de las propuestas. En relación con esta negociación debemos tener en cuenta los siguientes aspectos:

- Estrategia de negociación: podemos considerar dos posibilidades, proponer al usuario una de las opciones y permitirle que navegue pidiendo la opción anterior o

posterior, o directamente ofrecerle varias opciones y pedirle que elija una de ellas. En este caso, nos debemos plantear el número de opciones que el usuario puede asimilar y retener.

- Otro aspecto de la negociación es la información que el sistema ofrece para cada una de las opciones de tren. Esta información tiene que ser la suficiente para que el usuario pueda tomar una decisión pero lo más reducida posible para que se pueda asimilar y retener mejor. El análisis debe recoger los criterios de elección de viajes en tren más frecuentes: duración del viaje, hora de salida, tipo de tren,...

### **3.- Información de precios.**

En este caso, también es necesario especificar una opción de viaje concreta con lo que son aplicables los comentarios realizados tanto para la información de las opciones de viaje como de los tramos que las componen. Las propuestas de diseño están relacionadas con la manera en la que debemos satisfacer el objetivo. Generalmente existen muchos precios diferentes para un mismo viaje, dependiendo de la clase, el tipo de usuario (niño, adulto, persona mayor) y el tipo de billete (ida o ida y vuelta). El análisis nos debe revelar si a la hora de dar los precios deberíamos preguntar al usuario la clase, el tipo de usuario o el tipo de billete o no preguntarle esos detalles y ofrecerle la información parametrizada según estas variables (en el apartado 6.4.3.2 veremos el análisis de este aspecto).

### **4.- Reserva de un viaje.**

Para poder hacer una reserva también se requiere seleccionar un viaje de tren concreto, luego lo anteriormente comentado es aplicable en este caso. Además de los datos vistos, el usuario debe elegir la clase en la que quiere hacer la reserva y el número de plazas a reservar. El sistema preguntará al usuario el nombre y el número de teléfono para validar la reserva, ofreciendo un número de reserva que se debe presentar en la estación para retirar los billetes. Si no es posible hacer la reserva, el sistema ofrecerá la posibilidad en otra clase del tren o en otro viaje.

### **5.- Información sobre el tren de un viaje.**

Para poder ofrecer información del tren o trenes utilizados en un viaje, se debe especificar el viaje deseado, luego los comentarios realizados anteriormente son aplicables a este caso de análisis.

Por último, otro aspecto del diálogo en el que tenemos varias alternativas posibles es sobre el punto del diálogo en el que el usuario debe elegir el objetivo deseado. Si analizamos los objetivos descritos arriba y los datos asociados podemos ver que las estaciones origen y destino, la fecha y la hora aproximada de salida son datos comunes a todos ellos. Nos podríamos plantear si es mejor que el usuario seleccionase el objetivo al comienzo del diálogo o que se recogiesen esos datos comunes y luego se le preguntase el tipo de servicio deseado.

El análisis realizado en esta fase nos permite tomar contacto con los problemas que nos vamos a encontrar en el diseño del servicio y da pie a proponer las primeras alternativas de solución que se evaluarán posteriormente.

### 6.4.3 Diseño por Observación

El diseño por observación está basado en el análisis de conversaciones reales entre los usuarios y los operadores humanos en un servicio análogo al que se quiere automatizar. En este análisis, se debe apuntar las veces que aparecen las alternativas propuestas en la fase anterior, con el fin de evaluar el impacto de cada una de ellas. Para ello, podemos utilizar la hoja o tabla obtenida en el diseño por intuición.

Puede ocurrir que aparezcan nuevas opciones no consideradas anteriormente, lo que obligaría a incluirlas en nuestro análisis: podemos detectar que los usuarios están demandando una nueva funcionalidad que inicialmente no estaba prevista en el diseño. Dependiendo de la importancia de estos efectos, podemos replantear otra vez el diseño desde el principio: añadiríamos más información a la base de datos obteniendo un diagrama E-R ampliado y volveríamos a aplicar la intuición para proponer alternativas de diseño. En este caso se debería prestar especial atención sobre la nueva funcionalidad que se quiere incorporar.

#### 6.4.3.1 Descripción formal

Veamos los aspectos que debemos anotar y evaluar en esta fase:

##### 1.- En relación con los objetivos del servicio.

- Debemos detectar los *objetivos que más frecuentemente se solicitan* por parte del usuario y el orden en el que se especifican. Como ya hemos comentado, pueden aparecer nuevos objetivos que deben ser añadidos al análisis. Como resultado de este análisis debemos obtener el número de veces que un objetivo es solicitado y la posición más frecuente, en primer lugar, en segundo lugar, etc...
- Otro aspecto importante es la *información que el operador humano ofrece* al usuario para satisfacer un determinado objetivo: tanto la información concreta como el orden o la manera de ofrecerla. En el caso de que la información sea excesiva debemos anotar bajo qué circunstancias el operador pregunta algún dato nuevo al usuario, o si es él/ella el que fija un valor por defecto.

##### 2.- En relación con los datos a preguntar al usuario.

- Debemos anotar *qué datos son los necesarios para completar un objetivo* y la secuencia en la que el operador los pregunta. Para evaluar este punto debemos anotar las veces que un dato es especificado y el orden en el que lo dijo.
- Clasificación de cada dato como *obligatorio u opcional*.

- Obligatorio: consideraremos un dato como obligatorio cuando el sistema no puede satisfacer el objetivo sin disponer de un valor correcto para ese dato. Puede ocurrir que en los diálogos entre personas algunos de estos datos se omitan. En ese caso se debe anotar si el operador supone algún dato por defecto (y el valor por defecto de dicho dato) o si ofrece la información parametrizada según el valor de ese dato.
- Opcional: cuando no es obligatorio para satisfacer el objetivo, si el usuario lo ofrece, la información será más ajustada a ese valor.

Considerando los atributos que forman o no parte de la clave en el diagrama E-R obtuvimos una primera clasificación de los datos como obligatorios u opcionales. En esta fase debemos validar esa clasificación anotando cuantas veces aparece cada dato como de uno u otro tipo.

- Clasificación de cada dato como *simple o complejo*. Consideraremos un dato como complejo cuando se puede dividir en datos simples. Por ejemplo una fecha se puede dividir en día, mes y año. Dependiendo del tipo de tecnología del habla disponible los datos complejos deben ser divididos en datos simples o pueden ser preguntados con una única pregunta.
- Otro aspecto importante es el análisis de las *diferentes formas* que utiliza un usuario *para especificar el valor de un dato* concreto. Como resultado de este análisis debemos obtener el número o porcentaje de veces que se utiliza una u otra forma de especificación. El objetivo es detectar si merece la pena intentar reconocer todas las posibilidades o hacer frente a las más frecuentes con mayor tasa de reconocimiento.
- *Ordenamiento y agrupación de datos*. En esta fase de observación, debemos analizar la secuencia en la que el operador va preguntando al usuario los datos necesarios. Para ello debemos anotar la posición de cada dato en la secuencia. Por otro lado, también se debe hacer una agrupación de datos en subobjetivos o pasos del diálogo. Esta agrupación se suele realizar por afinidad semántica entre los datos (estaciones de origen y destino) o por proximidad en la secuencia de petición al usuario. Estos pasos nos permiten definir un ritmo del diálogo con zonas de descanso entre ellos en las que se pueden realizar confirmaciones conjuntas de varios datos o se pueden ofrecer mensajes de ayuda al usuario. Algunas reglas para la agrupación de datos son las siguientes:
  - Sólo se pueden agrupar datos que estén seguidos en la secuencia de petición.
  - No debemos agrupar datos que necesiten más de 4 preguntas para obtener sus valores. De esta forma no podemos agrupar más de 4 datos simples o menos si hay alguno complejo.
  - No conviene dejar datos sueltos.

En nuestro caso hemos agrupado la obtención de las ciudades origen y destino del viaje en un mismo paso. La fecha la hemos considerado en otro paso diferente

puesto que al disponer de reconocimiento de habla aislada el número de preguntas para el peor caso es superior a 4.

### **3.- En relación con la negociación entre sistema y usuario.**

Consideraremos que hay negociación cuando el usuario debe elegir o rechazar alguna de las alternativas que el sistema le ofrece, pudiendo cambiar alguna de las restricciones. Los aspectos que debemos analizar en relación con la negociación son los siguientes:

- Los criterios que más frecuentemente se utilizan para elegir una opción de entre un conjunto de opciones: precio, duración, hora de salida,... Estos criterios serán los que nos permitan diseñar el tipo de información que debemos ofrecer para cada una de las opciones. Intentaremos ofrecer siempre la información que ayude mejor al usuario a tomar su decisión. Como resultado del análisis debemos obtener una tabla de frecuencias para cada uno de los criterios utilizados.
- Para los casos en los que se opte por la estrategia de negociación en la que se presentan varias opciones simultáneamente (en lugar de ofrecer una única y permitir navegación), debemos analizar el número de opciones que el usuario puede retener simultáneamente. Este número será el límite de opciones que el sistema puede presentar al mismo tiempo. En el caso en el que aparezcan más opciones deben ser presentadas en turnos de diálogo diferentes.

La principal limitación de esta etapa es que estamos analizando conversaciones entre personas (usuario–operador) y en estos casos el comportamiento del usuario al hablar es diferente que cuando interacciona con un sistema automático (Doran et al, 2001). En esta fase por tanto, se pueden aprender características generales del diálogo como las que se han descrito, pero no detalles de comportamiento de los usuarios frente a un sistema automático. Esta fase tiene otro problema y es que no siempre se dispone de un servicio ofrecido por operadores humanos con anterioridad a su automatización. Puede ocurrir que existan servicios que nazcan con la idea de ser automáticos desde el principio. En este caso, esta fase de observación quedaría eliminada de la metodología y habría que pasar directamente a la fase de simulación en las que se probarán las alternativas propuestas en la fase de intuición.

Por otro lado, esta etapa nos permite estudiar diálogos sin ningún tipo de restricción y podemos medir la frecuencia de los diferentes eventos que se van produciendo. Estas medidas permiten evaluar cada una de las alternativas propuestas en el diseño por intuición sin la necesidad de tener ningún sistema implementado todavía.

#### **6.4.3.2 Aplicación al caso de ejemplo**

Para nuestro caso de ejemplo, hemos transcrito y analizado 100 diálogos entre usuarios y operadores en el servicio de información y reserva de billetes de tren que actualmente ofrece la empresa RENFE. En este apartado presentaremos los análisis más relevantes realizados para el diseño de este servicio. Los formularios/tablas utilizados en

esta fase son los obtenidos en la etapa de intuición aunque las alternativas propuestas se irán ampliando a medida que aparezcan nuevos eventos que se deban registrar.

El primer punto es el análisis de los objetivos, para ello hemos apuntado los objetivos que más frecuentemente se solicitan y su posición en la secuencia del diálogo. En la tabla 6-1, se presentan los resultados para los 100 diálogos analizados. Como se puede observar, en una misma llamada aparecen generalmente más de un objetivo.

<b>Análisis de los OBJETIVOS del servicio</b>						
<b>Objetivos considerados</b>	<b>Porcentaje de llamadas (%)</b>	<b>Posición</b>				
		<b>1º</b>	<b>2º</b>	<b>3º</b>	<b>4º</b>	<b>5º</b>
<b>Información de horarios</b>	<b>64</b>	<b>57</b>	6	1		
<b>Información de precios</b>	<b>46</b>	6	<b>30</b>	10		
<b>Horarios del viaje de vuelta</b>	<b>20</b>		<b>14</b>	5	1	
<b>Reserva del viaje</b>	<b>26</b>	14	4	2	3	3
<b>Frecuencia de los trenes</b>	2	2				
<b>Itinerario</b>	<b>14</b>	8	4	1	1	
<b>Descuentos</b>	5	1	2	1	1	
<b>Envío de los billetes de tren a casa</b>	4			4		
<b>Información para comprar un billete</b>	5		1	3		1
<b>Seguro de transporte</b>	1			1		
<b>Itinerario sobre trenes de Cercanías</b>	9	8	1			
<b>Otros</b>	<b>12</b>	8	2		2	

**Tabla 6-1:** Tabla de análisis sobre los objetivos del servicio de información y reserva de billetes de tren.

A tenor de los resultados, los objetivos más solicitados son los de información de horarios o precios (tanto para el viaje de ida como para el de vuelta), y la reserva del viaje. Por otro lado aparecen nuevos objetivos como la frecuencia de los trenes o información sobre ofertas de viaje. Para nuestro sistema hemos decidido diseñar el servicio de forma que se ofrezcan los objetivos que han sido solicitados en más de un 10% de las llamadas, excepto el objetivo sobre Itinerarios y el denominado Otros. En el caso de información sobre itinerarios porque no se dispone de información con la que completar la base de datos para ofrecer el servicio, y en el segundo caso, se engloban varios objetivos muy dispares que ninguno consigue aparecer en más del 10% de las llamadas. Con este análisis hemos detectado la importancia del viaje de vuelta a la hora de pedir información de horarios, precios o para hacer una reserva. Es un concepto que aparece con mucha frecuencia al hablar de información o reserva de viajes en tren.



Otro aspecto importante obtenido de este análisis es la secuencia en la que el usuario va solicitando los objetivos. En primer lugar se tiende a pedir información sobre los horarios y opciones de viajes posibles. En el caso de solicitar los precios, se realiza habitualmente después de haber obtenido información de las opciones posibles. En cuanto a la reserva se pueden ver dos tipos de comportamiento. El primero de ellos es que en varios casos, este objetivo se completa después de haber obtenido información de las opciones posibles y de sus precios para el viaje de ida y de vuelta. El segundo aspecto es que mucha gente solicita la reserva desde el comienzo mismo de la interacción. Este resultado pone de manifiesto que en muchos casos la reserva de un viaje en tren suele producirse en dos llamadas telefónicas: en la primera de ellas se obtiene información de las opciones de viajes y en la segunda, una vez decidido el viaje a realizar, se completa la reserva.

En el segundo análisis a realizar se deben estudiar los datos que son necesarios preguntar al usuario y el orden en el que se deben preguntar. Para cada uno de los datos debemos evaluar si es un dato obligatorio u opcional, simple o complejo y las diferentes posibilidades que utilizan los usuarios para especificarlo.

Veamos un ejemplo en el que se analiza el dato fecha. La fecha del viaje es un dato obligatorio en cualquiera de los objetivos considerados en nuestra aplicación. Este dato es un dato complejo puesto que está formado por varios datos simples (día, mes y año), y dado que disponemos únicamente de reconocimiento de habla aislada, necesitamos varias interacciones para conseguirlos. El diseño de estas interacciones depende de la manera en la que el usuario especifique habitualmente una fecha. Veamos en la tabla 6-2 el análisis realizado.

	<b>Semana actual</b>	<b>Hoy</b>	<b>Mañana</b>	<b>Día de la semana</b>	<b>Mes actual</b>	<b>Otro mes</b>	<b>Cualquiera</b>
	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>
<b>Fecha</b>	57,8	25,0	15,6	17,2	9,4	4,7	<b>28,1</b>

**Tabla 6-2:** Tabla de análisis sobre las diferentes maneras de especificar una fecha de viaje.

Como podemos observar en el 57,8% de los casos, los usuarios llaman al servicio la misma semana en la que quieren hacer el viaje en tren, un 25,0% en el mismo día, un 16,6% el día anterior y el 17,2% especifica la fecha con un día de la semana determinado (el lunes, el martes, el miércoles,...). Por otro lado, el 9,4% realiza el viaje el mismo mes y especifica la fecha mediante el día del mes (el quince, el veintiuno,...). Algo menos del 5% de los usuarios realizan el viaje en un mes posterior al considerado, y especifica la fecha con el nombre del mes y el día del mes. Generalmente los usuarios no suelen decir el año, ya que sabiendo la fecha actual y el mes en el que se quiere hacer el viaje, es fácil deducir el año, puesto que no se pueden realizar consultas con más de un año de antelación. Como podemos ver, el hábito más frecuente de los usuarios al llamar al sistema es hacerlo la misma semana del viaje. Este hecho hace que a la hora de plantear la secuencia de preguntas para obtener una fecha, se diseñe pensando en favorecer a este tipo de usuarios. Como conclusión de este análisis, hemos diseñado el siguiente esquema para la obtención de una fecha:

- 1- En primer lugar preguntamos si el usuario quiere realizar el viaje la misma semana. En caso afirmativo se le pregunta el día de la semana, si no pasamos al paso número dos. Consideraremos como esta semana, la semana activa desde el día de la llamada. Si la llamada se produce un martes la semana activa cubre hasta el siguiente martes.
- 2- En este punto le preguntamos si desea realizar el viaje el mismo mes. En caso afirmativo le preguntamos el día del mes si no, pasamos el punto tres.
- 3- Por último le preguntamos el mes y el día del mes.

De esta forma para la mayoría de los casos hacemos una pregunta de Sí/No, con una probabilidad muy alta de reconocimiento, y una pregunta del día de la semana, lo que supone también una muy alta tasa de éxito por lo reducido del vocabulario.

Otro aspecto que debemos remarcar es el elevado número de casos (más del 28%) que no especifican ninguna fecha. En este caso, dado que este dato es obligatorio en el cumplimiento de los objetivos del servicio, el operador no tiene más remedio que optar por alguna de las dos siguientes alternativas: fijar un valor por defecto, o informar para varios valores del dato, por ejemplo para todos los días de la semana. En este caso la información parametrizada por la fecha es demasiado abundante con lo que el operador define en la mayoría de los casos fijar un valor que suele ser la fecha de hoy. Estos datos revelan que en algunos casos, los usuarios solamente necesitan información orientativa de los trenes existentes entre dos ciudades, para tener una idea de la duración del trayecto en tren, de la cantidad de trenes disponibles, etc... pero no tienen especial interés en conocer con detalle los trenes existentes para una fecha determinada. El número de casos es lo suficientemente elevado como para que en el diálogo se tenga en cuenta este hecho. En una primera versión, planteamos hacerle una pregunta directa al usuario sobre si tenía la fecha de viaje decidida, para en caso afirmativo pedírsela o en caso negativo ofrecerle información generalizada suponiendo la fecha del día de hoy como valor por defecto. De esta forma se evita que el usuario se vea obligado a especificar una fecha. En este caso no podemos ofrecer información parametrizada por este dato porque la cantidad de información es excesiva. En la fase de diseño por simulación, veremos que esta propuesta tuvo que ser descartada y adoptada otra solución que se describirá más adelante.

De igual forma se han analizado todos los datos necesarios para satisfacer el objetivo. Por ejemplo, en el caso de la estación origen y destino de un viaje se ha puesto de manifiesto que en más del 90% de los casos los usuarios especifican el lugar de origen o destino utilizando un nombre de ciudad y no un nombre de estación. A la hora de diseñar la pregunta, solicitaremos siempre el nombre de la ciudad, si bien, puede haber casos en los que se diga la estación, por lo que mantendremos activos en los vocabularios de reconocimiento, los nombres de las estaciones más importantes (las pertenecientes a grandes núcleos urbanos como Madrid y Barcelona).

En cuanto a la información de precios se planteó la duda de si considerar la clase del tren en la que se desea viajar como un dato obligatorio o por el contrario ofrecer los precios para todas las clases disponibles en el tren. La primera opción supone introducir

una pregunta más en el diálogo para obtener la clase: más riesgo de error y más tiempo consumido. En este caso, dado que el número de clases en un tren no es muy elevado (2 ó 3), el tiempo consumido para ofrecer los precios para cada una de ellas es menor que el tiempo que tardamos en hacer la pregunta de la clase. Además la información ofrecida por el sistema siempre es mayor (sin ser excesiva) lo que puede ser útil al usuario en el caso de que tenga dudas sobre la clase a elegir.

Para el caso del diseño de la negociación entre el usuario y el sistema debemos analizar el criterio más frecuentemente utilizado para elegir la opción de viaje. En la tabla 6-3 se reflejan los resultados de este análisis.

Criterio de negociación								
Hora de salida	Hora de llegada	Duración	Precios	Tipo de tren	Número de tramos	Estación de conexión	Clase	Estación de llegada
41,0%	15,4%	5,0%	7,7%	12,8%	2,6%	2,6%	10,3%	2,6%

**Tabla 6-3:** Análisis de los criterios de elección de las opciones de viaje en tren.

Los criterios más utilizados son la hora de salida, la hora de llegada, el tipo de tren y las clases disponibles en el tren. De este análisis debemos comentar los siguientes aspectos:

- El número de tramos del viaje (o número de transbordos) no parece ser un criterio que se utilice habitualmente cuando habíamos previsto que era un factor muy importante en la comodidad del viaje, y por tanto, el usuario lo debería tener en cuenta. Lo que está ocurriendo es que habitualmente el operador filtra la información que obtiene de forma que sólo presenta al usuario información de viajes directos. Sólo en determinadas situaciones, cuando no hay viajes directos o cuando hay un número muy reducido, el operador ofrece alguna alternativa con algún transbordo y en estos casos sí que tiene importancia. Podríamos decir que este parámetro es tan importante que el sistema debería filtrar esta información antes de presentársela al usuario.
- Otro aspecto sorprendente es que el criterio de precios es menos importante que el tipo de tren (regional, expreso, talgo,...). La razón de este fenómeno es que existe una relación, ampliamente conocida por los usuarios, entre el tipo de tren y el precio del billete. Esta relación no es muy precisa pero suficiente para comparar económicamente las opciones. Además, el tipo de tren ofrece información adicional sobre la comodidad y servicios ofrecidos en el viaje, que también resulta útil en la elección.

## 6.4.4 Diseño por Simulación

El diseño por simulación constituye el cuarto paso de la metodología. En el paso anterior, hemos analizado diálogos entre usuarios y operadores humanos. Dado que este tipo de interacción es diferente a la que ocurre entre un usuario y un sistema automático,

debemos estudiar este tipo de interacciones mediante la herramienta de Mago de Oz (Wizard of Oz: WOZ). Con esta herramienta simulamos el comportamiento del sistema final sin tenerlo completamente implementado, especialmente los módulos de reconocimiento y comprensión. Estas tareas las realiza un operador humano que nunca interactúa directamente con el usuario. El objetivo más importante de esta herramienta es hacer creer al usuario que está interactuando con un sistema completamente automático, sin ninguna intervención humana, para poder aprender su comportamiento. Con esta herramienta nos podemos centrar en el análisis del flujo del diálogo, las preguntas a realizar al usuario, y la información a ofrecer para satisfacer el objetivo. Utilizando esta simulación, no es necesario prestar atención al diseño de los mecanismos de confirmación de datos puesto que no necesitamos realizar tales confirmaciones. Este hecho nos permite independizar el diseño del diálogo, del diseño de las estrategias de confirmación.

Con esta herramienta podemos implementar rápidamente las propuestas sugeridas en los pasos anteriores y evaluar su comportamiento. Además, nos permite experimentar con usuarios reales del servicio original sin que la calidad del servicio se vea afectada de manera importante. Veamos algunos comentarios sobre el diseño de una herramienta de Mago de Oz.

#### **6.4.4.1 Diseño de la herramienta del Mago de Oz (WOZ)**

Como ya hemos comentado anteriormente, la idea que está detrás de esta técnica es sencilla: una persona (llamado Mago) hace el papel del ordenador en una interacción simulada hombre-máquina. De esta forma, podemos analizar la respuesta de los usuarios ante un sistema sin tenerlo aún implementado (lo que nos ayuda de forma importante en su diseño).

Lo más importante de esta técnica es que el usuario, en todo momento, debe pensar que está interactuando con el sistema final y no con una persona. Esta idea es el matiz que diferencia esta técnica del resto. Si no conseguimos convencer al usuario de que está interactuando con el sistema final esta técnica deja de ser útil. Las simulaciones con WoZ deben cumplir los siguientes requisitos:

- Un requisito muy importante y que requiere mucho esfuerzo de desarrollo, es conseguir que las características del sistema automático sean perfectamente simuladas aun a pesar de las limitaciones del mago. Una persona no tarda lo mismo en realizar algunas operaciones aritméticas, o en realizar accesos a una base de datos, que lo que tardaría una computadora. Por esta razón hay que analizar y prever muy bien cuáles van a ser las acciones que va a realizar el mago en su interacción con el usuario, y desarrollar todas las herramientas necesarias para que el tiempo de respuesta sea parecido al de un sistema automático. Por ejemplo, en nuestro caso si queremos que el Mago haga la labor de reconocimiento debemos ofrecerle una lista de candidatos sobre la que le resulte fácil escoger uno de ellos. En nuestro caso se dispone de una primera versión del reconocedor luego lo interesante es ofrecer los candidatos ordenados por similitud acústica con lo pronunciado. De esta forma la corrección del Mago es más fácil e incluso en algunos casos no es necesaria.

- Otro requisito es que antes de simular con el WOZ el sistema, debemos hacer un estudio completo del servicio que queremos ofrecer a través de él. En este estudio debemos dejar muy bien especificadas las limitaciones finales del servicio. Si prevemos que nuestro sistema no va a ser capaz de resolver determinadas tareas, debemos fijar el comportamiento del Mago para que en esos casos, aunque él sí fuese capaz de resolverlas, genere un mensaje de error. Por ejemplo, si el Mago fuese capaz de reconocer que el usuario es una persona joven y que por tanto se podría aplicar algún tipo de descuento por carnet joven, si esta opción no ha sido incluida en el servicio, cuando se ofrezca información de precios se hará de forma general sin tener en cuenta este hecho.

Al plantear la simulación de parte o de la totalidad del sistema, debemos definir dos tipos de aspectos de diseño:

- Aspectos variables o modificables por el Mago de Oz. Son aquellas partes del diseño que se dejan a decisión del Mago. A medida que avanza la interacción con el usuario, el Mago va decidiendo la mejor acción a realizar. Son aquellos aspectos del diseño del diálogo de los que no conocemos muy bien sus características y que queremos, mediante una simulación con WOZ y una serie de experimentos con varios usuarios, obtener información para modelarlo. La determinación de estos puntos es una tarea difícil y a la vez decisiva para garantizar el éxito de esta metodología: las decisiones del Mago deben estar muy controladas y deben hacer referencia a aspectos concretos sobre los que se tiene incertidumbre en su diseño. Estos aspectos corresponden con las propuestas sugeridas y evaluadas en las etapas anteriores de la metodología. Por ejemplo, ante la duda sobre si ofrecer más o menos información de opciones de tren para un viaje concreto, el Mago podría ir controlando este flujo dependiendo de las circunstancias del diálogo. De esta forma podemos ver la cantidad de información que un usuario es capaz de asimilar.
- Aspectos fijados por el diseñador. Son aspectos del sistema que quedan fijados por el diseñador y que no pueden ser modificados por el WOZ en su interacción con el usuario. Pretender que un Mago sea capaz de modificar y definir una gran cantidad de variables hace que esta técnica se haga ineficiente porque el Mago tendrá problemas para tomar muchas decisiones en poco tiempo, y tenderá a realizar lo que menos le cueste sin tener en cuenta la mejor opción. Por esta razón es necesario definir una serie de aspectos de diseño que vendrán fijados de antemano y que por tanto no se podrán evaluar. Estos aspectos deben ser los que tengamos más claro su diseño, y por tanto no necesitamos evaluar varias alternativas.

En cualquier caso puede ocurrir que el número de aspectos a analizar exceda la capacidad de un Mago. En este caso habrá que realizar experimentos con grupos de usuarios diferentes para cada uno de los aspectos a estudiar. En cada uno de los experimentos, se considerarán partes modificables diferentes según la prueba.

A la hora de implementar la técnica de Mago de Oz debemos definir tres elementos importantes: las acciones que puede realizar el Mago (aspectos modificables), las que quedan definidas por el diseñador (aspectos constantes) y un plan de actuación del Mago. El plan de actuación no es más que un conjunto de reglas o pautas que rigen las

actuaciones del Mago. Estas reglas no pretenden fijar el comportamiento del Mago al decidir los aspectos modificables pero sí limitar y orientar sus acciones en ciertas situaciones. Algunos ejemplos de reglas podrían ser: definición de prioridades sobre las decisiones a tomar en el caso de que no le dé tiempo a responder a todos los aspectos, establecimiento de las opciones por defecto en el caso de que no se tome ninguna decisión, emisión de un mensaje del tipo “espere un momento por favor” antes de ponerse a hacer algo que le va a llevar varios segundos, paso a operadora o emisión de un mensaje de error cuando las opciones dadas por el usuario no se contemplan entre las posibles.

En los experimentos con el Mago de Oz se deben apuntar todos los eventos y decisiones que vaya tomando el Mago a lo largo de los diálogos.

El Mago de Oz utilizado en nuestro caso se ha realizado considerando las posibilidades que ofrece el entorno TADE (apéndice A, apartado A.2.2) a través de las cuales un operador puede modificar variables de una aplicación telefónica desarrollada con ese entorno, en tiempo de ejecución. En nuestro caso, la actuación del Mago ha ido orientada principalmente a realizar la tarea de reconocimiento de voz. Se han probado diferentes flujos de diálogo, obviando las confirmaciones de datos, lo que nos ha permitido centrarnos en su análisis. Por otro lado, se ha activado una primera versión del sistema de reconocimiento de forma paralela a la actuación del Mago, lo que nos ha permitido obtener automáticamente las tasas de reconocimiento para cada dato según las alternativas de diálogo probadas.

#### **6.4.4.2 Descripción formal del análisis**

En esta etapa, la estrategia de diseño se basa en probar varias alternativas del diálogo y evaluar el funcionamiento de cada una de ellas. Estas alternativas serán las propuestas en las etapas anteriores. Para que se puedan obtener conclusiones útiles de este análisis, los puntos a probar deben ser aspectos concretos del diálogo: por ejemplo la ordenación de las preguntas de los datos en subdiálogos, el diseño de las preguntas a realizar al usuario, el conjunto de preguntas necesarias para recoger un dato complejo,...

En este apartado veremos de forma general los aspectos que pueden ser evaluados en esta fase. Aunque la presentación detalla todos ellos, no implica que sea necesario realizarlos todos. En determinadas circunstancias, algunas partes pueden haber sido definidas con claridad, bien por las conclusiones obtenidas en la etapa de observación o bien porque nos viene impuesto por las limitaciones del servicio.

##### **1.- Aspectos a analizar**

En cuanto a los objetivos del servicio debemos evaluar si la cobertura del servicio es la adecuada o si los usuarios se esperaban más funcionalidad, y si la organización de los objetivos dentro del diálogo es la esperada.

En relación con los datos que el sistema debe preguntar al usuario debemos analizar:

- La redacción de las preguntas que hará nuestro sistema. Por un lado la finalidad es detectar si los usuarios entienden la pregunta, y por otro lado, ver si las respuestas de los usuarios pueden ser incluidas en el vocabulario del reconocedor (considerando la tecnología disponible) o si por el contrario es necesario hacer preguntas más restrictivas.
- Si los conceptos utilizados por el sistema resultan familiares a la mayoría de los usuarios. Por ejemplo, si es natural especificar un rango de hora aproximado considerando mañana, tarde y noche, o si el concepto ciudad origen y destino se perciben ampliamente como el comienzo y el final del viaje en tren.
- Si la secuencia de preguntas es la apropiada o si los usuarios tienen problemas al contestar a unas preguntas antes que otras.
- Cuando ofrecemos la posibilidad de que el usuario no especifique algún dato que es obligatorio para el sistema, debemos analizar la mejor solución: parametrizar la información por el valor de ese dato, o fijar un valor por defecto y en este último caso el valor concreto a fijar.
- El caso en el que tengamos que plantear varias preguntas para recoger un dato complejo, debemos decidir si la secuencia de preguntas es la más adecuada, es decir, ofrece una mayor tasa de éxito requiriendo un menor tiempo de consulta.

En cuanto a la etapa de negociación se deben analizar las dos posibilidades: presentar una única opción de las posibles y permitir al usuario que solicite la opción anterior/posterior (navegación), o presentar varias alternativas y que el usuario elija una de ellas. En nuestro caso nos hemos decidido por la solución en la que se presentan varias alternativas simultáneamente puesto que es el comportamiento más habitual observado en la etapa de observación. En este caso debemos analizar el número de opciones que un usuario puede manejar al mismo tiempo y la información que debe contener la descripción de cada una de ellas, para que el usuario pueda tomar una decisión.

Debemos prestar especial énfasis en aquellos puntos donde las conclusiones de la etapa de observación no coincidan con las obtenidas en la fase de intuición.

## **2.- Medidas de evaluación.**

La evaluación de las diferentes opciones implementadas consiste en que varias personas llamen al sistema para completar diversas consultas en las que se consideran situaciones o necesidades reales de los usuarios (escenarios). Estos escenarios han sido diseñados de forma que se garantice un número mínimo de ejemplos para cada una de las alternativas que queremos probar en esta fase de análisis. Durante estas llamadas, la herramienta del Mago de Oz debe ir generando un fichero de trazas en el que queden reflejados todos los aspectos del diálogo, las respuestas de los usuarios (grabadas en ficheros de voz), las acciones del sistema y las decisiones del Mago. En una fase posterior, los usuarios deben rellenar un cuestionario en el que se evaluarán los aspectos

subjetivos que no se pueden medir con las anotaciones que realiza la herramienta de Mago de Oz.

En cuanto a la evaluación de la cobertura y estructura de los objetivos que componen el servicio, tenemos las siguientes medidas:

- Las obtenidas por el sistema son: el número de veces que un determinado objetivo es solicitado por el usuario, el tiempo requerido y el número de turnos de diálogo (pregunta–respuesta) necesarios para satisfacerlo.
- En el cuestionario debemos introducir preguntas abiertas en las que el usuario puede proponer nuevos objetivos o funcionalidad del sistema, y preguntas en las que deba evaluar la facilidad con la que se han conseguido cada uno de los objetivos solicitados.

En relación con el diseño de las preguntas consideramos las siguientes medidas:

- Por parte del sistema debemos grabar todas las respuestas de los usuarios en ficheros de voz. Un análisis posterior de estos ficheros nos permitirá definir los vocabularios de reconocimiento para cada una de estas preguntas, o nos obligará a acotar la pregunta para evitar respuestas complicadas de reconocer. También debemos anotar el número de veces que el usuario se queda callado ante una pregunta, lo que revela el bajo nivel de comprensión de la pregunta. Por último si se dispusiese de una primera versión del reconocedor, podríamos evaluar la tasa de reconocimiento y su evolución al cambiar de una pregunta a otra. De esta manera podríamos orientar el diseño de la pregunta para conseguir una mayor tasa de reconocimiento.
- En algunos casos muy dudosos podríamos introducir algunas preguntas en el cuestionario en las que el usuario evaluase la inteligibilidad o aceptación de algunas de las preguntas.

Para analizar la mejor secuencia de preguntas que forman el diálogo, definiremos varias soluciones posibles y elegiremos una u otra en cada llamada de forma aleatoria. Por ejemplo, si no sabemos si preguntar primero la ciudad origen o la de destino podemos plantear varias alternativas de forma que en unas llamadas se elija unas y en otras llamadas otras. Las medidas a considerar son:

- Por parte del sistema debemos grabar las respuestas de los usuarios para su posterior análisis. En el caso de disponer de una primera versión del reconocedor, podemos calcular la tasa de reconocimiento de la secuencia como la multiplicación de las tasas de reconocimiento de cada uno de los datos obtenidos. Este valor nos da la probabilidad de éxito del subdiálogo que forman esas interacciones.
- En el cuestionario se podrían incluir preguntas sobre la preferencia de los usuarios. En muchos casos, el objetivo de incluir en el cuestionario preguntas que se podrían deducir de las anotaciones del sistema es reforzar o contrastar los datos objetivos con la opción subjetiva de los usuarios.



Cuando se dispone de un dato complejo, también pueden ser múltiples los conjuntos de preguntas que se pueden diseñar para obtener del usuario ese dato complejo. La evaluación de las diferentes alternativas se puede realizar con las siguientes medidas:

- Las anotadas por el sistema podrían ser el número medio de interacciones para obtener el dato complejo y el tiempo medio invertido. En el caso de disponer de una primera versión del reconocedor podríamos evaluar la tasa de reconocimiento del dato complejo obtenida como la multiplicación de las tasas de cada una de las preguntas que forman el subdiálogo.
- En el cuestionario podríamos incluir alguna pregunta sobre la naturalidad y comodidad de seleccionar ese dato o que el usuario eligiese la opción que le resultó más cómoda.

Otro aspecto importante referente a los datos, es la acción a realizar cuando el usuario no especifica un dato que es obligatorio para resolver el objetivo. En este caso podemos fijar un valor por defecto o podemos satisfacer el objetivo para varios de los valores del dato. Para evaluar la mejor opción debemos preguntar directamente a los usuarios (en el cuestionario) sobre su preferencia, y en el caso de que prefieran considerar un valor por defecto debemos preguntar sobre el valor por defecto deseado. En algunas situaciones puede ocurrir que no haya alternativas posibles (ej: la tarjeta de crédito en la venta de un billete), y en este caso se le informa al usuario de tal circunstancia, volviéndole a solicitar el valor del dato.

En relación con la negociación hemos optado por la solución de presentar varias alternativas simultáneamente y pedir al usuario que elija una de ellas. Para evaluar este aspecto hemos definido varias posibilidades en cuanto al número de opciones a presentar simultáneamente y el patrón en el que está estructurada la información para cada una de las opciones. Estas posibilidades cambian de forma aleatoria de unas llamadas a otras (pero no dentro de una misma llamada) de forma que los usuarios puedan probar varias de las opciones. Las medidas propuestas son:

- Respecto a las anotaciones del sistema podemos apuntar el número de preguntas necesarias en la negociación y el tiempo consumido.
- En el cuestionario podemos incluir preguntas sobre el tipo de información que mejor ayuda a elegir la opción de un tren y el número de opciones que el usuario puede retener simultáneamente.

Aparte de los aspectos parciales del diálogo podemos realizar una evaluación global del servicio ofrecido por el Mago de Oz. En este caso, mientras en el cuestionario podemos incluir preguntas sobre la facilidad de uso o de aprendizaje del sistema, su rapidez para obtener el objetivo deseado y las partes en las que se encontraron los mayores problemas, el sistema puede anotar el número medio de preguntas necesarias para ofrecer el servicio, el tiempo medio de llamada o las tasas de reconocimiento para cada una de las preguntas.

Un aspecto importante que debemos tener en cuenta es que los usuarios no pueden rellenar un cuestionario demasiado largo puesto que perderían la atención y sus respuestas no serían muy fiables. No debe ser más largo de una hoja. En este cuestionario sólo debemos incluir aquellas dudas que no pueden ser resueltas por otros medios como evaluaciones subjetivas o preferencias de los usuarios. Si vemos que debemos evaluar muchos aspectos a través del cuestionario, es mejor separar las preguntas en varios cuestionarios de forma que cada usuario responda únicamente a uno de ellos. En el apéndice D, podemos ver el cuestionario diseñado para nuestro caso. Siempre que algún aspecto pueda ser evaluado con medidas objetivas, las conclusiones de este tipo de medidas deben ser tenidas en cuenta de forma más relevante que las que se puedan obtener a través de los cuestionarios.

Como primera conclusión del diseño por simulación habría que decir que la herramienta de Mago de Oz es útil para analizar partes concretas de nuestro servicio y que no es viable plantearse un análisis global de un sistema completo con esta técnica. El Mago de Oz permite refinar partes muy concretas de nuestro diálogo. Otra conclusión que podemos deducir es que es necesario un modelado previo del problema a resolver que nos permita definir la intervención del Mago (acciones y plan de actuación) de forma razonable.

El gran problema de la fase de simulación es que hay que invertir mucho tiempo en el desarrollo de la herramienta de Mago de Oz y en las herramientas auxiliares que ayuden al Mago a reducir su tiempo de respuesta. Esta reducción es clave para hacer creer al usuario que está interactuando con un sistema completamente automático y garantizar el éxito de la simulación.

Por otro lado, la principal ventaja de esta técnica es que nos permite probar y analizar varias alternativas de diálogo y varios diseños de preguntas aunque no se disponga del sistema completamente implementado. Otra ventaja importante es que no son necesarias las confirmaciones de los datos puesto que es el Mago el que realiza la labor de reconocimiento y comprensión. De esta forma, permite realizar análisis independientes para el flujo positivo de la aplicación por un lado, y para las estrategias de confirmación y corrección de los datos por otro.

#### **6.4.4.3 Aplicación al caso de ejemplo.**

En esta fase de análisis 14 usuarios llamaron al sistema completando 6 escenarios de viaje diferentes lo que dio lugar a 84 consultas telefónicas. En el apéndice D se puede consultar los escenarios diseñados y el cuestionario que rellenaron los usuarios tras las llamadas.

Uno de los primeros aspectos analizados es el orden en el que se preguntan los datos de la ciudad origen y la ciudad destino. Si bien atendiendo a las respuestas de los usuarios en el cuestionario (apéndice D, pregunta 6), estos prefieren que se pregunte primero la ciudad destino y después la de origen (64,3% frente al orden contrario 35,7%). Pero si analizamos las tasas de reconocimiento obtenidas por una primera versión del reconocedor observamos los siguientes efectos:

- Al preguntar en primer lugar la ciudad destino, obtenemos una tasa de reconocimiento del 94,3%, lo que revela una pregunta muy natural y esperada por el usuario, pero el problema surge cuando se le pregunta la ciudad origen. En este punto, el usuario subestima al sistema y piensa que la ciudad de origen es conocida por el sistema (se puede identificar desde dónde estás llamando), y le resulta extraña la pregunta posterior. Este hecho se traduce en una tasa de reconocimiento muy baja para la ciudad origen (61,2%).
- Por el contrario si se pregunta primero la ciudad origen, el usuario toma constancia desde el principio de que es un sistema automático y de las limitaciones que eso conlleva obteniendo tasas de reconocimiento más homogéneas: 75,6% para la ciudad origen y 79,7% para la ciudad destino.

A la hora de elegir una u otra opción, deberíamos calcular la probabilidad de reconocimiento de la secuencia, obtenida como multiplicación de las tasas de cada uno de los datos. En este caso obtenemos valores de 57,7% y 60,3% respectivamente lo que revela que ambas soluciones son muy similares. En nuestro caso optaremos por la segunda opción (Origen-Destino), puesto que la tasa de reconocimiento de la secuencia es mayor aunque las diferencias entre ambos valores no sean estadísticamente significativas.

El siguiente aspecto importante que debemos analizar es la especificación de la fecha del viaje. Desde las primeras llamadas nos dimos cuenta que la pregunta sobre si el usuario tenía decidida la fecha del viaje, era una pregunta que estaba fuera de lugar y que producía mucha confusión al usuario. Ante esta pregunta los usuarios contestaban respuestas muy variopintas. Una vez decían que sí cuando realmente no sabían la fecha, otras decían la fecha concreta y otras veces preguntaban al sistema sobre el sentido de la pregunta. Este hecho obligó inmediatamente a eliminar esta pregunta. El problema por tanto surgió en cómo detectar los casos en los que el usuario podría no tener la fecha decidida. La solución adoptada fue considerar en nuestro vocabulario de reconocimiento expresiones que un usuario podría decir en esta circunstancia como: “no lo sé”, “todavía no lo he pensado”, etc...

Otro problema importante que hemos detectado en esta fase ha sido la decisión del valor por defecto para la fecha como HOY, en el caso en el que el usuario no la tuviera decidida. Habitualmente los usuarios, cuando no conocen la fecha exacta, no quiere decir que no tengan una idea de cuándo se va a realizar el viaje. Normalmente, aunque no conozcan la fecha sí saben el día de la semana en el que le gustaría hacerlo: si es en fin de semana, en día laboral, si es durante un fin de semana de viernes a domingo... Además las características de los trenes pueden variar bastantes de unos días a otros de la misma semana pero sin embargo los patrones de una semana a otra suelen ser bastantes similares. Esto nos obliga a tener que cambiar el valor por defecto puesto que si un usuario está pensando en hacer un viaje un sábado, la distribución de trenes puede ser muy diferente a la disponible el martes, que es cuando está llamando. Para resolver este problema lo que hemos decidido ha sido lo siguiente: cuando se detecta que el usuario no tiene decidida la fecha (mediante el reconocimiento de alguna de las expresiones comentadas anteriormente), se le da un mensaje de ayuda en el que se le

pide que especifique un día de la semana actual como ejemplo, para que el sistema le pueda ofrecer información más detallada.

El tercer problema con el que nos encontramos en relación con la fecha del viaje fue el concepto de semana activa (de martes a martes). Este concepto no es fácil de asimilar por parte de los usuarios. Este hecho, unido con el fenómeno de que mucha gente puede llamar al sistema un domingo para pedir información de los viajes del lunes o del martes hace que la estructura de preguntas diseñada en la fase de observación no funcione correctamente. En este último caso el usuario diría que no a la pregunta de si quiere viajar esta semana, y el diálogo pasaría a preguntar si es el mes actual y el día del mes deseado. Para evitar que en estos casos hubiera que preguntar el día del mes, se optó por la siguiente estructura de preguntas para obtener la fecha:

- 1- En primer lugar le preguntamos si desea viajar esta semana, la que viene o más adelante.
- 2- En los casos en los que diga esta semana o la que viene se le preguntará el día de la semana. En el caso de “más adelante” se pasa el punto 3.
- 3- En este punto se pregunta al usuario el mes, ofreciéndole únicamente el mes actual y los dos siguientes, y después el día del mes elegido. El hecho de ofrecer tres meses activos nos garantiza una alta tasa de reconocimiento y cubre la mayoría de los casos, puesto que son raros los casos en los que se solicita información con más de tres meses de antelación. De hecho, los usuarios asocian con facilidad esta limitación, en los meses ofrecidos, a una característica del servicio ofrecido y no a una limitación propiamente dicha.

Veamos ahora el análisis de la negociación entre el sistema y el usuario para elegir la opción del viaje. Se plantearon diferentes opciones según el número de alternativas presentadas, y diferentes patrones para la información ofrecida por cada opción de viaje. En cada llamada se selecciona de forma aleatoria un número de opciones y un patrón de información determinado. En la tabla 6-4, vemos los resultados del número de preguntas y tiempo consumido en la negociación, anotados por el sistema para los casos en los que presentemos las opciones de una en una, de dos en dos, o de tres en tres.

De 1 en 1		De 2 en 2		De 3 en 3	
Nº preguntas	Tiempo	Nº preguntas	Tiempo	Nº preguntas	Tiempo
6,1	66 seg.	5,8	97 seg.	4,9	107 seg.

**Tabla 6-4:** Análisis automático de la negociación de las pruebas con el Mago de Oz.

Veamos ahora en la tabla 6-5, el número de opciones preferido por los usuarios y obtenido de los cuestionarios (apéndice D, pregunta 3) y en la tabla 6-6 los criterios de negociación más utilizados.

De 1 de 1	De 2 de 2	De 3 de 3
21,4%	21,4%	57,2%

**Tabla 6-5:** Análisis de la negociación: preferencias de los usuarios sobre el nº de opciones.

Criterio de negociación				
Tipo tren	Hora de salida	Hora de llegada	Precios	Duración
35,7%	85,7%	57,1%	35,7%	14,2%

**Tabla 6-6:** Análisis de los criterios de negociación.

Como podemos ver los usuarios prefieren disponer de tres opciones simultáneamente. En este caso la negociación requiere más tiempo y menor número de preguntas. A tenor de estos resultados hemos decidido mantener la negociación de 3 en 3 opciones reduciendo la información que se ofrece por opción. En un principio se ofrecía el tipo de tren, las horas de salida y llegada, y las estaciones concretas de donde salía y a donde llegaba el tren. A partir de ahora dejamos reducida la información al tipo de tren, y a las horas de salida y de llegada. En este caso no hemos considerado la información de precios puesto que el dato “tipo de tren” ofrece una información orientativa de los precios y de las características del tren.

### 6.4.5 Diseño por Mejora Iterativa

Este último paso de la metodología consiste en un proceso iterativo de prueba y mejora del sistema hasta llegar a una situación o versión estable. En este paso vamos a implementar la primera versión de un sistema completamente automático. Con la herramienta de Mago de Oz habíamos definido una primera versión del flujo del diálogo: las preguntas a realizar para conseguir cada uno de los datos, su secuencia y la información que el sistema debe ofrecer para satisfacer cada uno de los objetivos. Para conseguir automatizar todo el proceso, debemos diseñar las estrategias de confirmación de los datos. Estas estrategias tienen el objetivo de validar los datos obtenidos por el sistema. Con el fin de hacer una gestión eficiente de estos mecanismos de confirmación, es necesario utilizar medidas de confianza de reconocimiento. Dependiendo de la confianza obtenida en reconocimiento debemos adoptar una u otra estrategia de confirmación. También comentaremos más adelante las medidas de evaluación para los mecanismos de confirmación y algunos consejos sobre el diseño de las frases de confirmación a utilizar.

Por otro lado, en esta fase incorporamos también técnicas de modelado de usuario para adaptar el funcionamiento del sistema (preguntas e informaciones que da el sistema) a la destreza del usuario en su interacción con él. Este mecanismo no puede incorporarse con anterioridad a esta fase de diseño puesto que en la fase de simulación el Mago puede comprender todo lo que dice al usuario y adaptarse a él sin dificultad. En un sistema completamente automático, los módulos de reconocimiento y comprensión tienen limitaciones, y por tanto, es necesario prever mecanismos que adapten las preguntas y el flujo del diálogo a la habilidad del usuario, haciendo que sus respuestas entren dentro de las capacidades de estos módulos.

### 6.4.5.1 Diseño de las confirmaciones de los datos

#### 1.- Estrategias de confirmación.

En primer lugar veamos las estrategias de confirmación que podemos utilizar a la hora de diseñar los mecanismos de confirmación en un sistema automático. Según el número de datos a confirmar tenemos:

- *Confirmaciones de un solo dato:* en este caso se confirma el último dato obtenido. Por ejemplo si el último dato fue la ciudad de origen “Madrid”, la confirmación podría ser “¿Desea salir de Madrid?”.
- *Confirmaciones de varios datos:* cuando se confirman varios datos en la misma pregunta. Por ejemplo: ¿Desea viajar de Madrid a Sevilla?

Dependiendo de la confianza del dato obtenido podemos considerar:

- *Confirmación explícita:* el sistema confirma el valor del dato con una pregunta directa al usuario. Por ejemplo: “Entiendo que desea salir de Madrid, ¿Es correcto?”
- *Confirmación implícita:* el sistema no le pregunta al usuario, simplemente le informa sobre el valor reconocido. Por ejemplo, “He entendido que desea salir de Madrid, ¿A dónde quiere ir?”. El sistema informa de lo reconocido y hace una nueva pregunta para obtener el nuevo dato.
- *Confirmación semi-implícita:* es similar a la confirmación implícita pero informamos al usuario de la posibilidad de corregir el valor. Por ejemplo: “He reconocido que desea salir de Madrid. En caso de error diga corregir si no, indique la ciudad de destino”.

Esta confirmación no es muy amigable puesto que incrementa considerablemente el tiempo de la llamada. En el hecho de informar de la posibilidad de corregir en una pregunta, hace que el usuario reciba demasiada información en la misma frase: valor reconocido, comando para corregir y nuevo dato preguntado. La información de las posibilidades que ofrece el sistema para corregir se debe realizar en una ayuda inicial, y a lo largo del diálogo, sólo cuando se detecten situaciones con problemas: el usuario no contesta o el ruido ambiente sea excesivo. De esta forma se evita cargar en exceso las preguntas de los datos.

- *No confirmación:* el sistema tiene tan buena confianza de lo reconocido que no informa sobre su valor, directamente lo da por bueno. Esto ocurre con mucha frecuencia en las preguntas de sí/no utilizadas para las confirmaciones explícitas.
- *Rechazo del valor reconocido y repetición de la pregunta:* cuando el sistema tiene mucha incertidumbre sobre el valor reconocido no intenta confirmarlo con el usuario, directamente lo rechaza y le vuelve a preguntar: “Perdone, no le he entendido muy bien. ¿Desde dónde desea salir?”

## 2.- Medidas de confianza en reconocimiento.

La obtención de medidas de confianza en reconocimiento es un elemento indispensable para el diseño de los mecanismos de confirmación. Estas medidas se pueden considerar de dos tipos:

- **A priori:** dependen del tamaño del diccionario de reconocimiento utilizado en cada caso y de la confusión acústica entre las palabras que componen este diccionario. Estas medidas son poco precisas pero en algunos casos son las únicas disponibles. Dependiendo del tamaño del diccionario de reconocimiento y de la confusión entre las palabras podemos intentar decidir la mejor estrategia de confirmación. Por ejemplo, para preguntas de sí y no podemos optar por no confirmar lo reconocido. En el caso de diccionarios menores de 100 palabras o expresiones, podemos optar por una confirmación implícita y para diccionarios mayores por una confirmación explícita.
- **A posteriori:** son medidas obtenidas del proceso de decodificación. Estas medidas tienen en cuenta el tamaño del diccionario, la confusión de las palabras y la pronunciación particular del usuario en ese momento (posible existencia de ruidos, tos, cambios en la velocidad de elocución,...). Estas medidas son mucho más precisas, y además, permiten diseñar mecanismos de confirmación que se adapten a cada pronunciación en particular. De esta forma para una misma pregunta podemos tener diferente tipo de confirmación según la pronunciación del usuario. Un análisis más en profundidad de este tipo de medidas se puede consultar en el capítulo 5 de la presente tesis.

Los sistemas de reconocimiento utilizados en este servicio son los disponibles en el entorno de desarrollo TADE (ver apéndice A). En este entorno contamos con la versión en tiempo real del reconocedor de palabras deletreadas desarrollado en la presente tesis, y con un reconocedor de gran vocabulario para palabras o expresiones cortas como “el lunes”, “la semana que viene”, etc... (Ferreiros et al, 1998; Macías-Guarasa et al, 1999; Macías-Guarasa et al, 2000a; Córdoba et al, 2001; Macías-Guarasa, 2001).

Para obtener más información de las medidas de confianza del reconocedor de palabras deletreadas podemos consultar el capítulo 5 donde se describen los parámetros y resultados para la detección de errores de reconocimiento y palabras fuera del diccionario. En cuanto a las medidas de confianza para el reconocedor de palabras y expresiones presentaremos brevemente los experimentos realizados y los resultados obtenidos. Estas medidas se han obtenido siguiendo las propuestas para la obtención de medias de confianza en sistemas de reconocimiento basados en una arquitectura de hipótesis y verificación (Macías-Guarasa et al, 2000b; Macías-Guarasa, 2001; San-Segundo et al, 2001b). Los mejores parámetros utilizados están relacionados con la etapa de verificación y son los siguientes:

- **Verosimilitud Acumulada por trama para la mejor Palabra reconocida:** es el logaritmo de la verosimilitud acumulada en el reconocimiento de la mejor palabra o expresión, una vez reordenados las palabras en la etapa de verificación, y dividido por el número de tramas.

- **Diferencia de Verosimilitud para las dos mejores palabras:** diferencia de verosimilitudes para las dos mejores palabras reconocidas en la etapa de verificación, dividida por el número de tramas.
- **Verosimilitud media por trama de las 10 mejores palabras:** es la media del logaritmo de la verosimilitud para las 10 mejores palabras obtenidas de la etapa de verificación, dividida por el número de tramas.
- **Varianza de la Verosimilitud para las 10 palabras candidato:** es la varianza del logaritmo de la verosimilitud obtenida para las 10 palabras, dividida por el número de tramas de voz.
- **Diferencia de Verosimilitudes entre Módulos:** es la diferencia entre los logaritmos de la verosimilitud obtenida para la mejor secuencia de fonemas (en la fase de hipótesis), y la verosimilitud de la mejor palabra reconocida en la fase de verificación, dividido por el número de tramas.

Estos 5 parámetros son análogos a los presentados en el apartado 5.3.2.2 (capítulo 5) para el caso del reconocedor de nombres deletreados. De igual forma que vimos en el capítulo 5, utilizaremos una Red Neuronal para combinar todos los parámetros y obtener una única medida de confianza. En este caso utilizaremos los parámetros como entradas directas a la red. Debemos hacer un reescalado posterior de los parámetros para ajustar su rango de variación al intervalo [0, 1]. La red dispone de 10 neuronas en la capa oculta y una única salida para estimar la confianza de la palabra. Durante la estimación de los pesos, asignamos un valor objetivo de 1 para los aciertos y de 0 para los fallos.

La base de datos utilizada para los experimentos de medidas de confianza se ha obtenido con los resultados de reconocimiento para la evaluación de un diccionario de 1.000 palabras. En este caso, tenemos 2.204 ejemplos, 1450 para entrenamiento de la red, 370 para validación y 370 para evaluación. Hemos repetido los experimentos 6 veces (6-round robin) para verificar los resultados. En este apartado presentaremos los resultados media de todos ellos.

En la tabla 6-7, se presentan los porcentajes de detección correcta de errores para una detección incorrecta del 5% y del 10%. En este caso ofrecemos resultados para la detección de las situaciones en las que el reconocedor falla en el primer candidato (la palabra correcta no es la primera ofrecida por el reconocedor), o cuando falla en los dos primeros (no es ni en el primer ni en el segundo candidato).

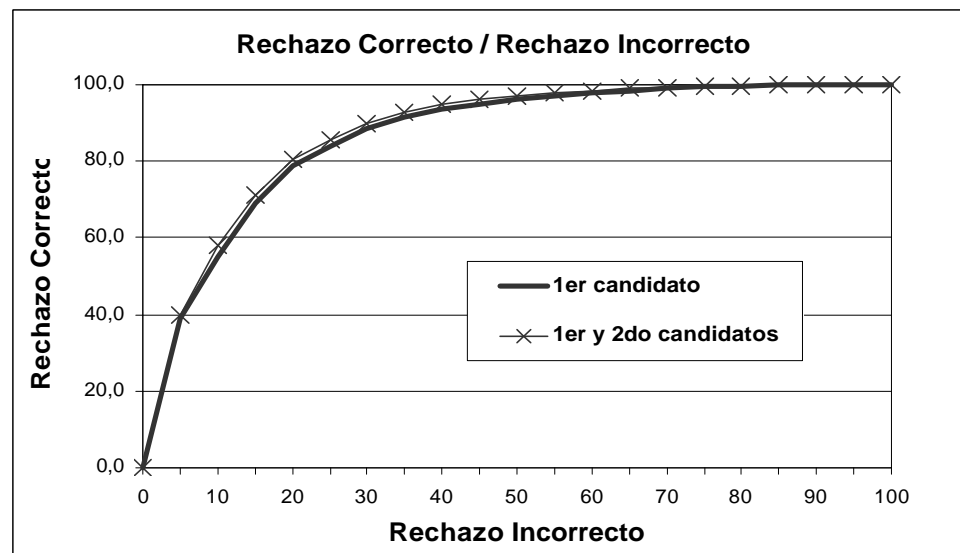
Como podemos ver más del 39% de los errores se pueden rechazar para una tasa de Rechazo Incorrecto (RI, ver capítulo 5 apartado 5.2.4.) del 5%. El error de clasificación se ha visto reducido en 1,8 puntos (un 11,4% relativo) en el primer caso. Los resultados son algo mejores para el caso de rechazar errores considerando el primer y el segundo



candidato. En la figura 6-12 se presenta la evolución del Rechazo Correcto (RC) según el Rechazo Incorrecto (RI)<sup>2</sup>.

<b>Detección de errores en el primer candidato y en los dos primeros candidatos.</b>			
	<b>Rechazo Correcto</b>		<b>Mínimo Error de Clasificación</b>
	<b>5% RI</b>	<b>10% RI</b>	
<b>1º Cand.</b>	39,1%	55,2%	14,0% (Referencia: 15,8%)
<b>1º y 2º Cand.</b>	40,2%	58,3%	9,2% (Referencia: 9,5%)

**Tabla 6-7:** Rechazo Correcto de errores para Rechazos Incorrectos de 5% y 10% para el primer candidato y para los dos primeros candidatos. También se muestra el Mínimo Error de Clasificación y el Error de Referencia.



**Figura 6-12.** Rechazo Correcto vs. Rechazo Incorrecto (RI) para la detección de errores de reconocimiento en el primer candidato y en los dos primeros candidatos.

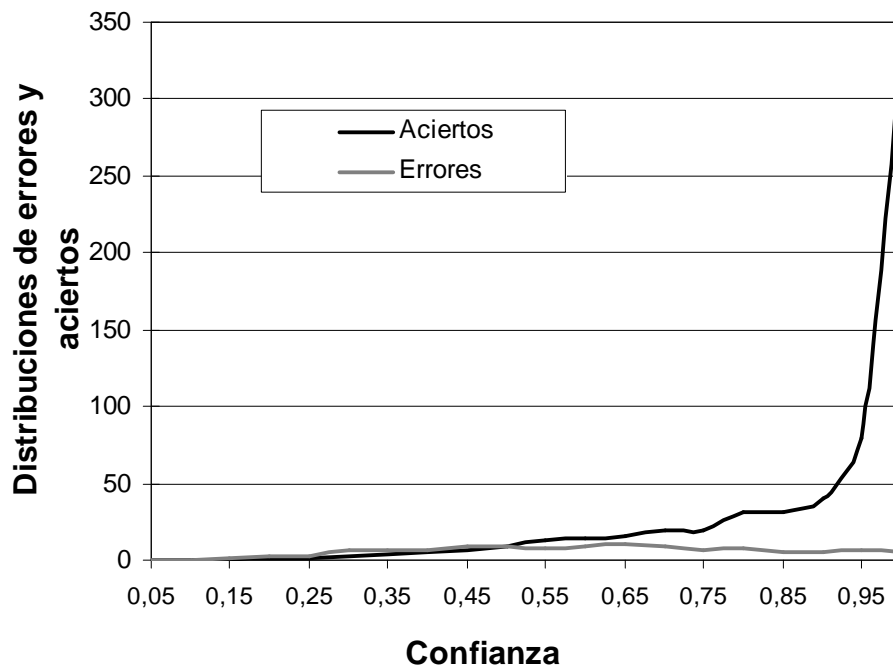
Este análisis de medidas de confianza se presenta en este capítulo y no en el capítulo 5 porque para el caso del reconocedor de palabras y expresiones no hemos realizado un análisis exhaustivo de parámetros. Simplemente hemos aplicado las principales conclusiones de los análisis realizados en el capítulo 5 para reconocedores con una arquitectura basada en dos etapas: hipótesis y verificación.

### 3.- Diseño de los mecanismos de confirmación.

Para utilizar las medidas de confianza en el diseño de los mecanismos de confirmación, debemos representar las distribuciones de aciertos y errores de reconocimiento en función del valor de confianza obtenido. En la figura 6-13 podemos ver estas dos distribuciones para el caso del reconocedor de palabras y expresiones

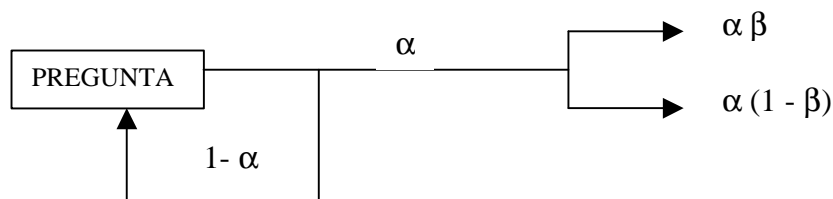
<sup>2</sup> La definición de los conceptos Rechazo Correcto (RC) y Rechazo Incorrecto (RI) se puede consultar en el capítulo 5, apartado 5.2.4.

utilizado, con un vocabulario de reconocimiento de 1.000 palabras. Como se puede apreciar, los aciertos se sitúan en zonas muy cercanas al valor unidad de confianza, mientras que los errores tienen una distribución más uniforme.



**Figura 6-13:** Distribución de aciertos y errores en función del valor de confianza obtenido para el reconocedor de palabras y expresiones utilizado.

A continuación vamos a mostrar dos ejemplos de aplicación de esta gráfica al diseño de las confirmaciones. En el primer ejemplo, consideraremos la situación más sencilla que consiste en rechazar lo reconocido y volver a preguntar cuando la confianza obtenida está por debajo de un determinado umbral que denominaremos  $\theta$ . Cuando se define un determinado umbral de confianza podemos fijar dos nuevos valores:  $\alpha$ , probabilidad de que una palabra reconocida obtenga un valor de confianza superior al umbral, y  $\beta$ , tasa de aciertos para los casos con confianza superior al umbral. El primer valor  $\alpha$  se calcula como la división entre los casos que superan el umbral (tanto aciertos como fallos), dividido por el número de casos totales. El segundo valor  $\beta$ , se obtiene dividiendo los aciertos que superan el umbral por el número de casos totales que superan el umbral (tanto aciertos como fallos). El diagrama de flujo se puede ver en la figura 6-14.



**Figura 6-14:** Diagrama de flujo para el caso de utilizar las medidas de confianza para rechazar las hipótesis de baja confianza y volver a preguntar.

Con las fórmulas siguientes, podemos calcular la probabilidad de éxito y el número medio de preguntas necesarias para la obtención del dato, en función de  $\alpha$  y  $\beta$ .

$$\text{Prob}_{\text{ÉXITO}} = b \sum_{n=0}^{n=N-1} (1-a)^n a \quad (6-3)$$

$$\text{N}^{\circ}\text{Preg}_{\text{MEDIO}} = \frac{ab \sum_{n=0}^{n=N-1} (n+1) (1-a)^n}{\text{Prob}_{\text{ÉXITO}}} \quad (6-4)$$

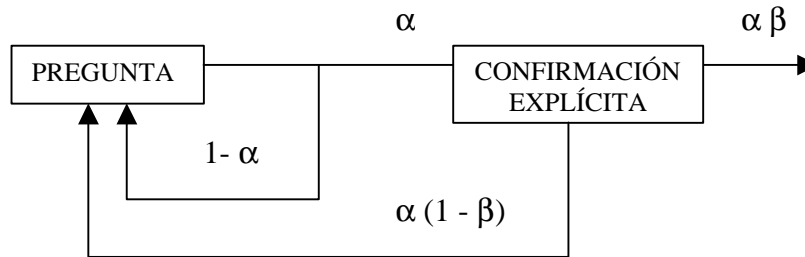
donde N es el número máximo de intentos que un usuario puede aguantar sin colgar. En la tabla 6-8, vamos a representar las probabilidades de éxito y el número de preguntas medio en función del umbral  $\theta$ . Vamos a suponer un número máximo de intentos de 3,  $N=3$ .

Nº medio de preguntas y Prob. de éxito según el umbral de confianza. N=3.				
q	a	b	Nº medio	Prob. Éxito
0,9	0,561	0,974	2,059	0,892
0,8	0,674	0,957	1,781	0,924
<b>0,7</b>	<b>0,762</b>	<b>0,937</b>	<b>1,558</b>	<b>0,924</b>
0,6	0,837	0,910	1,369	0,906
0,5	0,896	0,891	1,227	0,890
0,4	0,942	0,871	1,122	0,870
0,3	0,973	0,857	1,055	0,857
0,2	0,992	0,846	1,016	0,846
0,1	0,999	0,843	1,002	0,843
0	1,000	0,842	1,000	0,842

**Tabla 6-8.** Número medio de preguntas y probabilidad de éxito según el umbral de confianza considerado en la estrategia de rechazo presentada en la figura 6-14.

Como vemos, a medida que el umbral disminuye, el número medio de preguntas se reduce, mientras que la probabilidad de éxito obtiene un máximo para el umbral de 0,7 y 0,8. A la hora de elegir el umbral debemos elegir aquel que, garantizándonos una mayor tasa de éxito, consigue resolver la tarea con un menor número de preguntas. Como ya habíamos comentado en el apartado 6.3, a la hora de elegir la mejor estrategia en cualquier parte del diálogo debemos tener en cuenta estos dos parámetros: probabilidad de éxito y número medio de preguntas (o tiempo necesario para resolver la tarea). La mayor o menor importancia que se le dé a estos parámetros condicionará la solución final elegida.

Veamos a continuación un segundo ejemplo más complicado. En este caso introducimos una pregunta para realizar la confirmación explícita del dato, y en caso de que el usuario no lo confirme, volvemos a preguntar. El diagrama de flujo se representa en la figura 6-15.



**Figura 6-15:** Diagrama de flujo para el caso de introducir una pregunta de confirmación explícita.

En este ejemplo estamos suponiendo que la tasa de acierto del reconocedor de sí y no es del 100% al hacer la confirmación explícita. A pesar de esta simplificación no es posible obtener unas fórmulas sencillas para el cálculo de la probabilidad de éxito y del número medio de preguntas para cualquier  $N$ . Las siguientes fórmulas corresponden a un  $N=3$ .

$$\text{Prob}_{\text{ÉXITO}} = a b \sum_{n=0}^{N-1=2} ((1-a) + (a(1-b)))^n \quad (6-5)$$

$$\text{N}^{\circ} \text{ Preg. MEDIO} = \frac{ab (2 + 3(1-a) + 4a(1-b) + 4(1-a)^2 + 10(1-a)a(1-b) + 6(a(1-b))^2)}{\text{Prob}_{\text{ÉXITO}}} \quad (6-6)$$

En la tabla 6-9, se representan las probabilidades de éxito y el número de preguntas medio en función del umbral  $\theta$ . Vamos a suponer un número máximo de intentos de 3,  $N=3$ .

Lo primero que podemos observar es el incremento del número medio de preguntas debido a la introducción de una nueva pregunta de confirmación explícita. Por otro lado las probabilidades de éxito que se pueden conseguir son muy elevadas. En este caso tenemos un mínimo en el número de preguntas para el umbral 0,4, lo que supone más del 99% de éxito para este valor. Podríamos reducir este umbral para aumentar la tasa de éxito, pero la mejora que tendríamos en tasa no compensa el aumento del número de medio de preguntas. La condición de diseño final vuelve a definirse como un compromiso entre estos dos parámetros: probabilidad de éxito y tiempo consumido.

Niimi presenta en sus trabajos (Niimi y Kobayashi, 1995; Niimi y Kobayashi, 1997) análisis similares pero en su exposición no comentan la posibilidad de que el sistema vuelva a preguntar el dato tras el fallo de una confirmación explícita, lo que dificulta enormemente el cálculo del número medio de preguntas y de la probabilidad de éxito.

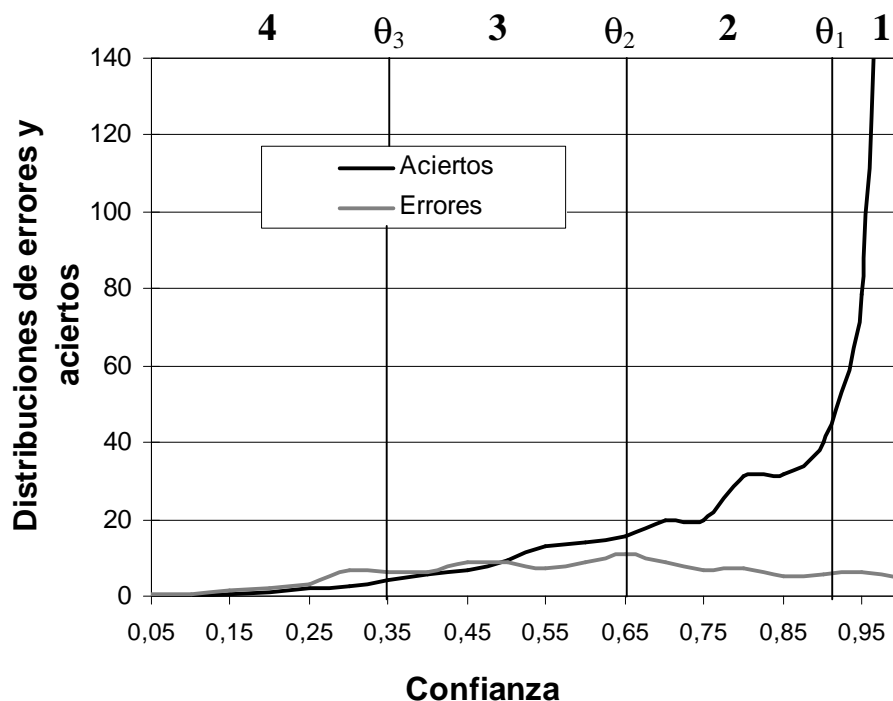
Nº medio de preguntas y Prob. de éxito según el umbral de confianza. N=3.				
q	a	b	Nº medio	Prob. éxito
0,9	0,561	0,974	2,538	0,907
0,8	0,674	0,957	2,443	0,955
0,7	0,762	0,937	2,384	0,977
0,6	0,837	0,910	2,358	0,987
0,5	0,896	0,891	2,338	0,992
<b>0,4</b>	<b>0,942</b>	<b>0,871</b>	<b>2,338</b>	<b>0,994</b>
0,3	0,973	0,857	2,341	0,995
0,2	0,992	0,846	2,349	0,996
0,1	0,999	0,843	2,350	0,996
0	1,000	0,842	2,352	0,996

**Tabla 6-9:** Número medio de preguntas y probabilidad de éxito según el umbral de confianza considerado en la estrategia de rechazo presentada en la figura 6-15.

Cuando nos planteamos casos más complicados en los que se permitan varios tipos de confirmaciones; implícitas, explícitas, rechazo, no confirmación, o cuando se plantea la confirmación de varios datos en la misma pregunta, no es posible hacer un estudio matemático que nos permita obtener una fórmula con la que calcular la probabilidad de éxito o el número de preguntas. Además, se necesitan varios umbrales de confianza para definir la aplicación de un tipo de confirmación u otro. En estos casos, hay que recurrir a diseños con ajustes manuales: tenemos que definir manualmente diferentes umbrales sobre la representación de las distribuciones de aciertos y errores según el nivel de confianza. Veamos un ejemplo en el que vamos a diseñar los mecanismos de confirmación para los datos ciudad origen y ciudad destino de un viaje en tren. En la figura 6-16, podemos ver el detalle de la parte de inferior de las distribuciones de aciertos y errores con los umbrales que hemos definido.

En la esta gráfica, hemos definido 4 niveles de confianza (3 umbrales). Son los siguientes:

- **Confianza MUY ALTA:** en este caso el número de aciertos es mucho mayor que el número de errores. En este nivel el sistema está bastante seguro sobre la validez del dato reconocido. Para este caso podemos proponer una estrategia de *confirmación implícita* de uno o varios datos a la vez.
- **Confianza ALTA:** en este nivel el número de aciertos es mayor que el de errores. A este nivel proponemos una estrategia de *confirmación explícita* pero confirmando *varios datos* a la vez.



**Figura 6-16:** Distribución de aciertos y errores con los umbrales definidos manualmente para el diseño de los mecanismos de confirmación.

- **Confianza BAJA:** la zona definida entre los umbrales  $\theta_2$  y  $\theta_3$  tiene distribuciones de aciertos y de errores muy parecidas. El sistema no puede estar seguro sobre la validez del dato reconocido debiendo hacer una *confirmación explícita de cada uno de los datos*.
- **Confianza MUY BAJA:** en la última zona, el número de aciertos es inferior al de errores lo que pone de manifiesto que lo más probable es que se haya reconocido con error. En este caso la propuesta es *rechazar el valor ofrecido* por el reconocedor y formular la pregunta otra vez.

Consideremos el caso de la confirmación de las ciudades origen y destino de un viaje. Definiremos  $N(O)$  como el nivel de confianza de la ciudad Origen y  $N(D)$  el nivel de confianza de la ciudad Destino. Dependiendo de los niveles obtenidos para cada uno de los dos datos, hemos definido las siguientes estrategias de confirmación:

- $N(O) = 1$  y  $N(D) = 1$ : en este caso utilizaremos una **confirmación implícita de los dos datos**. Ej: “Entiendo que desea viajar de Madrid a Sevilla. ¿Cuándo desea salir?”
- $N(O) = 2$  ó  $N(D) = 2$ : cuando ambos datos obtuvieron un nivel de 2 realizaremos una **confirmación explícita de ambos datos**. Ej: “¿Desea viajar de Madrid a Barcelona?”

- c)  $N(O) = 3$  ó  $N(D) = 3$ : si alguno de los datos obtiene una confianza baja debemos realizar una **confirmación explícita de ese dato**. Ej: “¿Ha dicho Madrid?”
- d)  $N(O) = 4$  ó  $N(D) = 4$ : cuando se obtiene muy baja confianza **rechazamos el dato** reconocido y volvemos a preguntar. Ej: “Lo siento no he entendido muy bien, ¿Desde dónde quiere salir?”

Generalmente, cuando el usuario no confirma un dato, el sistema le vuelve a preguntar. En ciertos casos en los que el vocabulario de reconocimiento sea grande (como para el caso de la ciudad origen o la ciudad destino), hemos incorporado la posibilidad de que se pregunte por el segundo candidato de reconocimiento. En este caso, es muy útil el análisis de medidas de confianza para los dos primeros candidatos que hemos visto en la figura 6-12. Sobre este análisis es necesario definir un único umbral que me permita decidir si merece la pena o no ofrecer el segundo candidato de reconocimiento o pasar directamente a preguntar el dato otra vez.

En condiciones de gran ruido ambiente o problemas de reconocimiento puede ser necesario pedir al usuario que deletree el nombre de la ciudad pronunciado. En este caso debemos utilizar el análisis de medidas de confianza para este reconocedor tanto para la detección de errores como de palabras no pertenecientes al vocabulario de reconocimiento (ver capítulo 5, apartado 5.3).

Un detalle importante con el que nos hemos enfrentamos en la implementación de este servicio, es la variación que sufren las medidas de confianza cuando se introducen varias expresiones diferentes para un mismo valor de un dato. Por ejemplo cuando preguntamos por el día del viaje permitimos que los usuarios digan expresiones del tipo “martes” o “el martes”. En este caso, al tener ambas pronunciaciones incluidas en el diccionario de reconocimiento, y dado que las medidas de confianza se basan en la diferencia entre las verosimilitudes de primer y segundo candidato (segundo parámetro de confianza visto anteriormente), la confianza obtenida en este caso es muy baja, aunque la certeza sea muy alta. La solución adoptada ha sido la de saltar las expresiones que hacen referencia al mismo dato (“martes”, “el martes”), y considerar los parámetros acústicos para expresiones que hagan referencia a valores diferentes.

#### 4.- Diseño de las frases de confirmación.

En las confirmaciones implícitas, las frases a utilizar están formadas principalmente por dos partes: en la primera parte, el sistema informa sobre el valor reconocido en la interacción anterior, y en una segunda parte, el sistema pregunta al usuario el nuevo dato a obtener, o realiza una consulta a la base de datos para acceder a la información solicitada. Por ejemplo: “Entiendo que desea salir de Madrid [1ª parte], ¿A dónde quiere ir? [2ª parte]”. Esta estructura nos permite realizar análisis independientes para las frases de confirmación del dato anterior, y para la frase con la que solicita un nuevo dato al usuario. Sobre el ejemplo anterior podríamos modificar la frase con la que se pregunta el dato (2ª parte) dejando intacta la frase de confirmación del dato anterior. Por ejemplo: “Entiendo que desea salir de Madrid [1ª parte], diga la ciudad destino de su viaje [2ª parte].”

En muchos casos existe una relación entre el dato o acción reconocida y el nuevo dato o la nueva acción a realizar. En estos casos se puede unir ambas frases en una sola. Veamos el siguiente ejemplo de diálogo:

*Sistema:* ¿En qué mes desea viajar?

*Usuario:* en julio

*Sistema:* Diga el **día de julio**.

*Usuario:* el veinticuatro

...

En este ejemplo, vemos cómo hay una relación entre el mes, que es el dato a confirmar, y el día del mes que es lo siguiente a preguntar. En este caso la unión de ambas preguntas aporta una mayor agilidad y naturalidad al diálogo. Por otro lado, esta unión hace que no podamos hacer análisis independientes para cada una de las partes, o que no podamos modificar una parte dejando la otra constante.

En el caso de confirmaciones explícitas, la locución del sistema está formada por una única frase en la que el sistema pregunta explícitamente (mediante una pregunta de sí/no) si el dato es correcto. Por ejemplo: ¿Desea salir el 18 de julio?. Esta frase podría dividirse en dos y adoptar una estructura similar a la de las confirmaciones implícitas, en una primer parte se presenta el dato reconocido y en una segunda parte se pregunta su validez. Por ejemplo: “Entiendo que desea salir el 18 de julio [1ª parte]. ¿Es correcto? [2ª parte]”. En el caso de adoptar este tipo de estructura, sería muy fácil hacer convivir ambos tipos de confirmación: implícita y explícita. La primera parte podría ser común a los dos tipos de confirmación y sólo modificaríamos la segunda parte según la confianza del dato reconocido. En el caso de confirmación explícita haríamos una pregunta directa y en el caso de confirmación implícita preguntaríamos el siguiente dato. La separación en dos partes de las frases de confirmación, tiene el grave inconveniente de que aumenta considerablemente el tiempo de la llamada, haciendo la interacción más lenta y pesada al usuario. Esta propuesta es válida cuando se está diseñando un diálogo muy guiado por el sistema: por problemas en el reconocimiento o porque el usuario es inexperto en el uso del sistema. En otras situaciones conviene tender a reducir la locución a pronunciar para agilizar y dar viveza a la interacción.

Para concluir este apartado veamos algunas recomendaciones generales para el diseño de los mecanismos de confirmación:

- La confirmación implícita debe utilizarse tanto como sea posible porque agiliza enormemente la interacción. El hecho de incorporar este tipo de confirmación obliga a prever mecanismos de corrección que permitan al usuario modificar el valor de un dato incorrecto. En los apartados siguientes se propondrán diferentes mecanismos de corrección.
- En el diseño de las frases para realizar las confirmaciones implícitas, siempre que sea posible, hay que utilizar una única frase que al mismo tiempo pregunte el nuevo dato y confirme el dato anterior. De esta manera se permite mayor viveza y agilidad en el diálogo.



- A la hora de diseñar la frase de confirmación, se debe tener en cuenta la manera en la que el usuario ha especificado el dato. Si por ejemplo el usuario utiliza la frase “Quiero viajar el lunes que viene” para especificar la fecha del viaje, la confirmación debería ser: “Entiendo que desea viajar el lunes que viene,...” en lugar de utilizar una locución como: ”Entiendo que desea viajar el 18 de julio,...”

### **5.- Medidas de evaluación de las estrategias de confirmación.**

Dado que podemos elegir entre varias estrategias de confirmación, es necesario definir medidas que nos permitan evaluar el funcionamiento de una estrategia en una parte concreta del diálogo.

Para evaluar el tipo de confirmación implícita debemos analizar los siguientes parámetros:

- La tasa de reconocimiento (transcribiendo las respuestas de los usuarios), que nos permite obtener los casos en los que fue necesario corregir.
- Porcentaje de errores que fueron corregidos con éxito.
- Tiempo medio de corrección: tiempo medio necesario para que el usuario modifique el dato sugerido por el sistema.

A la hora de plantearnos la utilización de esta estrategia debemos considerar un compromiso entre estos tres parámetros. De esta forma, si el tiempo medio de corrección no es muy elevado y la mayoría de los casos son corregidos con éxito, puede ser razonable utilizar esta estrategia aun no teniendo una tasa de reconocimiento muy elevada.

En cuanto a la confirmación explícita debemos evaluar también el error de reconocimiento, o porcentaje de veces que el usuario rechaza la opción. Si este número es reducido, podemos plantearnos pasar a utilizar una confirmación implícita.

Para los casos de rechazo del dato y vuelta a preguntar, debemos considerar la tasa de rechazo incorrecto (datos correctos que fueron rechazados por error), no pudiendo superar el 5% aproximadamente. En el caso de no confirmación, debemos evaluar la tasa de reconocimiento para cerciorarnos si es lo suficientemente elevada como para no confirmar lo reconocido.

A nivel general podemos evaluar los mecanismos de confirmación con las siguientes medidas:

- Número medio de preguntas necesarias y tiempo medio empleado para confirmar los valores de los datos.
- Número de veces que el usuario rechaza el valor propuesto (tasa de reconocimiento).
- Porcentaje de los casos de error que han sido corregidos con éxito.

- Para analizar el impacto de las medidas de confianza en la elección de una u otra estrategia y por tanto en la rapidez de la interacción, podemos calcular el porcentaje de confirmaciones implícitas frente a las explícitas. Cuando mayor sea este valor, mayor será la velocidad del diálogo.

### 6.4.5.2 Modelado del usuario

Otro aspecto importante a considerar en todo Servidor Vocal Interactivo es la incorporación de técnicas de modelado de usuario. Estas técnicas tienen como objetivo adaptar la gestión del diálogo a la destreza del usuario en su interacción con el sistema. En un servicio por teléfono orientado al gran público, como es el que estamos desarrollando, es bastante difícil realizar esta adaptación debido a la gran cantidad de usuarios diferentes, y sobre todo, por la duración corta de la interacción. Esta duración no permite obtener medidas fiables para definir con certeza dicha adaptación, ni permite disponer de nuevos turnos de diálogo para ponerla en práctica. Aun así, es necesario establecer mecanismos sencillos que ofrezcan cierta adaptación con el fin de hacer más amigable el sistema, y sobre todo, evitar que ciertas llamadas se pierdan.

La técnica de modelado propuesta en la presente tesis está basada en la definición de niveles de destreza para cada uno de los aspectos a modificar de nuestro diálogo, y en la consideración de ciertos eventos que, ocurridos en el transcurso de la interacción, nos hacen cambiar de uno a otro nivel de destreza (Hirschberg et al, 2001). Veamos los diferentes aspectos a considerar en la técnica propuesta.

#### 1.- Aspectos modificables/adaptables del diálogo y definición de niveles de destreza para cada uno de ellos.

Los aspectos que podemos adaptar en un Servidor Vocal Interactivo, junto con los niveles de destreza que podemos definir son los siguientes:

- *Las preguntas que formula el sistema* al usuario para obtener los datos necesarios en la satisfacción de cada objetivo. En muchos casos una mala interpretación de la pregunta puede hacer que el usuario se quede callado o conteste algo no previsto en nuestro vocabulario. En este caso, nuestro sistema de reconocimiento cometerá un error y producirá que el usuario no confirme el valor ofrecido. A la hora de volver a preguntar, es importante no repetir exactamente la misma pregunta porque el usuario tenderá a repetir también la misma respuesta otra vez, y el problema persistirá. Ante esta situación, es mejor dar más información al usuario de forma que la nueva respuesta que obtengamos encaje con las previstas por nuestro sistema (Jokinen et al, 2001).

En este aspecto podemos definir 4 niveles de menor a mayor destreza:

- 1) En este primer nivel de destreza suponemos que el usuario no conoce cómo interactuar con el sistema. En este nivel las preguntas deben contener la siguiente información: una breve explicación de *cómo interactuar* con el sistema, el *dato* que se le está preguntando, los *posibles valores* a recoger y

*cómo especificar uno de ellos. Veamos el siguiente ejemplo de diálogo para preguntar el período del día en el que se desea viajar.*

*Sistema:* Recuerde *hablar después de escuchar la señal*. Diga el *período del día* en el que desea viajar: *por la mañana, por la tarde o por la noche*.

*Usuario:* por la mañana.

...

- 2) En un segundo nivel suponemos que el usuario sabe cómo interactuar con el sistema pero no conoce ni el dato, ni los valores posibles, ni la manera de especificar uno de ellos. En este caso la pregunta para el período del día podría ser:

*Sistema:* Diga el *período del día* en el que desea viajar: *por la mañana, por la tarde o por la noche*.

*Usuario:* por la mañana.

...

- 3) En el tercer nivel de destreza suponemos que el usuario no conoce el dato que se le va a preguntar pero que una vez preguntado, sabría los posibles valores aceptados por el sistema y la manera de especificar cada uno de ellos. En este caso la pregunta podría ser:

*Sistema:* Diga el *período del día* en el que desea viajar.

*Usuario:* por la mañana.

...

- 4) Por último en el cuarto nivel, el usuario conoce todos los aspectos y podemos relajar mucho la pregunta de forma que el diálogo se acorte. Este nivel sólo se suele producir cuando el usuario es un usuario experto o cuando se realizan varias consultas en una misma llamada.

*Sistema:* ¿Cuándo desea salir?

*Usuario:* por la mañana.

...

Un aspecto que debemos resaltar es que a medida que aumentamos el nivel de destreza las preguntas son más cortas y ágiles, ofreciendo un diálogo más vivaz

- *Las ayudas ofrecidas por el sistema.* En el caso de que un usuario tenga problemas en la interacción, debemos ofrecer informaciones y ayudas más completas que permitan al usuario conocer las limitaciones del sistema y las causas de error. De esta forma, el usuario puede llegar a entender el comportamiento del sistema ante sus respuestas, y corregirlas oportunamente si fuese necesario.

En este aspecto podemos definir 3 niveles de menor a mayor destreza:

- 1) En este primer nivel de destreza las ayudas ofrecidas al usuario contienen información sobre tres aspectos diferentes: descripción de las *causas generales de error en el reconocimiento* como condiciones de ruido elevado,

hablar antes del tono, hablar despacio,... También se incluye una *descripción explícita de las limitaciones* del sistema, como por ejemplo informar de que los precios ofrecidos son orientativos para un adulto y que no se ofrecen precios por tipo de usuario,... El tercer aspecto a considerar es *justificar la secuencia de preguntas* que se deben realizar: como por ejemplo explicar que para poder acceder a la base de datos necesitamos conocer la ciudad origen, la ciudad de destino y una fecha concreta de viaje.

- 2) En un segundo nivel de destreza se deberían omitir las explicaciones sobre las posibles causas de error en el reconocimiento pero no las limitaciones del sistema concreto implementado o la justificación de la secuencia de preguntas formuladas. Suponemos que cuando un usuario tiene un mínimo de experiencia en la interacción con este tipo de sistemas, únicamente necesita conocer las limitaciones del servicio concreto que está utilizando y no los aspectos que puedan ser generales a todos los servicios.
- 3) En el nivel más alto de destreza, el objetivo de las ayudas es recordar algunos aspectos del sistema sin entrar en justificar/explicar los siguientes pasos del diálogo, o entrar en una descripción de las posibles limitaciones.

De igual forma que en el diseño de las preguntas de los datos, la consideración de niveles de destreza más elevados, supone el diseño de ayudas más cortas y por tanto más ágiles.

- *Los mecanismos de confirmación.* En relación con este aspecto, podemos pensar que ante usuarios de mayor experiencia, podemos arriesgar un poco más haciendo confirmaciones más ligeras pero más difíciles de corregir (mayor porcentaje de confirmaciones implícitas). Los mecanismos de confirmación se han diseñado, como vimos en el apartado anterior, utilizando medidas de confianza para la combinación de varias estrategias. En esta combinación se definían umbrales  $\theta_i$  para la elección de una u otra estrategia. Modificando la posición de estos umbrales podemos cambiar el comportamiento de los mecanismos de confirmación haciéndoles más o menos conservadores. El número de niveles de destreza a considerar no debe ser muy elevado (2 o 3 niveles) para facilitar su diseño y ajuste.
- *Diseño de subdiálogos concretos.* En determinadas partes del diálogo nos podemos plantear varias posibilidades en la secuencia de interacciones dependiendo de la destreza del usuario. Por ejemplo a la hora de reconocer un dato complejo (formado por varios datos simples), en el caso de disponer de reconocimiento de habla continua, nos podemos plantear recoger varios datos simultáneamente o hacer varias preguntas para cada uno de los datos. El número de niveles que se pueden definir irá acorde con las posibilidades planteadas.

En estos puntos se han descrito todos los aspectos con los posibles niveles de destreza a definir en cada uno de ellos, pero la aplicación de esta técnica de modelado de usuario no obliga a considerar todos los aspectos y todos los niveles. En determinadas aplicaciones puede no ser demasiado útil considerar todos los niveles en

algunos de los aspectos, como por ejemplo considerar sólo dos niveles en la definición de las preguntas, o podemos no prever la adaptación en alguno de los aspectos, como por ejemplo que los mecanismos de confirmación no cambien con la destreza del usuario.

## **2.- Eventos indicadores de la destreza del usuario.**

Son medidas extraídas de la evolución del diálogo entre usuario y sistema que nos dan idea del nivel de destreza del usuario en su interacción. Estos indicadores son los que deciden o regulan los cambios de uno a otro nivel. En este punto veremos los principales eventos que se pueden considerar, y en el siguiente, veremos su aplicación a la selección del nivel de destreza.

Los eventos los podemos clasificar en positivos cuando apoyan el aumento del nivel de destreza y negativos cuando refuerzan una disminución del mismo. Los principales eventos positivos que podemos considerar son los siguientes:

- N° de veces que un usuario confirma la propuesta del sistema.
- N° de veces que un usuario corrige un posible error del sistema.
- Porcentaje de confirmaciones implícitas frente a explícitas.

Como eventos negativos podríamos considerar:

- N° de veces que un usuario no confirma la propuesta del sistema.
- N° de veces que el usuario no puede corregir un error del sistema.
- N° de veces que el usuario se queda callado ante la pregunta del sistema.
- N° de veces que pide que el sistema le repita la información.
- N° de veces que solicita volver a empezar.
- Porcentaje de confirmaciones explícitas frente al de implícitas.

La confianza media obtenida en el reconocimiento de los datos hasta ese punto del diálogo también nos puede reflejar una idea tanto de las condiciones de reconocimiento como de la destreza del usuario.

## **3.- Utilización de los eventos en la definición de los niveles de destreza.**

A la hora de definir el nivel de destreza de cada uno de los aspectos comentados en el punto 1, debemos considerar varios de los eventos descritos: no todos los eventos tienen porque influir sobre todos los aspectos. Veamos una posible clasificación:

- Para las preguntas del sistema podemos considerar como eventos positivos el número de veces que el usuario confirma lo reconocido por el sistema. Como eventos negativos consideramos el número de veces que el usuario se queda callado ante la pregunta, y el número de veces que no confirma lo reconocido.
- En relación con el tipo de ayuda a dar, podemos considerar todos los eventos descritos en el punto anterior. Según el tipo de ayuda a diseñar podrán tener más importancia unos u otros.

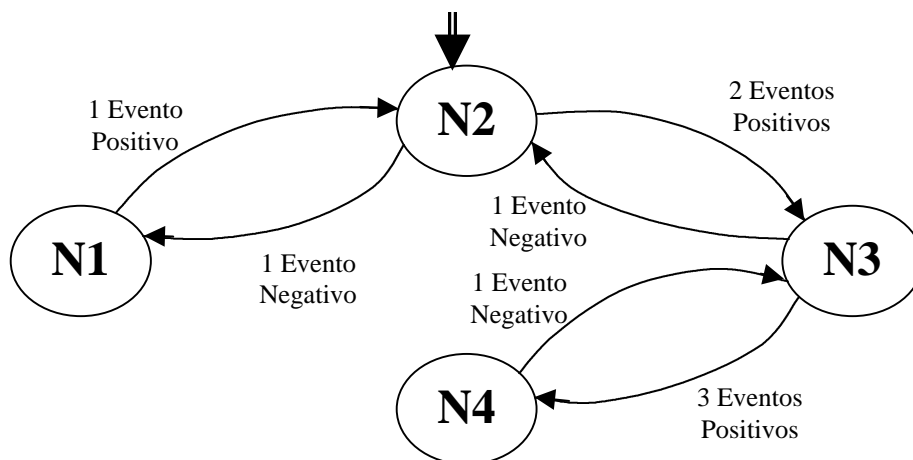
- Sobre las confirmaciones de los datos consideraremos como eventos positivos el número de veces que se confirma un dato positivamente, el número de veces que se corrige un dato con error y el porcentaje de confirmaciones implícitas frente al de explícitas. Por otro lado se podrán utilizar como eventos negativos el número de veces que no se confirma un dato, número de veces que no se puede corregir y porcentaje de confirmaciones explícitas frente al de implícitas.
- Para el diseño de subdiálogos concretos debemos apoyar estrategias con cierta iniciativa del usuario cuando tengamos mayor número de confirmaciones implícitas, mayor tasa de acierto (confirmaciones positivas) y mayor confianza media lo que nos asegura el buen funcionamiento del sistema de reconocimiento. Por otro lado se debe tender a estrategias de diálogo más guiadas cuando se haga mucho uso de confirmaciones explícitas, la tasa de reconocimiento no sea muy buena (confirmaciones negativas), el usuario no corrija con facilidad, o cuando la confianza media de los reconocimientos es baja.

Para combinar los diferentes eventos positivos o negativos de forma que obtengamos una única medida podemos utilizar una combinación lineal (ecuaciones 6-7 y 6-8) en la que se pueden definir diferentes pesos para controlar la importancia de uno u otro aspecto en la definición del nivel.

$$Eventos\ Total\ Positivos = \sum_{i=0}^N EventoPositivo_i \times Peso_i \quad (6-7)$$

$$Eventos\ Total\ Negativos = \sum_{i=0}^N EventoNegativo_i \times Peso_i \quad (6-8)$$

Veamos un ejemplo representado en la figura 6-17. Supongamos que estamos diseñando los niveles de destreza para las preguntas que realiza el sistema al usuario. En este caso tenemos 4 niveles definidos. El nivel activo en un punto del diálogo depende del estado inicial y del número de eventos positivos o negativos ocurridos a lo largo de la interacción hasta ese punto.



**Figura 6-17:** Diagrama de niveles de destreza en la técnica de modelado de usuario.

Al comienzo de la interacción el sistema ofrece una ayuda inicial y se sitúa en el segundo nivel por defecto. Cuando ocurre un evento negativo desciende al nivel 1, mientras que si ocurren dos eventos positivos el sistema aumenta hasta el nivel 3. Como podemos observar de la figura 6-17, a medida que aumentamos el nivel de destreza exigimos un mayor número de eventos positivos para seguir aumentando de nivel. De esta forma se evita que el sistema aumente demasiado deprisa el nivel, produciendo que ante la ocurrencia de un error sean necesarias muchas interacciones para bajar de nivel y recuperarse. Por otro lado, cuando detectamos un evento negativo debemos disminuir inmediatamente el nivel para ofrecer preguntas más explicativas que ayuden inmediatamente a corregir el posible problema. Si repitiésemos otra vez la misma pregunta el usuario podría contestar lo mismo y el problema se repetiría. Veamos un ejemplo de diálogo.

[El sistema está situado en el nivel 3]

*Sistema:* Diga el período del día en el que desea viajar.

*Usuario:* después de comer.

[El sistema reconoce por la noche]

*Sistema:* ¿Ha dicho por la noche?

*Usuario:* no.

[El sistema pasa del nivel 3 al 2]

*Sistema:* Diga el período del día en el que desea viajar: por la mañana, por la tarde o por la noche.

*Usuario:* por la tarde.

...

Para cada uno de los aspectos comentados en el punto 1, debemos definir un diagrama de transiciones con el número de eventos necesarios para transitar de uno a otro estado. En nuestro caso concreto únicamente hemos implementado la técnica para la definición de las preguntas de los datos puesto que este ha sido el aspecto más importante en nuestro sistema.

#### **4.- Medidas de evaluación del modelado de usuario.**

Para evaluar la adaptabilidad del modelo de usuario implementado podemos considerar como medida de evaluación la varianza del nivel de destreza a lo largo del diálogo. De esta forma una mayor varianza reflejará una mayor flexibilidad y adaptabilidad de la técnica. Por otro lado el nivel de destreza medio nos dará información sobre la agilidad del usuario y sobre la calidad de la interacción realizada.

### **6.4.5.3 Funcionalidad de carácter general**

En todo Servidor Vocal Interactivo debe estar disponible una funcionalidad de carácter general que resulta muy útil para mejorar la interacción usuario-sistema. Los aspectos a considerar son los siguientes:

- Debemos ofrecer la posibilidad de que el usuario pueda solicitar en cualquier punto del diálogo el comenzar la interacción desde el principio. Ante esta situación el sistema debe borrar toda la información obtenida y comenzar de

nuevo. De esta forma evitamos que el usuario tenga que colgar y volver a llamar con el coste de dinero que supone volver a establecer una llamada.

- Otro aspecto importante es prever mecanismos de corrección. Cuando en un sistema se incorporan estrategias de confirmación implícita o en algunos casos de no confirmación, debemos prever la posibilidad de que el sistema se equivoque y el usuario pueda corregirlo. Esta posibilidad debe ser conocida por el usuario, de forma que será necesario informarle con anterioridad o cuando se detecten ciertas expresiones que revelen un intento de corregir. En nuestro sistema se ha introducido el comando “corregir” de forma que el sistema supone que la última confirmación implícita ha sido errónea y vuelve a preguntar el último dato no confirmado.
- Dado que la interacción se realiza por teléfono, la información de la que dispone el usuario es estrictamente verbal. Por esta razón, el usuario puede despistarse y perder cierta información en cualquier punto. Para evitar que este despiste repercuta en el fracaso completo de la llamada, se debe ofrecer la posibilidad de que el usuario solicite repetición al sistema. En este caso el sistema debe repetir la última pregunta formulada o la última información ofrecida.
- La posibilidad de solicitar ayuda en cualquier punto del diálogo es una funcionalidad muy útil que permite al usuario conocer las posibilidades y limitaciones de los servicios automáticos en general y del servicio que se esté utilizando en particular.
- Otro aspecto importante es la posibilidad de suspender la interacción por unos momentos sin necesidad de cortar la llamada. Por ejemplo cuando estamos realizando una reserva de tren por teléfono y alguien nos interrumpe para hacernos una consulta, en lugar de colgar y volver a llamar, será muy útil la posibilidad de parar la interacción del sistema y retomarla después desde el mismo punto.

En la primera versión del sistema implementado, únicamente hemos considerado los tres primeros aspectos.

#### **6.4.5.4 Ajustes finales**

En este apartado, vamos a comentar algunos de los ajustes finales realizados durante esta última etapa de diseño basada en la mejora iterativa. Si bien, muchos de estos aspectos son dependientes del servicio concreto desarrollado, su exposición tiene como objetivo mostrar, sobre un caso concreto, la aplicación de algunas ideas que, aunque no son de carácter general, sí que pueden aplicarse en servicios similares.

El primer ajuste realizado corresponde con la manera de conseguir la fecha para un viaje en tren. En el apartado 6.4.4.3 describíamos la secuencia de preguntas que habíamos definido para la obtención de una fecha: a la hora de solicitar un día de la semana actual o de la semana siguiente, habíamos previsto que los usuarios iban a



utilizar el nombre del día (lunes, martes, miércoles,...) para definirlo. En los experimentos se observó que los usuarios también utilizaban el número del día del mes (el dieciocho, el veinte,...), con lo que hubo que considerar esta posibilidad.

Un aspecto que ofreció mucha robustez al servicio fue la comprobación de ciertas condiciones en relación con la fecha y la hora del viaje. Las condiciones fueron las siguientes:

- Cuando se especifica una fecha de viaje anterior a la actual, el sistema informa de tal hecho y sugiere la introducción de una nueva fecha.
- Cuando la fecha de salida corresponde al mismo día de la llamada, no se deben ofrecer trenes cuya hora de salida sea anterior a la actual. De forma que si el usuario está llamando al mediodía, no tiene sentido ofrecer información de los trenes que salieron por la mañana.
- De igual forma para el viaje de vuelta se debe comprobar que la fecha de vuelta es posterior o igual a la de ida, y en el caso de realizarse la vuelta el mismo día, la hora de salida para el viaje de vuelta debe ser posterior a la hora de llegada del viaje de ida.

La comprobación de este tipo de condiciones es muy sencilla y permite aumentar sensiblemente la robustez del sistema. De esta forma, se ofrece una mejor imagen al usuario puesto que ante problemas de reconocimiento, nunca se va a ofrecer una fecha o una hora ilógica.

Otro ajuste importante está relacionado con la opción de volver al comienzo del diálogo. Nuestro sistema, cuando reconoce alguna expresión como “volver a empezar” o “empecemos de nuevo”..., comienza la interacción desde el comienzo. En nuestro caso, en lugar de borrar toda la información obtenida hasta ese momento, empezamos desde el principio confirmando grupos de datos (pasos del diálogo). Cuando uno de estos grupos no es confirmado, el sistema comienza la interacción preguntando el primer dato del grupo no confirmado. Veamos el siguiente ejemplo:

*Sistema:* La opción elegida es un tren Intercity que sale a las 14:00 y llega a las 15:15.  
*Usuario:* volver a empezar.  
*Sistema:* Comencemos de nuevo la consulta.  
*Sistema:* ¿Desea viajar de Madrid a Barcelona?.  
*Usuario:* sí.  
*Sistema:* ¿Desea viajar el 19 de Julio por la noche?.  
*Usuario:* sí.  
*Sistema:* Entiendo que está interesado en un Intercity que sale a las 14:00 y llega a las 15:15 ¿Es correcto?.  
*Usuario:* no.  
*Sistema:* Bien, para el 19 de Julio dispone de 3 posibilidades, la primera opción...  
 ...

En el ejemplo, el dato no confirmado es la opción de tren elegida, luego el sistema comienza su interacción desde ese punto.

Por último, hay que comentar un ajuste realizado en la negociación de las opciones de tren. En nuestro caso, la estrategia de negociación consiste en que el sistema va presentando de 3 en 3 las opciones al usuario, y le pide que elija alguna de ellas o si prefiere que el sistema siga con las siguientes. En un principio, cada 3 opciones nuevas, el sistema las referenciaba como la primera, la segunda y la tercera opción del grupo. Esta solución complicaba enormemente la gestión para que el usuario pudiera retroceder y elegir alguna opción del grupo anterior. Esta posibilidad se puede ofrecer fácilmente referenciando las opciones de forma absoluta, y no de forma relativa a cada grupo ofrecido. Veamos el siguiente ejemplo de diálogo:

*Sistema:* La primera opción es..., la segunda es ..., y la tercera consiste en....

*Sistema:* Elija una opción de la primera a la tercera o diga continuar.

*Usuario:* continuar.

*Sistema:* La cuarta opción es..., la quinta es ..., y la sexta consiste en....

*Sistema:* Elija una opción de la primera a la sexta o diga continuar.

*Usuario:* la tercera.

...

En este caso, el usuario se ha decidido por la tercera opción una vez que ha escuchado la cuarta y no se ajustaba a sus expectativas.

Por último cabe comentar que en el apéndice F, se resumen las principales recomendaciones para el diseño de gestores de diálogo. Dichas recomendaciones han sido fruto del trabajo realizado durante la presente tesis doctoral y de otros trabajos previos relacionados con el diseño de gestores de diálogo.

## 6.4.6 Evaluación: pruebas de campo

En este apartado vamos a presentar los resultados de la evaluación de campo realizada sobre el sistema de información y reserva de billetes de tren. Esta evaluación ha consistido en diversas llamadas al sistema por parte de 105 usuarios con el fin de completar 4 escenarios de viaje. En total se obtuvieron 355 consultas puesto que no todos los usuarios completaron los 4 escenarios. En la información que se entregaba a cada usuario se le presentaban 4 escenarios tipo, pero se le sugería que generase 4 escenarios diferentes. Una vez completadas las llamadas, cada usuario debía rellenar un cuestionario (ver apéndice E). Las medidas de evaluación se obtuvieron de las anotaciones del sistema y del análisis de los cuestionarios. En los cuestionarios, se preguntaban principalmente aspectos subjetivos difíciles de medir con las anotaciones del sistema. En la tabla 6-10 se presentan las principales medidas obtenidas de las anotaciones realizadas por el sistema de forma automática.

Para poder interpretar estos resultados debemos saber que sólo el 19,5% de las consultas solicitaron información de un viaje de ida y vuelta, y que el 21,3% completó la reserva del viaje, tanto para casos de ida como viajes de ida y vuelta. Sólo en un 19,8% de las llamadas el usuario colgó sin recibir ningún tipo de información, luego el

porcentaje de llamadas en las que se ofreció información sobre los trenes fue de un 80,2%. Este valor es bastante alto. La razón de haber obtenido este valor es que más de la mitad de los usuarios que hicieron las pruebas (55 de los 105 totales) eran estudiantes de la universidad. Estos usuarios, conscientes de que se estaba evaluando el sistema, fueron más pacientes e intentaron completar todas las llamadas haciendo los intentos necesarios. Para estos usuarios la tasa de llamadas fue muy alta (93,7%) lo que influyó de manera importante sobre los resultados finales. En el apéndice G se hace un análisis detallado de la evaluación de campo dependiendo del tipo de usuario.

Medida	Valor
Duración media de la consulta (segundos)	195
Nº medio de preguntas realizadas por el sistema	18,83
Nivel medio de destreza del usuario	1,88
Varianza del nivel de destreza del usuario	0,32
% de confirmaciones implícitas	71,0%
% de confirmaciones explícitas	29,0%
Nº medio de veces que se vuelve a empezar en la mitad de una consulta	0,26
Nº de veces que el usuario solicita la corrección de un dato	0,32
Duración de la negociación de la opción del tren (segundos)	58
Nº medio de veces que el usuario solicita la repetición de las opciones de tren	0,18

**Tabla 6-10:** Medidas obtenidas de las anotaciones realizadas por el sistema.

Sobre las tasas de reconocimiento podemos decir que para vocabularios pequeños (menos de 50 palabras), como por ejemplo el día de la semana o las opciones de tren, se obtuvieron tasas superiores al 95%. Para el caso de la ciudad origen o destino del viaje, donde hemos utilizado un vocabulario de reconocimiento con 770 palabras, la tasa obtenida considerando los mecanismos de rechazo ha sido de un 87,2%. Para los errores, el 25,5% de los casos se resolvió con el segundo candidato de reconocimiento y el 43,6% con el reconocedor de palabras deletreadas. Para el resto de los casos fueron necesarias más interacciones. En estos experimentos obtuvimos un 3,4% de ciudades fuera del vocabulario, consiguiendo detectar con el reconocedor de palabras deletreadas (ver capítulo 5) sólo un 33,3%. En el resto de los casos los usuarios colgaron tras varios intentos.

Con la alta tasa de reconocimiento y la utilización de medidas de confianza ha sido posible utilizar de forma frecuente las confirmaciones implícitas en lugar de las explícitas. El porcentaje de confirmaciones implícitas es muy superior al de explícitas (ver tabla 6-10). En este análisis no hemos incluido las preguntas de Sí y No (no se confirman) y las respuestas que fueron rechazadas directamente por el sistema. Por otro lado, si tras una confirmación implícita el usuario solicita la corrección del dato, esta

confirmación ha sido considerada como si fuera una confirmación explícita, puesto que el número de preguntas es el mismo que en el caso de una confirmación explícita.

Como podemos ver, la duración media de la consulta es de 195 segundos, mayor que la duración media del servicio con un operador humano (150 segundos), pero similar a otros servicios parecidos que utilizan también reconocimiento de habla aislada (Baggia et al, 2000). En cuanto a la negociación, cabe resaltar que tras los cambios realizados en la fase de simulación y los ajustes realizados en la fase de mejora iterativa, hemos conseguido reducir sensiblemente su duración media de 107 seg. a 58 seg.

Otro aspecto importante es la utilización del comando corregir. Con este comando hemos evitado que el usuario tenga que volver al principio (esta opción es poco utilizada: 1 de cada 4 llamadas). El número tan reducido de veces que los usuarios solicitan la repetición de las opciones de tren nos da una idea de que los usuarios no tienen excesivos problemas para retener 3 opciones de tren simultáneamente.

En cuanto al cuestionario, se han analizado aspectos subjetivos pidiendo al usuario que evalúe cada uno de ellos de 1 a 5. El cuestionario utilizado se puede consultar en el apéndice E y los resultados medios obtenidos se presentan en la tabla 4-11.

Medida subjetiva	Valor
El sistema comprende lo que le dices.	3,1
Las respuestas del sistema son claras y concisas.	3,4
Entiendo lo que el sistema me dice.	3,5
Se accede a la información de trenes rápidamente.	2,9
El sistema es fácil de usar.	3,5
Es fácil de aprender su funcionamiento.	3,7
El sistema me ayuda durante la conversación. Tengo información de lo que está haciendo el sistema.	3,1
En caso de error la corrección fue fácil.	2,9
El sistema me hace las preguntas en un orden lógico.	3,6
En general, es un buen sistema.	3,0

**Tabla 6-11:** Medidas subjetivas recogidas de los cuestionarios.

La experiencia media de los usuarios, evaluada de 1 a 5, fue de 2,8, un valor menor de la media (3), lo que indica una experiencia no muy alta de los usuarios que probaron el sistema. En los resultados presentados en la tabla 6-11, podemos ver cómo muchos de los aspectos obtienen una puntuación superior a la media (3), lo que indica que el sistema está funcionando razonablemente bien. Los aspectos peor valorados son la agilidad o rapidez del sistema y los mecanismos de corrección. Esto se debe a que no se dispone de módulos de reconocimiento de habla continua y comprensión que permitiesen por un lado, agilizar más la interacción, y por otro, permitir al usuario que corrigiese los

errores con expresiones de lenguaje natural (ej: no, Madrid no, Cádiz). Las mejores puntuación se obtuvieron para la inteligibilidad del sistema y para la estructura del diálogo. Este hecho ha sido posible al disponer de un conversor texto-voz de voz femenina para dominios restringidos (Montero et al , 2000) y al análisis del diálogo realizado en la presente tesis.

Los aspectos del diálogo que más resaltan los usuarios como problemas encontrados en el sistema son los siguientes (ordenados de mayor a menor frecuencia): reconocimiento de la fecha, corrección de los errores de reconocimiento y reconocimiento de las ciudades origen y destino.

A la pregunta sobre qué mejoraría del sistema los aspectos más comentados por los usuarios han sido los siguientes: introducir mayor naturalidad en la voz sintética, ofrecer mayor flexibilidad en las respuestas de los usuarios, y agilizar la interacción para permitir un acceso a la información más rápido.

Al final del cuestionario, se preguntaba sobre la preferencia de los usuarios a la hora de obtener la información de trenes entre varias alternativas: el 35,2% prefería llamar a un sistema como este, el 35,2% por acceso web y el 25,7% prefirió la opción de ir a preguntar a la estación (el 3,9% insistió en la opción de operadores humanos).

En el apéndice G, se hace un análisis más detallado de la evaluación realizada, ofreciendo resultados diferentes según el tipo de usuario involucrado en la evaluación.

## 6.5 Conclusiones

En este apartado se comentan y resumen las principales conclusiones obtenidas de este capítulo. La principal conclusión obtenida es la propuesta y evaluación de una metodología para el diseño de gestores de diálogo en Servidores Vocales Interactivos. La presentación de esta metodología se realiza sobre la aplicación al caso de un servicio de información y reserva de billetes de tren. Esta metodología está formada por 5 fases. En la primera fase se realiza un análisis de la base de datos que contiene la información para ofrecer el servicio. Este análisis consiste en la descripción del diagrama Entidad-Relación que ofrece una representación semántica de los datos. La definición de los conjuntos entidad y de sus atributos clave, juega un papel importante en la definición posterior del diálogo. En la etapa de diseño por intuición, proponemos la técnica de “braim-storming” realizada sobre el diagrama Entidad-Relación con la idea de proponer diferentes objetivos que se pueden ofrecer en el servicio, los datos a preguntar al usuario para satisfacer cada objetivo y la secuencia para preguntar dichos datos.

En la fase de diseño por observación, es necesario grabar conversaciones entre los usuarios y operadores humanos, y transcribirlas. Utilizando estas grabaciones podemos evaluar cada una de las alternativas sugeridas en la etapa de diseño por intuición sin haber realizado aún ninguna implementación del sistema. El problema que surge de este análisis es que las interacciones usuario-usuario son diferentes de las interacciones usuario-sistema, luego podremos hacer análisis de aspectos generales pero no de detalles muy concretos de la interacción.

En la fase de simulación utilizamos la herramienta de Mago de Oz para simular una interacción usuario–sistema sin disponer del sistema completo. Esta técnica nos permite probar y evaluar diferentes estrategias de diálogo con usuarios reales. En esta simulación, el aspecto más importante es dotar al Mago de herramientas para que su tiempo de respuesta sea reducido y realmente simule a un sistema automático en la realización de su labor. En esta fase proponemos varias medidas obtenidas de las anotaciones del sistema y de preguntas en los cuestionarios, para evaluar cada una de las propuestas de diálogo probadas.

En la etapa de mejora iterativa se describe la utilización de medidas de confianza para la gestión de los mecanismos de confirmación. Además, se presentan recomendaciones para el diseño de las frases de confirmación y medidas de evaluación de los mecanismos definidos. En esta etapa, también se describe la técnica de modelado de usuario implementada para adaptar el funcionamiento del sistema a la destreza demostrada por el usuario a lo largo de su interacción. En el apartado de funcionalidad general de esta etapa, se proponen dos estrategias para la recuperación de errores en el caso de confirmaciones implícitas: la opción de corregir el dato anterior, y la posibilidad de volver al principio confirmando grupos de datos conjuntamente hasta situarse en el punto del diálogo deseado.

Considerando los resultados de evaluación presentados en el apartado 6.4.6, podemos concluir que la metodología propuesta nos ha permitido implementar un sistema con buena aceptabilidad por parte de los usuarios. Los usuarios dieron un valor medio de 3,0 en una escala de 1 a 5.

En este punto es necesario comentar que el sistema puede ser mejorado con la incorporación de módulos para el reconocimiento de habla continua y compresión de lenguaje natural. El punto del diálogo donde sería importante su incorporación es en el reconocimiento de la fecha.