

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 403 014**

51 Int. Cl.:

G06F 17/30

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **03.05.2001** **E 01931756 (9)**

97 Fecha y número de publicación de la concesión europea: **17.10.2012** **EP 1297451**

54 Título: **Provisión de servicios en una red de comunicaciones**

30 Prioridad:

06.06.2000 FI 20001352

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

13.05.2013

73 Titular/es:

**TELIASONERA FINLAND OYJ (100.0%)
TEOLLISUUSKATU 15
00510 HELSINKI, FI**

72 Inventor/es:

**KATAIKKO, MIKA;
SIITONEN, TOMMI y
NISSINEN, ILKKA**

74 Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

ES 2 403 014 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Provisión de servicios en una red de comunicaciones

Campo de la invención

- 5 La invención se refiere en general a la provisión de servicios en una red de comunicaciones. Más específicamente la invención se refiere a un procedimiento y un sistema para proporcionar servicios a los que pueden acceder los usuarios de la red enviando peticiones de servicio en un lenguaje natural.

Antecedentes de la invención

- 10 La técnica de antecedentes relevantes se trata en los documentos US 5895466, US 5913214 y WO 99/53676. El documento US 5895466 desvela un sistema de servicio de clientes que incluye un dispositivo de lenguaje natural, un dispositivo remoto acoplado remotamente al dispositivo de lenguaje natural sobre una red y una base de datos acopada al dispositivo de lenguaje natural. El documento WO 99/53676 desvela un procedimiento de usar una pregunta de lenguaje natural para recuperar información desde una fuente de datos y además realizar una acción solicitada usando la información recuperada. El documento US 5913214 desvela un sistema para preguntar a fuentes de datos heterogéneas, dispares sobre una red.
- 15 El fuerte crecimiento en el número de usuarios y en los servicios de Internet proporcionados a través de la Internet ha sido uno de los fenómenos más notables en las comunicaciones en los últimos años. Otra tendencia actual es el fuerte incremento en el uso de diversos terminales móviles, tal como los ordenadores portátiles, los equipos PDA (Asistente Digital Personal), y los teléfonos inteligentes.
- 20 Estas dos tecnologías de red que evolucionan rápidamente, las comunicaciones inalámbricas y la Internet, están convergiendo gradualmente para hacer disponibles los servicios de datos de paquetes conmutados usados en la Internet a los usuarios móviles. A medida que aparecen en el mercado los sistemas y servicios con una complejidad en aumento, se está haciendo más crucial la necesidad de interfaces hombre - máquina más amigables para el usuario para la utilización eficiente de los servicios.
- 25 En el presente la mayor parte de las interfaces hombre - máquina son relativamente complicadas y difíciles de usar. En los servicios móviles típicos, por ejemplo, el usuario tiene que recordar las semánticas y sintaxis requeridas para la interfaz del servicio. La tolerancia a errores de estos sistemas es pobre; incluso errores menores de deletreo, por ejemplo, impiden el uso del servicio. Es por lo tanto deseable que el usuario pueda expresar el comando o pregunta de forma tan natural como sea posible, es decir usando un lenguaje natural.
- 30 Para superar este problema, se han introducido en el mercado las interfaces de lenguaje natural, que permiten al usuario expresar sus intenciones en el lenguaje natural. Un sitio Web conocido que permite a los usuarios buscar en la Internet en Inglés llano es el servicio Ask Jeeves (disponible en www.askjeeves.com). Este servicio se basa en una combinación de una interpretación de lenguaje natural y una base de conocimientos propietaria. La naturaleza precisa de la pregunta se analiza en primer lugar usando un procesamiento tanto semántico como sintáctico, y a continuación se buscan respuestas a partir de la base de conocimientos que contiene información acerca de la mayor parte de las preguntas formuladas frecuentemente.
- 35 Otro sistema conocido basado en la interpretación del lenguaje natural se desvela en la Solicitud de Patente Internacional WO 99/66381, que desvela una arquitectura software adaptativa orientada a un agente para las interfaces hombre - máquina. En este sistema, el mensaje de lenguaje natural se introduce a una red de agentes de software. Uno o más de los agentes actúan como agentes de entrada que reciben el mensaje de lenguaje natural. Los diferentes agentes en la red están "hiper-conectados" entre sí en el sentido de que reciben preguntas y comandos sólo desde un subconjunto predeterminado de otros agentes en la red y transmiten las preguntas y comandos solo a otro subconjunto de agentes en la red.
- 40 Un inconveniente principal en los presentes sistemas basados en la interpretación del lenguaje natural es que la interpretación de la entrada de usuario está fuertemente ligada a la implementación del resto del sistema. En otras palabras, se hace una interpretación solo para el servicio proporcionado por el resto del sistema. Por lo tanto un proveedor de servicios que ofrece una amplia diversidad de servicios dentro de una estructura de servicio única no ha podido utilizar la interpretación del lenguaje natural de forma eficiente para proporcionar los servicios.
- 45 Es un objeto de la invención obtener una solución por medio de la cual es posible eliminar los inconvenientes descritos anteriormente y ofrecer una solución que posibilite no solo un uso no complicado de una amplia diversidad de diferentes servicios sino también una provisión eficiente de nuevos servicios y combinaciones de servicios.
- 50

Sumario de la invención

Este y otros objetos de la invención se logran de acuerdo con los principios de la presente invención proporcionando un sistema en el cual el usuario puede acceder a una amplia red de servicios suministrando peticiones de servicio en un lenguaje natural. La idea de la invención es separar la interpretación del lenguaje natural de la implementación y

la ejecución del servicio utilizando una capa intermedia que en primer lugar genera una descripción del servicio de formato genérico a partir de la entrada de usuario y a continuación analiza esta descripción dentro de la información indicando las llamadas al servicio a realizar y los parámetros relacionados con dichas llamadas. Esta información se devuelve a un componente de la lógica del servicio que a continuación implementa el servicio en un modo conocido tal como realizando dichas llamadas de servicio. De este modo, la entrada del lenguaje natural se convierte en primer lugar en una descripción del servicio que describe las funcionalidades del servicio requeridas por el mensaje de lenguaje natural. Esta descripción del servicio se saca en un lenguaje de marcación generalizado. La descripción se analiza a continuación en los elementos que identifican las operaciones (tales como las llamadas del servicio) a realizar y los parámetros relacionados con estas operaciones, por lo que el servicio se puede proporcionar en un modo convencional.

De este modo la interpretación del lenguaje natural se puede usar en la provisión del servicio de un modo eficiente de modo que se pueden proporcionar no solo una amplia diversidad de servicios sino también combinaciones arbitrarias de los servicios, en respuesta a una entrada de usuario. De este modo, el usuario puede obtener una amplia diversidad de servicios nuevos y existentes sin tener que recordar la semántica y la sintaxis requeridas por cada una de las interfaces del servicio. Un proveedor del servicio que ofrece una diversidad de servicios separados puede de este modo ofrecer nuevos servicios combinando los servicios de un modo determinado por el mensaje de entrada desde el usuario.

Breve descripción de los dibujos

En lo siguiente se describen la invención y sus realizaciones preferidas refiriéndonos más estrechamente a los ejemplos mostrados en las Figuras 1 a 7 en los dibujos adjuntos, en los que:

la Figura 1 ilustra la arquitectura general del sistema de acuerdo con la presente invención.
la Figura 2 ilustra un mensaje de petición de servicio que el usuario introduce desde su terminal,
la Figura 3 ilustra un ejemplo de transferencia de mensaje entre los elementos del sistema en respuesta a una petición de servicio de la Figura 2,
la Figura 4 ilustra la estructura funcional del interprete de lenguaje natural de acuerdo con la presente invención,
la Figura 5 ilustra un ejemplo de la salida desde el interprete de lenguaje natural,
la Figura 6 ilustra un diagrama de flujo que describe el funcionamiento del analizador de XML, y
la Figura 7 ilustra el funcionamiento del componente de la lógica del servicio en la provisión del servicio en respuesta a la salida del analizador de XML.

Breve descripción de la invención

La Figura 1 muestra la arquitectura del sistema general de acuerdo con la presente invención. El sistema incluye varios servidores que proporcionan servicios a los usuarios del sistema. Dos de estos servidores (S1 y S2) se han mostrado en la figura. Los servidores ofrecen una amplia red de recursos y servicios para los usuarios de la red y los servidores pueden pertenecer a uno o más proveedores de servicios. En este ejemplo, todos los servidores son nodos de la Internet o una red de Intranet (o una red de TCP/IP equivalente). En este contexto, el término "servicio" se refiere a una aplicación de servicio que reside en un servidor. Más adelante, el conjunto de servidores que proporcionan los servicios se llaman también la estructura de servicio.

Los terminales del usuario (UT1 y UT2) tienen acceso a los servidores que proporcionan los servicios de un modo conocido como tal. Los terminales del usuario pueden ser terminales fijos, como se muestra en conjunción con el terminal UT1, o los terminales móviles que tienen acceso inalámbrico al sistema o a la red, como se muestra en conjunción con el terminal UT2. El acceso inalámbrico se puede implementar a través de diversos puntos de acceso (AP1) o alternativamente a través de puertas de enlace (GW1) que traducen las peticiones desde el protocolo utilizado entre el terminal y la puerta de enlace, tal como WAP, al protocolo usado entre la puerta de enlace y el servidor, es decir el HTTP.

En términos de la idea inventiva, los tipos de terminales del usuario y las conexiones entre los terminales de usuario y los servidores que proporcionan los servicios no son significativos. En este contexto la única característica esencial de los terminales es que se proveen con buscadores u otros software de cliente conocidos por medio de los cuales pueden comunicar con los servidores de provisión del servicio de la red.

El sistema comprende además un componente de la lógica del servicio SL, un intérprete del lenguaje natural NLI y un analizador PS. Estas tres unidades juntas forman el núcleo del sistema de acuerdo con la presente invención. Por medio de este núcleo, los usuarios pueden usar los servicios dentro de la estructura del servicio eficientemente suministrando peticiones del servicio de lenguaje natural. El componente de la lógica del servicio recibe las peticiones del servicio desde los terminales del usuario, redirige las peticiones al intérprete del lenguaje natural, recibe los resultados de la interpretación desde el intérprete, y redirige los resultados al analizador para un procesamiento ulterior. Además, el componente de la lógica del servicio recibe los resultados del análisis desde el analizador y, en respuesta a los mismos, ejecuta los objetos que realizan llamadas a los servicios deseados dentro de la estructura del servicio. El componente de la lógica del servicio actúa a continuación de este modo como un primer centro que coordina la interpretación y el análisis de la petición de servicio, y a continuación implementa el

servicio por medio de los resultados analizados. El intérprete de lenguaje natural interpreta la entrada de lenguaje natural desde el usuario y convierte los resultados interpretados en una descripción del servicio de formato genérico que describe la funcionalidad del servicio requerida para cumplir la petición del usuario.

- 5 La Figura 3 ilustra un ejemplo de transferencia de mensajes entre los elementos del sistema cuando el usuario usa un servicio que reside dentro de la estructura del servicio. En este contexto, se usa como ejemplo un servicio que proporciona información de contacto que ilustra la provisión del servicio de acuerdo con la presente invención.

En primer lugar, el usuario da el comando o mensaje de entrada en un lenguaje natural tecleando o suministrando de otro modo el comando o mensaje deseado desde su terminal. La Figura 2 ilustra un ejemplo de un mensaje que el usuario podría suministrar a su terminal para encontrar la información de contacto de una cierta persona (es decir Jaco Väisänen). El usuario envía a continuación este mensaje a la lógica de servicio SL (etapa 301 en la Figura 3) que redirige el mensaje al intérprete del lenguaje natural NLI (etapa 302). El intérprete analiza la entrada del mensaje de lenguaje natural introducido por el usuario (etapa 303) y saca una descripción de servicio en un lenguaje de marcación generalizado, preferiblemente en el formato XML. El XML (Lenguaje de Marcación Extensible) es un lenguaje normalizado diseñado para facilitar la representación y uso de los documentos estructurados y datos en la Web. La interpretación como tal se realiza en un modo conocido. La descripción del servicio del formato genérico incluye los nombres de las funcionalidades de servicio que se necesitan para cumplir la petición del usuario, y los parámetros (suministrados por el usuario) que requieren dichas funcionalidades. La descripción del servicio se devuelve a continuación a la lógica del servicio en la etapa 304. Sobre todo, la descripción del servicio se especifica como una aplicación de XML.

20 La lógica del servicio envía a continuación al analizador, que es ahora un analizador de XML, una petición para analizar la descripción XML recibida desde el intérprete (etapa 305). La petición incluye la cadena de caracteres del formato XML recibida desde el intérprete. El analizador analiza la cadena de caracteres y, en base a las identidades encontradas en la cadena de caracteres, ejemplifica las clases que corresponden a las identidades encontradas. Los ejemplos creados se llaman objetos, como es común en el campo de la programación orientada a objetos. Los objetos incluyen el código de programa que corresponde a las funcionalidades especificadas en la descripción de XML (etapa 306). En este ejemplo, el analizador encuentra en la descripción XML el nombre de la funcionalidad que describe la función de recuperación de la información de contacto desde una base de datos (etapa 306). El analizador devuelve a continuación los objetos correspondientes a la lógica del servicio en la etapa 307. La lógica del servicio almacena los objetos en el orden correspondiente al orden de ejecución de dichas funcionalidades. Un objeto que proporciona una cierta funcionalidad típicamente incluye el envío de una petición de servicio al servidor que proporciona el servicio en cuestión (etapa 309) y que genera una respuesta al usuario a partir de los resultados del servicio a devolver al usuario (etapa 311). La lógica del servicio ejecuta un objeto en un momento en la etapa 308. La Figura 3 ilustra la ejecución de un único objeto, es decir, la Figura 3 muestra solo un par de servicio petición / respuesta.

35 El analizador ejemplifica a continuación las clases que implementan las funcionalidades indicadas por la descripción de XML. Estos ejemplos (es decir, objetos) se devuelven a la lógica del servicio SL que a continuación ejecuta las funcionalidades llamando a los objetos en el orden correcto.

El funcionamiento de la provisión de servicio se desvela a continuación con más detalle desvelando los elementos individuales del sistema.

40 La Figura 4 ilustra la estructura funcional del intérprete de lenguaje natural. El intérprete comprende una red de agentes o comunidades de agentes. En la figura cada uno de los óvalos representa un agente. El mensaje de lenguaje natural lo suministra el usuario a un agente de entrada 41, mensaje que en este ejemplo es igual que el mostrado en la Figura 2. Los diferentes agentes en la red están conectados entre sí de modo que un agente individual recibe peticiones o comandos solo desde un subconjunto predeterminado de otros agentes y transmite preguntas o comandos solo a otro subconjunto de agentes. Cada uno de los agentes tiene un dominio de responsabilidad en la interpretación del mensaje de lenguaje natural. Cada funcionalidad (es decir cada servicio) proporcionado por el sistema tiene asignado un agente responsable para ese servicio, y cada agente es preferiblemente responsable solo de un servicio. Además, parte de los agentes son responsables de la interpretación de los parámetros requeridos por los servicios. Por ejemplo, uno de los agentes es un agente de la información de contacto (42) responsable del servicio de la información de contacto tomada como un ejemplo de los servicios. Este agente recibe todos los mensajes con palabras relativas al servicio de la información de contacto, tal como las palabras "información de contacto", "número de teléfono", etc. Además, uno de los agentes puede ser un agente de nombre completo (43) que es responsable de los parámetros requeridos por el servicio de la información de contacto y de este modo recibe y procesa los nombres de las personas cuya información de contacto se busca.

55 La red de agentes se puede implementar de forma similar a como se describe en la Solicitud de Patente Internacional mencionada anteriormente WO 99/66381, por lo que la implementación de los agentes no se trata con más detalle en este punto. Además de la implementación descrita en dicha publicación WO, los agentes individuales se han designado para generar una descripción del formato XML del resultado de la interpretación. De este modo, el agente de la información de contacto genera elementos XML indicando que la funcionalidad en cuestión es el servicio de información de contacto y el agente de nombres completos genera los parámetros del formato XML sobre

la base del nombre de la persona cuya información de contacto se busca. Todos los elementos de la descripción XML generados se redirigen a un agente de la norma XML 44 que tiene una definición de la norma XML asociada. Usando esta definición, el agente de la norma XML establece los elementos generados en el orden correcto de modo que se obtiene una descripción XML del servicio deseado. De este modo, esta descripción se ha generado sobre la base de la entrada de lenguaje natural suministrada por el usuario.

En lugar de una red de agentes de software se pueden usar otras soluciones para la interpretación. La arquitectura del sistema no establece ningún requisito para la estructura del intérprete, excepto que la salida esté en un lenguaje de marcación generalizado, preferiblemente en XML.

La Figura 5 ilustra un ejemplo de descripción XML obtenida como resultado de la entrada de usuario de acuerdo con las Figuras 2 y 4. En la Figura 5, la primera fila contiene la declaración XML y la segunda fila la declaración del tipo de documento. La tercera fila incluye el elemento raíz que contiene todos los demás elementos del documento. Como se conoce comúnmente, cada uno de los elementos comprende una etiqueta de comienzo, una etiqueta de fin y el contenido entre las etiquetas. La cuarta fila indica que el nombre de la funcionalidad en cuestión es "buscar_contacto", la fila que genera el agente de la información de contacto en la Figura 4. La quinta fila indica que la funcionalidad tiene un parámetro asociado cuyo nombre es "NombreCompletoContacto" y cuyo valor es en este caso "Jaako Väisänen". Esta fila la ha generado el agente de nombres completos en la Figura 4.

De este modo, en esta etapa el intérprete de lenguaje natural ha interpretado el mensaje de lenguaje natural original de modo que la funcionalidad involucrada es "buscar_contacto", que es el servicio de la información de contactos dentro del sistema, y que el elemento de la información de contacto a buscar es donde el parámetro "NombreCompletoContacto" tiene el valor "Jaako Väisänen". Los resultados interpretados están en un formato genérico XML, como se muestra en la Figura 5.

La Figura 6 es un diagrama de flujo que ilustra la operación del analizador de XML. El analizador recibe en primer lugar la descripción del servicio de formato XML, tal como la descripción de acuerdo con la Figura 5, desde el componente de la lógica del servicio (etapa 610). El analizador divide a continuación la cadena de caracteres recibida en elementos XML (etapa 612) de acuerdo con las etiquetas dentro de la cadena de caracteres, es decir las etiquetas separan la cadena de caracteres en los elementos. Un elemento comprende los artículos entre la etiqueta del tipo de comando (<comando...>) y la siguiente etiqueta de fin de comando (</comando>). De este modo, en el ejemplo de la Figura 5 el analizador encuentra solo un elemento que incluye los artículos desde la cuarta fila a la fila séptima.

Los elementos, que incluyen los nombres de las funcionalidades y los valores de los parámetros asociados, se analizan a continuación y se examina el tipo de cada uno de los elementos en la etapa 612, un elemento cada vez. Una vez determinado el tipo del primer elemento, el analizador ejemplifica la clase correspondiente al tipo encontrado (etapa 613). De este modo, en esta etapa el analizador mapea la descripción del formato XML a una clase que representa una implementación del servicio que se puede ejecutar por el componente de la lógica del servicio. Para este propósito, el analizador puede usar cualquier procedimiento de mapeo proporcionado por el entorno de desarrollo (por ejemplo una reflexión en Java). Cada uno de los atributos de tipos mapeados del elemento de XML es una clase y cada uno de los atributos de tipos corresponde a una clase. Una clase corresponde a una funcionalidad. El analizador crea una instancia desde cada una de las clases encontradas en base a los atributos de tipos. Como se ha mencionado anteriormente, los ejemplos de las clases se llaman en este punto objetos.

Después de esto, el analizador comprueba si quedan elementos en la cadena de caracteres de XML analizada (etapa 614). En caso afirmativo, el procedimiento vuelve a la etapa 612 para examinar el tipo del siguiente elemento. Cuando se han ejemplificado las clases correspondientes a todos los tipos encontrados, el analizador envía los resultados del análisis a la lógica del servicio SL (etapa 615). Los resultados del análisis incluyen los objetos que corresponden a las funcionalidades descritas en la descripción XML. Estos objetos incluyen además los valores de parámetros asociados con cada una de las funcionalidades.

La Figura 7 ilustra un ejemplo de la operación de la lógica del servicio SL cuando recibe una respuesta analizada desde el analizador de XML. El lado izquierdo de la figura muestra el funcionamiento de la lógica del servicio en un formato de diagrama de flujo. Los círculos en el diagrama de flujo ilustran los puntos en los que se envían los mensajes a la red. Una vez recibidos los objetos desde el analizador (etapa 710), la lógica del servicio SL los almacena en el orden de ejecución (etapa 711). Después de esto, la lógica del servicio SL llama al procedimiento de ejecución del primer objeto en la etapa 712. Cada una de las clases tiene un procedimiento de ejecución por si mismo para la implementación de la funcionalidad correspondiente. Cuando se llama a este procedimiento, se ejecuta la funcionalidad de acuerdo con la clase en cuestión. En otras palabras, se envía un comando del servicio al servidor que proporciona el servicio en cuestión (etapa 309 en la Figura 2). Cuando se recibe una respuesta (es decir, en este ejemplo la información del contacto de Jaakko Väisänen) desde el servidor Sn que proporciona el servicio de la información de contacto (etapa 310 en la Figura 3), la lógica de servicio SL comprueba si quedan objetos cualesquiera (etapa 714). Si este es el caso, la lógica del servicio procesa los objetos restantes de forma similar, es decir el procedimiento vuelve a la etapa 712 para llamar al procedimiento de ejecución del siguiente objeto. En general, cuando se han ejecutado todos los objetos, la lógica de servicio SL convierte las respuestas

generadas de cada uno de los objetos ejecutados en un formato compatible con el tipo del terminal de usuario. Típicamente, la lógica del servicio genera una página HTML sobre la base de los resultados recibidos y la envía al usuario (etapa 715 en la Figura 7 y etapa 311 en la Figura 3).

Además de la ejecución sencilla descrita anteriormente, la funcionalidad puede requerir la interacción del usuario en el medio del procedimiento descrito anteriormente. Esto puede ocurrir si el objeto del comando actualmente en ejecución detecta que no tiene toda la información necesaria para la ejecución de su funcionalidad, es decir, los objetos anteriores no produjeron suficiente información para completar la ejecución del objeto actualmente en ejecución. La lógica del servicio inicia a continuación una interacción del usuario enviando al usuario un mensaje solicitando información adicional. La respuesta desde el usuario puede iniciar a continuación la ejecución de nuevos objetos antes de la que pueda continuar la ejecución actual. Estos nuevos objetos se almacenan a continuación en el mismo almacenamiento con los objetos restantes y se ejecutan antes de que continúe la ejecución actual.

La lógica de servicio SL recibe de este modo al menos un objeto desde el analizador, el objeto que incluye el código para la funcionalidad indicada en la descripción del servicio de XML. Una vez recibidos los objetos junto con los parámetros asociados con cada uno de los objetos, la lógica del servicio SL tiene una descripción completa de la funcionalidad del servicio a implementar. En otras palabras, la lógica del servicio ahora tiene el conjunto de objetos que incluye el código a ejecutar para obtener la funcionalidad deseada, tal como la recuperación de la información de contacto de una persona, así como los valores de parámetros a usar. En lugar de devolver los datos binarios ejecutables a la lógica de servicio, es posible optimizar el funcionamiento del sistema enviando a la lógica del servicio sólo referencias a los objetos (es decir, punteros). De hecho, las herramientas comerciales que se pueden usar para implementar el sistema típicamente operan de este modo (es decir, todas las variables y objetos son referencias).

Para dar un ejemplo del funcionamiento descrito anteriormente, el nombre de la clase correspondiente al servicio de la información de contacto podría ser "BuscarContacto" y el procedimiento de ejecución de esta clase "BucarContacto.ejecutar()", por ejemplo. Este procedimiento de ejecución podría obtener un comando de servicio

```
resultadoServicio = LibroContactos.buscarContacto ("Jaakko Väisänen"),
```

es decir, podría llamar a un procedimiento de búsqueda de contacto de un servicio llamado LibroContactos, que reside en el servidor que proporciona el servicio de información de contacto, suministrando los parámetros analizados por el analizador de XML (es decir NombreCompleto = "Jaakko Väisänen") en la llamada. En respuesta a la clase "BuscarContacto" recibe desde el procedimiento del servicio LibroContactos los resultados a partir de los cuales la clase genera una respuesta al usuario usando su propio procedimiento "generarRespuestaUsuario":

```
respuestaUsuario = generarRespuestaUsuario (resultadoServicio),
```

donde resultadoServicio es la información de contacto de Jaakko Väisänen.

Esta respuesta se entrega (es decir se hace compatible) a los diferentes tipos de terminales de usuarios por medio de un procedimiento de la lógica del servicio "SCL.resultado2Terminal":

```
respuestaUsuarioEntregada = SCL.resultado2Terminal (respuestaUsuario),
```

Después de esto, el resultado entregado que en este caso es una página HTML que contiene la información de contacto de Jaakko Väisänen, se envía al usuario.

En los ejemplos anteriores se asumió que el intérprete del lenguaje natural pudo interpretar la entrada de usuario, es decir que la entrada del usuario incluía suficiente información para una interpretación satisfactoria. Si la entrada no incluye suficiente información o si no se puede interpretar sin ambigüedades, el intérprete del lenguaje natural activa una interacción en la cual se envía en primer lugar un mensaje informando de la situación a la lógica del servicio. La lógica del servicio envía a continuación una petición de interacción al usuario, es decir el procedimiento salta de la etapa 303 a la etapa 311.

Como se ha ilustrado anteriormente, el sistema de acuerdo con la presente invención incluye una capa intermedia entre el usuario y los servicios. Esta primera capa genera una descripción del servicio de formato genérico a partir de la entrada del usuario y a continuación analiza esta descripción dentro de entidades que indican las llamadas de servicio que tienen que realizarse y los parámetros relativos a cada una de las llamadas de servicio a realizar. La invención permite ejecutar varios servicios sobre la base de una entrada de usuario, ya que la lógica del servicio puede llamar a varios servicios en respuesta a una entrada del usuario. En otras palabras, una entrada de usuario ya no está ligada a exactamente un servicio, sino que se pueden ofrecer una combinación de servicios en respuesta a una entrada de lenguaje natural del usuario. La combinación de servicios proporcionados puede ser una arbitraria, dependiendo del contenido de la entrada del lenguaje natural. Además, la lógica del servicio puede procesar los resultados obtenidos a partir de servicios separados en una diversidad de formas. Esto también posibilita un modo eficaz de generar nuevos servicios adaptados de los servicios existentes.

5 Los atributos del tipo de los elementos de XML también se pueden mapear directamente a llamadas de servicio, sin
 10 usar las clases descritas anteriormente. Una implementación como esta, en la cual el analizador produce llamadas
 del servicio que corresponden a los atributos de los tipos encontrados por el analizador, es factible en la práctica
 solo para funcionalidades simples. Si las funcionalidades son más complicadas, requiriendo una interacción del
 usuario, por ejemplo, es preferible la implementación descrita anteriormente que usa las clases. Esto se debe al
 hecho de que la implementación de la interacción del usuario mencionada anteriormente requiere el manejo del
 estado del usuario lo que no es razonable de implementar sin las clases descritas anteriormente. Además, la
 provisión de combinaciones de servicio por el uso de respuestas del servicio como información de entrada de otros
 servicios, así como la monitorización del estado de la sesión del usuario se puede implementar de una forma más
 general y controlada si se usan las clases en el modo descrito anteriormente.

15 Los servicios se pueden integrar en el sistema de muchas formas. En general, un servicio que pertenece a la
 estructura del servicio puede ser cualquier módulo que proporciona contenido, tal como una aplicación de servicio o
 una base de datos. Las aplicaciones del servicio se puede usar bien usando una interfaz de programación del
 servicio con cualquier lenguaje de programación compatible o usando cualquier interfaz de usuario del servicio
 descrita por un lenguaje de descripción generalizado, tal como el WILD (Lenguaje de Descripción de la Interfaz de
 Web) que es una aplicación del XML. De este modo, también se pueden usar las solicitudes de servicio usando una
 descripción XML de la interfaz de usuario. Se pueden implementar diferentes clases de adaptadores para la
 integración del servicio.

20 Aunque la invención se ha descrito anteriormente con referencia a los ejemplos mostrados en los dibujos adjuntos,
 es obvio que la invención no está limitada a estos, sino que la pueden modificar los expertos en la materia sin
 apartarse del alcance de la invención, que se define por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento para proporcionar una amplia diversidad de servicios diferentes y combinaciones arbitrarias de servicios en una red de comunicaciones con al menos un servidor de aplicación (S1, S2) que almacenan las aplicaciones del servicio, asignándose a cada una de las aplicaciones de servicio una operación por la cual se
5 arranca la aplicación del servicio, y terminales de usuario (UT1, UT2) para suministrar peticiones de servicio en lenguaje natural, incluyendo el procedimiento las etapas de:
 - dar una petición de servicio del usuario en lenguaje natural (301), **caracterizado por** las etapas adicionales de:
 - 10 - interpretar (303) la petición del servicio del usuario y generar una descripción del servicio de formato genérico sobre la base de dicha interpretación, estando la descripción del servicio en un lenguaje de marcación generalizado, incluyendo la descripción del servicio, los nombres de las funcionalidades que se necesitan para cumplir la petición de servicio del usuario y los parámetros que requieran dichas funcionalidades.
 - 15 - recibir (601) la descripción del servicio de formato genérico,
 - dividir (611) la descripción del servicio de formato genérico en uno o más elementos, incluyendo un elemento un nombre de dicha funcionalidad como un tipo y los valores de los parámetros asociados,
 - analizar (306) los elementos y examinar (612) el tipo de los elementos y mapear (613) dicho tipo a una entidad, indicando las entidades las llamadas de servicio a realizar y los parámetros relativos a las llamadas de servicio para implementar las funcionalidades especificadas en la descripción de servicio de formato
20 genérico, y
 - usar dichas entidades para utilizar dichas aplicaciones de servicio (309).
2. Un procedimiento de acuerdo con la reivindicación 1, **caracterizado porque** la etapa de análisis incluye además:
 - mapear (613) dichos elementos en clases que implementan dicha utilización de las aplicaciones de servicio,
25 y
 - crear objetos desde dichas clases.
3. Un procedimiento de acuerdo con la reivindicación 2, **caracterizado porque** la etapa de realización incluye la ejecución de dichos objetos con dichos valores de parámetros.
4. Un procedimiento de acuerdo con la reivindicación 1, **caracterizado porque** la etapa de generación incluye generar de una descripción de servicio de formato XML.
- 30 5. Un procedimiento de acuerdo con la reivindicación 1, **caracterizado porque** la etapa de interpretación incluye el uso de una red de agentes de software.
6. Un procedimiento de acuerdo con la reivindicación 1, **caracterizado porque** la etapa de análisis incluye analizar la descripción del servicio en los elementos XML.
- 35 7. Un procedimiento de acuerdo con la reivindicación 2, **caracterizado porque** la etapa de creación incluye la ejemplificación de clases que implementan dicha utilización de aplicaciones de servicio.
8. Un procedimiento de acuerdo con la reivindicación 3, **caracterizado porque** la etapa de ejecución incluye llamar al procedimiento de ejecución de cada uno de los objetos identificados en la etapa de análisis.
9. Un procedimiento de acuerdo con la reivindicación 1, **caracterizado por** la inclusión adicional de la etapa de inicialización de una interacción con el transmisor de la petición de servicio.
- 40 10. Un sistema de suministro de servicio, para el suministro de una amplia diversidad de servicios diferentes y combinaciones arbitrarias de servicios a los usuarios de una red de comunicaciones, comprendiendo el sistema:
 - al menos un servidor de aplicaciones (S1, S2) que almacena aplicaciones de servicio, asignándose a cada una de las aplicaciones de servicio una operación por la que se arranca la aplicación de servicio, y
 - 45 - terminales de usuario para suministrar peticiones de servicio en un lenguaje natural **caracterizado porque** el sistema comprende además:
 - un medio de interpretación (NLI) para interpretar una de dichas peticiones de servicio del usuario en lenguaje natural y generar una descripción del servicio de formato genérico sobre la base de dicha interpretación, siendo la descripción del servicio en lenguaje de marcación generalizado, incluyendo la descripción de servicio los nombres de las funcionalidades que se necesitan para cumplir la petición de servicio del usuario y los parámetros que requieren dichas funcionalidades,
 - 50 - medios de análisis (PS), que responden al medio de interpretación, incluyendo medios para recibir la descripción del servicio de formato genérico a partir del medio de interpretación, dividiendo la descripción del servicio de formato genérico en uno o más elementos, incluyendo un elemento uno de dichos nombres de una funcionalidad como un tipo y los valores de los parámetros asociados, y para cada uno de los elementos analizar el elemento y examinar el tipo de elemento y mapear dicho tipo a una entidad, indicando las
55

entidades las llamadas de servicio a realizar y los parámetros relativos a las llamadas de servicio para la implementación de las funcionalidades especificadas en la descripción del servicio de formato genérico, y - medios de ejecución (SL), que responden al medio de análisis, usando dichas entidades para utilizar dichas aplicaciones de servicio.

- 5 11. Un sistema de acuerdo con la reivindicación 10 **caracterizado porque** el medio de interpretación incluye una red de agentes de software.
12. Un sistema de acuerdo con la reivindicación 11 **caracterizado porque** la red está adaptada para generar una descripción de servicio de formato XML en respuesta a una petición de servicio.
- 10 13. Un sistema de acuerdo con la reivindicación 10 **caracterizado porque** los medios de análisis incluyen un medio adaptado para mapear dichos elementos en clases que implementan dicha utilización de las aplicaciones de servicio, y crean objetos a partir de dichas clases.
14. Un sistema de acuerdo con la reivindicación 13 **caracterizado porque** los medios de ejecución están adaptados para llamar al procedimiento de ejecución de cada una de las clases de objetos creadas por los medios de análisis.
- 15 15. Un sistema de acuerdo con la reivindicación 10 **caracterizado porque** los medios de ejecución incluyen medios para la inicialización e interacción con los terminales de usuario (UT1, UT2).

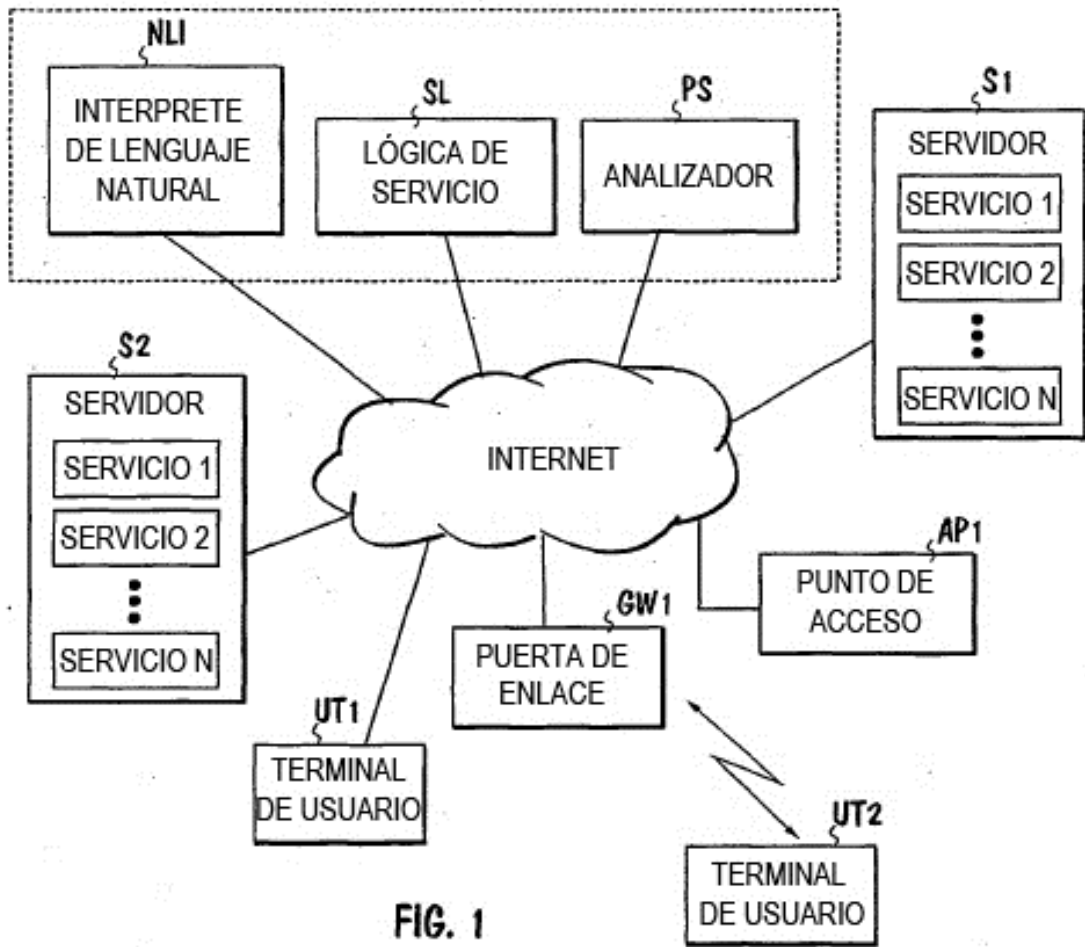


FIG. 1

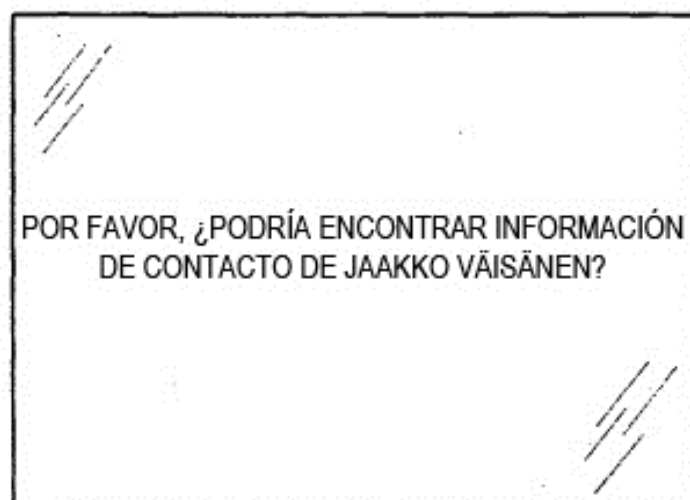


FIG. 2

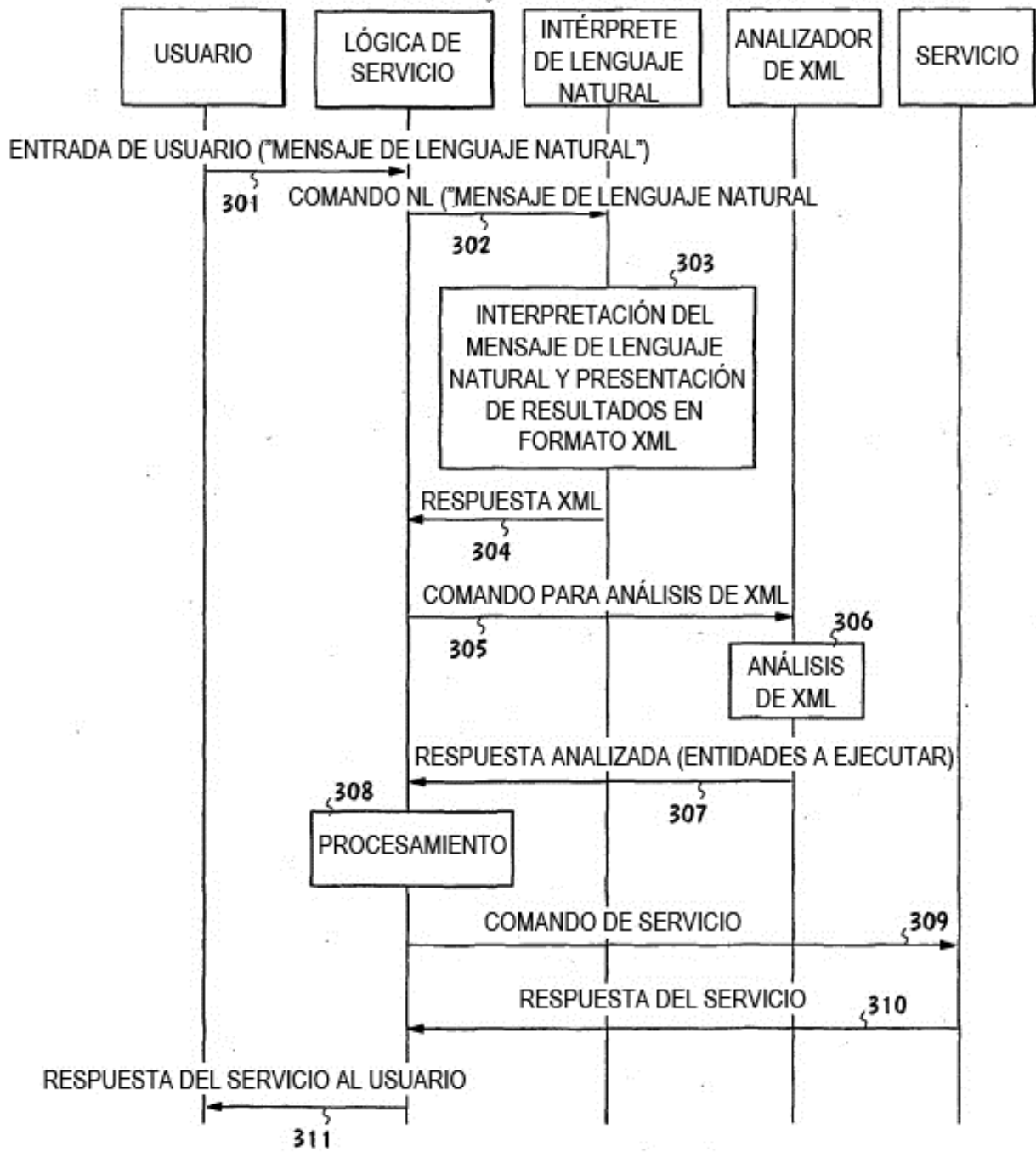


FIG. 3

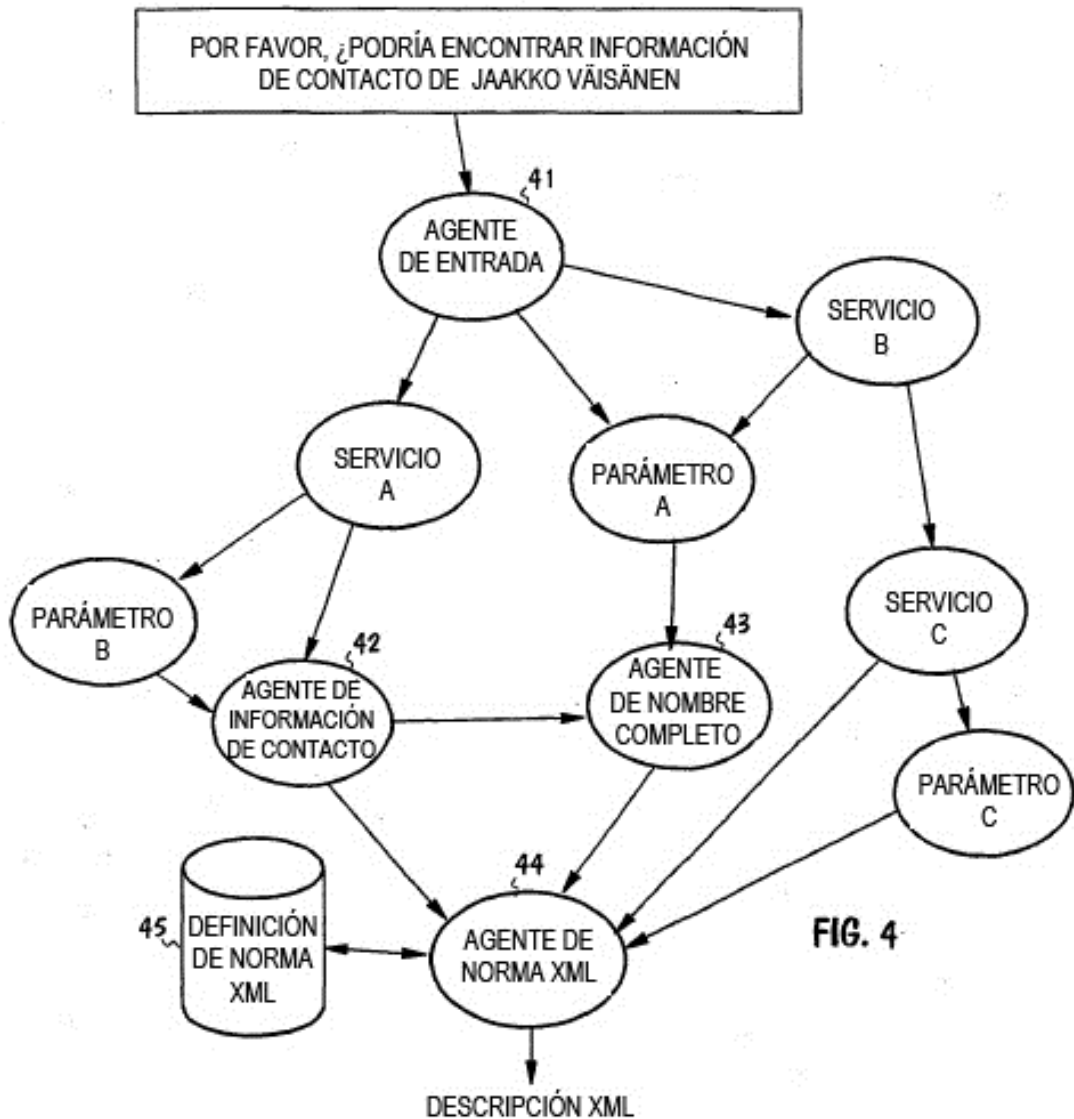


FIG. 4

```

<?versión XML = "1.0" codificación = "UTF-8"?>
<¡DOCTIPO comandorespuesta SISTEMA "comandorespuesta.dfd">
<comandorespuesta>
<comando tipo = "buscar_contacto" indice = "1">
<parámetro tipo = "NombreCompletoContacto"> Jaakko Väisänen
</parámetro>
</comando>
</comandorespuesta>
    
```

FIG. 5

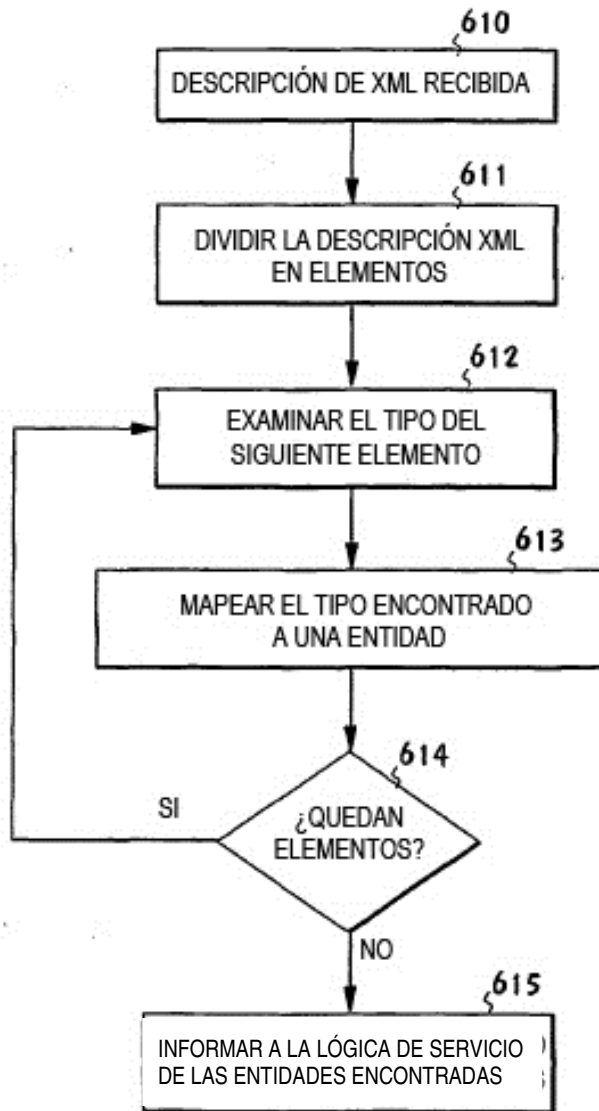


FIG. 6

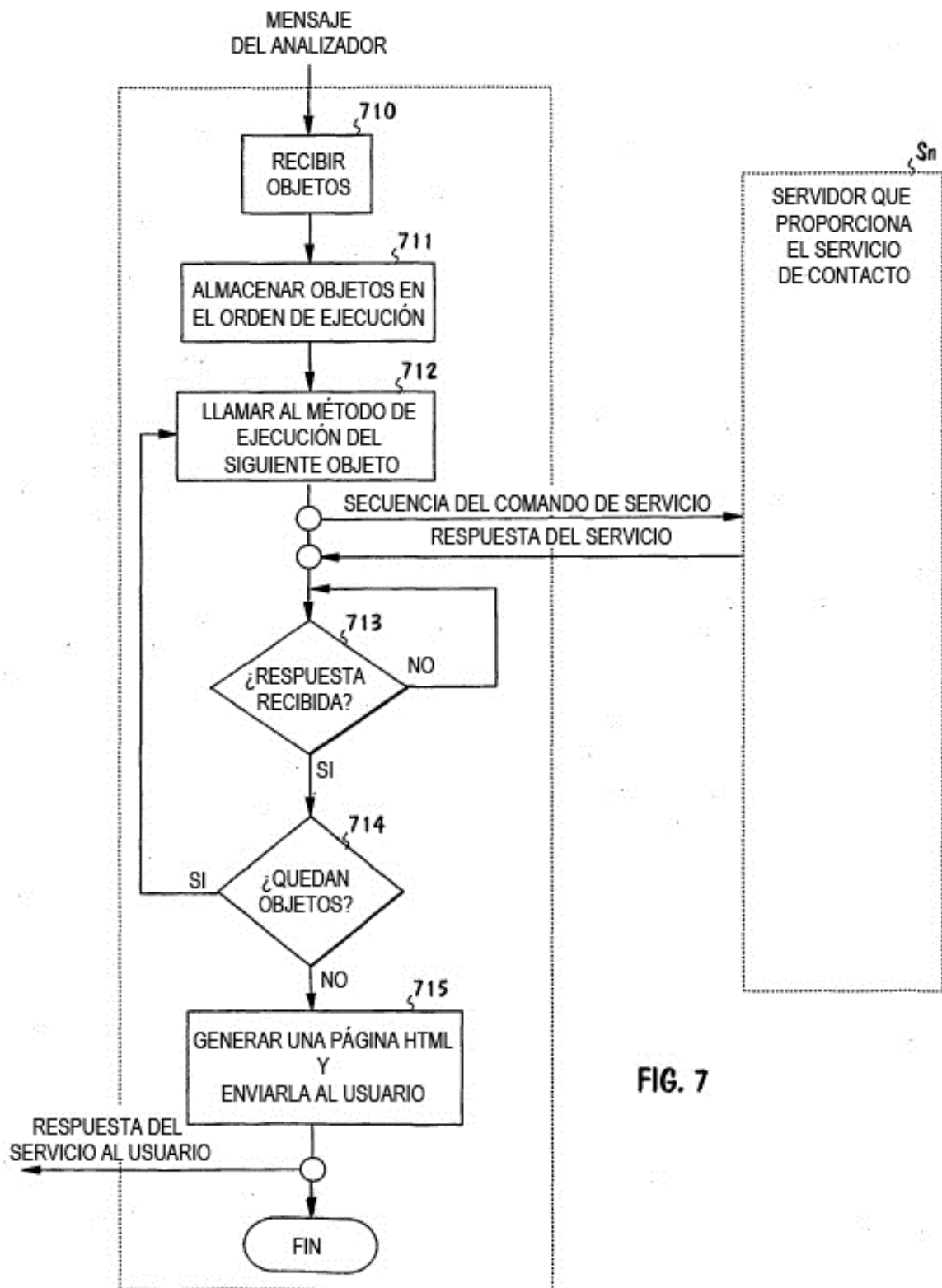


FIG. 7