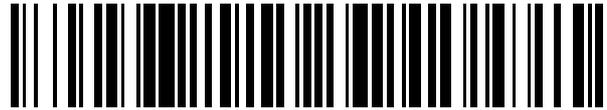


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 458 303**

51 Int. Cl.:

G06F 17/30 (2006.01)

H04L 29/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **27.10.2006 E 06805092 (1)**

97 Fecha y número de publicación de la concesión europea: **29.01.2014 EP 1845688**

54 Título: **Método, sistema, terminal de cliente y servidor para realizar la sincronización de datos**

30 Prioridad:

27.10.2005 CN 200510116802
14.08.2006 CN 200610109591

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
30.04.2014

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.0%)
Huawei Administration Building Bantian
Longgang District
Shenzhen, Guangdong 518129, CN

72 Inventor/es:

TIAN, LINYI

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 458 303 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

Método, sistema, terminal de cliente y servidor para realizar la sincronización de datos

5 CAMPO DE LA INVENCION

La presente invención se refiere al campo de las especificaciones de sincronización de datos del Lenguaje de Marcas de Sincronización (SyncML) que se definen por la denominada Open Mobile Alliance (OMA) y en particular, a un método, sistema, terminal de cliente y servidor para poner en práctica la sincronización de datos basada en el protocolo SyncML.

ANTECEDENTES DE LA INVENCION

Para formular las especificaciones estándar para la sincronización de datos de información personal y datos de empresas entre una pluralidad de plataformas y redes, la alianza OMA propone especificaciones de sincronización de datos SyncML. El objetivo del desarrollo de SyncML es poner en práctica el trabajo colaborativo entre usuarios de terminales, desarrolladores de dispositivos, desarrolladores de componentes básicos, proveedores de datos, todo ellos con el fin de posibilitar el acceso a datos de cualquier red en cualquier lugar y en cualquier momento utilizando cualquier dispositivo de terminal. El escenario operativo típico es la sincronización de datos entre un dispositivo móvil/un servidor de aplicación y un servidor de red. Además, el lenguaje SyncML puede utilizarse también en la sincronización de datos, entre homólogos, tal como la sincronización de datos entre dos ordenadores personales (PC).

La Figura 1 es un diagrama esquemático que ilustra la sincronización de datos sobre la base de las especificaciones de SyncML. Después de haber negociado parámetros en una fase de inicialización de sincronización de datos, un cliente y un servidor envían los datos que han intercambiado para garantizar que los datos estén sincronizados entre las dos partes.

Un cliente de sincronización de datos (DS Client) suele referirse a un terminal inteligente, tal como software de PC, teléfono móvil o Asistente Digital Personal (PDA), etc. Una base de datos de clientes está configurada en el dispositivo para memorizar los datos de usuarios, incluyendo: una agenda de direcciones, calendario, bloc de notas, mensajes cortos, correos electrónicos, etc. Los formatos de todos estos datos se definen en especificaciones estándar y el DS Client es capaz de convertir los datos en un formato estándar y de enviar los datos convertidos a un servidor DS Server, que procesa los datos y los memoriza en su base de datos.

El servidor DS Server puede recibir un mensaje de sincronización que contenga órdenes de sincronización desde el DS Client y enviar un mensaje de sincronización de nuevo al DS Client. El servidor DS Server puede ser un servidor de red o un ordenador PC. Una base de datos de servidor está configurada en el servidor DS Server para memorizar los datos del DS Server.

Los identificadores de datos se memorizan en el DS Client y en el DS Server. El DS Client utiliza un Identificador Único Local (LUID) como el identificador de datos mientras que el DS Server utiliza un Identificador Único Global (GUID) como el identificador de datos.

La Figura 2 es un diagrama esquemático que ilustra el almacenamiento de datos de un DS Client y de un DS Server. Según se ilustra en la Figura 2, solamente la relación de correspondencia entre varios identificadores LUIDs y datos necesita mantenerse en el DS Client, pero en el DS Server necesita mantenerse no solamente la relación de correspondencia entre varios identificadores GUIDs y datos, sino también la relación de correspondencia entre varios identificadores GUIDs y LUIDs necesita mantenerse. Existen múltiples tipos de sincronización de datos, según se ilustra en la tabla 1.

Tipo de sincronización	Información de descripción
Sincronización bidireccional	Un tipo de sincronización normal en donde el cliente y el servidor intercambian información sobre datos modificados en estos dispositivos. El cliente envía primero las modificaciones.
Sincronización lenta	Una forma de sincronización bidireccional en la que se comparan todos los elementos de datos, entre sí sobre una base de campo por campo. En la práctica esto significa que el cliente envía todos sus datos desde una base de datos al servidor y el servidor realiza el análisis de sincronización (campo por campo) para estos datos y los datos en el servidor.
Sincronización unidireccional desde el cliente solamente	Un tipo de sincronización en la que el cliente envía sus modificaciones al servidor pero el servidor no envía sus modificaciones de nuevo al Cliente.

Sincronización de recuperación desde cliente solamente	Un tipo de sincronización en la que el cliente envía todos sus datos desde una base de datos al servidor (esto es, exportaciones). Está previsto que el servidor sustituya todos los datos en la base de datos objetivo con los datos enviados por el cliente.
Sincronización unidireccional desde servidor solamente	Un tipo de sincronización en donde el cliente obtiene todas las modificaciones desde el servidor, pero el cliente no envía sus modificaciones al servidor.
Sincronización de recuperación desde servidor solamente	Un tipo de sincronización en el que el servidor envía todos sus datos desde una base de datos al cliente. Está previsto que el cliente sustituya todos los datos en la base de datos objetivo con los datos enviados por el servidor.
Alerta de sincronización de servidor	Un tipo de alerta de sincronización, que proporciona los medios para que un servidor proporcione una alerta al cliente para realizar la sincronización. Cuando el servidor alerta al cliente, también le comunica al cliente qué tipo de sincronización iniciar.

Tabla 1

Además, el procedimiento de sincronización de datos definido en las especificaciones de SyncML suele incluir tres fases:

1. Una fase de inicialización de sincronización, en donde, es una tarea principal poner en práctica la sincronización de la autenticación, negociación de la base de datos de sincronización, la negociación de capacidades de sincronización (incluyendo: qué formatos de datos y tipos de sincronización son soportados por el cliente y/o servidor, etc.) y dichos procedimientos de negociación pueden tener que realizarse durante varias veces.
2. Una fase de sincronización, que incluye principalmente: un lado del DS Client y del DS Server envía los datos que han cambiado al otro lado de los dos mediante una orden operativa en conformidad con la modificación del estado de datos y el otro lado realiza la orden operativa (p.e., una orden operativa de actualización, supresión o adición), con los datos que se han cambiado para actualizar sus propios datos con el fin de conseguir la finalidad de la sincronización de datos.
3. Una fase de realización de sincronización, que incluye principalmente: el DS Client y el DS Server que confirman la realización de la sincronización mutua.

En la técnica anterior, el modo de memorización de Folder (Carpeta) y File (Archivo) ha sido definido para los datos, que simula la estructura arborescente basada en las carpetas y ficheros en el ordenador PC. Para los datos con relación de jerarquía, de forma lógica o física, los datos pueden presentarse como una estructura arborescente que consiste en al menos un nodo, pudiendo ser cada uno de dichos nodos un nodo de carpeta (también referido como un elemento de carpeta) o un nodo de elemento de datos (también referido como un elemento de datos). No obstante, no es posible sincronizar un nodo específico con su contenido o un sub-árbol como se requiere en la técnica anterior. Además, el método para sincronizar la agenda de direcciones por grupos se pone en práctica utilizando la técnica de filtros basada en el campo de Grupo dentro de la tarjeta vCard, cuyo defecto es que el protocolo de sincronización está fuertemente acoplado con formatos de datos específicos, lo que no es general para todos los formatos de datos, por lo que un nodo específico, con su contenido o un sub-árbol en la estructura arborescente, puede no estar sincronizado cuando se requiera.

Sin embargo, actualmente, existen muchos datos necesarios para sincronizarse que se memorizan como un hecho de la estructura arborescente con relación jerárquica desde el punto de vista lógico o físico. A modo de ejemplo, algunos datos están organizados por carpetas en una vista de estructura arborescente que existe, de forma lógica o física, en el teléfono móvil de un usuario, tal como una agenda de direcciones, mensajes cortos clasificados por categorías y correos electrónicos organizados por buzones, etc. Además, un calendario o correo electrónico, con documentos adjuntos, podría considerarse como organizado también en una manera jerárquica. En conformidad con la técnica anterior, el usuario solamente puede sincronizar la base de datos completa en lugar de una parte de la base de datos. Tomando, a modo de ejemplo, mensajes cortos, el usuario solamente puede sincronizar la base de datos completa del mensaje corto pero no puede sincronizar solamente una categoría del mensaje corto denominada "bless" y abandonar otra categoría denominada "joke" no sincronizada en este momento. Con respecto a los documentos adjuntos memorizados fuera del calendario/correo electrónico, el documento adjunto no se puede sincronizar y la relación de jerarquía del documento adjunto y del calendario o correo electrónico no se puede describir en la técnica anterior. Asimismo, no puede ponerse en práctica que un solo elemento de datos exista en dos categorías. A modo de ejemplo, Zhang San pertenece a los grupos de "grupo de colegas" y "grupo de amigos" en la agenda de direcciones al mismo tiempo.

En resumen, las técnicas de sincronización de datos existentes no pueden satisfacer las demandas reales y en particular, no pueden soportar la noción de un solo elemento de datos en una manera jerárquica o la sincronización de datos de cualquier nivel de nodos en la estructura jerárquica.

El documento WO 03/083684 resuelve un problema existente en la técnica anterior para SyncML al permitir solamente la sincronización con respecto a cambios en unidades de datos y da a conocer un método para sincronizar la forma en que los datos se organizan en una memoria de datos.

- 5 El documento EP 1267283 resuelve el problema del método de selección aproximada descrito en WO 03/083684 extendiendo el protocolo de SyncML durante la fase de inicialización de sincronización; sin embargo, el algoritmo de selección de nodos está basado en metadatos adicionales y el cálculo de la relación de importancia.

SUMARIO DE LA INVENCION

10 Formas de realización de la invención dan a conocer un método para sincronizar una pluralidad de dispositivos que incluye:

15 la obtención, por un primer dispositivo, de la dirección de un nodo informacional a sincronizarse entre una pluralidad de nodos informacionales dispuestos en una estructura arborescente jerárquica en un segundo dispositivo, en donde dicha dirección comprende una posición jerárquica que se utiliza para localizar dicho nodo informacional entre la estructura arborescente y el nodo informacional es un elemento de datos o una carpeta en una base de datos;

20 el envío, por el primer dispositivo, de una orden de sincronización durante una fase de inicialización de sincronización, al segundo dispositivo para diseñar el nodo informacional a sincronizarse, en donde la orden de sincronización comprende la dirección del nodo informacional,

25 la utilización, por el primer dispositivo, de la dirección durante una fase de sincronización para sincronizar el nodo informacional.

Las formas de realización de la invención dan a conocer, además, un aparato para la sincronización, que incluye:

30 un primer módulo, adaptado para obtener la dirección de un nodo informacional a sincronizarse entre una pluralidad de nodos informacionales dispuestos en una estructura arborescente jerárquica en un segundo dispositivo y para proporcionar la dirección a un segundo módulo, en donde dicha dirección comprende una posición jerárquica que se utiliza para localizar dicho nodo informacional entre la estructura arborescente;

35 el segundo módulo, adaptado para enviar una orden de sincronización durante una fase de inicialización de sincronización al segundo dispositivo para designar el nodo informacional a sincronizarse y utilizar la dirección obtenida desde el primer módulo para sincronizar el nodo informacional durante una fase de sincronización, en donde la orden de sincronización comprende la dirección del nodo informacional, siendo el nodo informacional un elemento de datos o una carpeta en una base de datos.

40 Las formas de realización de la invención dan a conocer otro aparato para la sincronización, que incluye:

45 un primer módulo, adaptado para recibir una orden de sincronización desde un primer dispositivo, durante una fase de inicialización de sincronización, para designar un nodo informacional a sincronizarse entre una pluralidad de nodos informacionales dispuestos en una estructura arborescente jerárquica en un segundo módulo, en donde la orden de sincronización comprende la dirección del nodo informacional, comprendiendo dicha dirección una posición jerárquica que se utiliza para localizar dicho nodo informacional entre la estructura arborescente y el nodo informacional es un elemento de datos o una carpeta en una base de datos;

el segundo módulo, adaptado para sincronizar el nodo informacional durante una fase de sincronización.

50 Por lo tanto, las formas de realización de la invención se adoptan para poner en práctica, de forma flexible, la sincronización de datos con respecto a un determinado nivel, sin transmitir los datos de la base de datos completa entre el cliente y el servidor durante la sincronización de datos, con el fin de elevar la eficiencia de la sincronización de datos y ahorrar recursos del sistema así como para satisfacer las demandas de usuarios reales.

55 BREVE DESCRIPCION DE LOS DIBUJOS

La Figura 1 es un diagrama esquemático de la puesta en práctica de sincronización de datos.

60 La Figura 2 es un diagrama esquemático que ilustra la memorización de datos de un cliente y de un servidor.

La Figura 3a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario en conformidad con una primera forma de realización de esta invención.

65 La Figura 3b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una primera forma de realización de esta invención.

La Figura 4a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario según una segunda forma de realización de esta invención.

5 La Figura 4b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una segunda forma de realización de esta invención.

La Figura 5a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario según una tercera forma de realización de esta invención.

10 La Figura 5b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una tercera forma de realización de esta invención.

La Figura 6a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario según una cuarta forma de realización de esta invención.

15 La Figura 6b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una cuarta forma de realización de esta invención.

20 La Figura 7a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario según una quinta forma de realización de esta invención.

La Figura 7b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una quinta forma de realización de esta invención.

25 La Figura 8a es un diagrama esquemático que ilustra la estructura de datos definida por un usuario según una sexta forma de realización de esta invención.

La Figura 8b es un diagrama esquemático que ilustra la memorización de datos de un cliente según una sexta forma de realización de esta invención.

30 La Figura 9 es un diagrama esquemático que ilustra la arquitectura de sistema para la sincronización de datos según las formas de realización de esta invención.

DESCRIPCIÓN DETALLADA DE LAS FORMAS DE REALIZACIÓN

35 Esta invención se describirá, a continuación, en detalle, haciendo referencia a los dibujos adjuntos y a las formas de realización específicas.

40 Las formas de realización de esta invención dan a conocer un método para poner en práctica la sincronización de datos, en donde un lado de un cliente y un servidor envían una primera orden de sincronización al otro lado. Antes de que se envíe la primera orden de sincronización, el cliente y el servidor determinan la dirección del nodo a sincronizarse. Después de recibir la primera orden de sincronización, el lado receptor de la primera orden de sincronización realiza la sincronización de datos del nodo determinado a sincronizarse en función de la primera orden de sincronización recibida. El nodo a sincronizarse puede ser cualquier nodo de una estructura arborescente jerárquica, p.e., una base de datos completa, una carpeta lógica o física o un elemento de datos.

45 La sincronización de datos puede dividirse en tres fases en conformidad con el protocolo de SyncML, que incluye: una fase de inicialización de sincronización, una fase de sincronización y una fase de realización de sincronización. El proceso de la fase de realización de sincronización, en las formas de realización de esta invención, es el mismo que el de la técnica anterior, por lo que, a continuación, se describe solamente la fase de inicialización de sincronización y la fase de sincronización.

50 En la fase de inicialización de sincronización, la dirección del nodo a sincronizarse debe designarse mediante la negociación del cliente y del servidor mediante el envío de una segunda orden de sincronización, que puede ser una dirección de base de datos, una identidad de carpeta lógica o física o una identidad de elemento de datos. Un lado del cliente y del servidor envía una segunda orden de sincronización que contiene la dirección del nodo a sincronizarse al otro lado que luego determina el nodo actualmente a sincronizarse en función de la dirección del nodo a sincronizarse contenida en la segunda orden de sincronización. El nodo a sincronizarse puede incluir uno o más nodos. Si el nodo a sincronizarse incluye más de un nodo, un lado del cliente y el servidor envía una segunda orden de sincronización para todos los nodos al otro lado, que contiene las direcciones de todos los nodos y el otro lado determina todos los nodos incluidos en el nodo a sincronizarse actual en función de las direcciones de todos los nodos contenidos en la segunda orden de sincronización. En un lado del cliente, el servidor envía una segunda orden de sincronización para cada nodo al otro lado que contiene la dirección del nodo y el otro lado del cliente determina todos los nodos incluidos en los nodos actualmente a sincronizarse, uno por uno, en función de cada dirección de nodo contenida en cada segunda orden de sincronización. Como alternativa, la dirección del nodo a sincronizarse puede configurarse en el cliente y el servidor por anticipado, de modo que no exista ninguna necesidad

de negociar el nodo a sincronizarse por intermedio de la segunda orden de sincronización en la fase de inicialización de sincronización. El nodo a sincronizarse, mencionado en las formas de realización de esta invención, puede incluir la base de datos, la carpeta, el elemento de datos o la combinación de al menos una carpeta y al menos un elemento de datos. La puesta en práctica de las formas de realización de esta invención se describe, a continuación, con respecto al nodo a sincronizarse incluyendo al menos una carpeta para aclarar el principio de las formas de realización de esta invención, mientras que otra posible combinación del nodo a sincronizarse no será especificada, estando, sin embargo, cubierta por el alcance de protección de esta invención.

Más concretamente, existen dos clases de modos para realizar la transmisión de la dirección del nodo a sincronizarse en la segunda orden de sincronización como sigue (en caso de que el nodo esté sincronizado incluyendo al menos una carpeta, la dirección del nodo a sincronizarse es la dirección de la carpeta a sincronizarse).

Modo 1: Un elemento para indicar una base de datos objetivo (también referida como un elemento para contener la dirección de los datos a sincronizarse) en el protocolo existente se extiende para ser capaz de indicar la carpeta (realmente puede indicar cualquier nodo dentro de la estructura arborescente) a sincronizarse, que originalmente, como un elemento para indicar una dirección a nivel de base de datos, se extiende como un elemento capaz de indicar una dirección de nodo de cualquier nivel. El proceso de puesta en práctica detallado puede incluir: definición previa del formato del Identificador de Recursos Uniformes (URI), esto es, la designación previa del URI de cuyo nivel identifica una dirección de nivel de base de datos y el URI de cuyo nivel identifica una dirección de nivel de nodo e indica el nodo a sincronizarse por el URI preestablecido. A modo de ejemplo, la designación del URI de “/ficheros” es la dirección del nivel de base de datos y el URI de “/files/folder1 ID” es la dirección de la carpeta, denominada folder1, en la base de datos, en donde “files” es un nombre de base de datos y “folder1 ID” es una identidad de carpeta. El Identificador Único (UID) de una base de datos o de una carpeta o de un elemento de datos puede utilizarse en el URI para designar, con mayor exactitud, la dirección del nodo a sincronizarse. A modo de ejemplo, “/4980” se designa para ser la dirección de la base de datos a sincronizarse, “/4980/4560” se designa para ser la dirección de la subcarpeta identificada por UID 4560 en la base de datos nº 4980 identificada por UID 4980. Si existen más niveles en la base de datos, se adoptará el URI con una estructura multinivel, tal como “/4560/4980/556”, etc. EL método de indicación de URI, utilizando la mezcla del número del identificador UID junto con el nombre de la carpeta, puede adoptarse también, tal como “/files/4890”. Por supuesto, si existen múltiples carpetas a sincronizarse, pueden proporcionarse múltiples identificadores URIs para indicar estas carpetas. Además, en una estructura jerárquica, un nodo de carpeta puede tener al menos un sub-nodo y/o al menos un nodo de elemento de datos, por lo que una dirección del nodo a sincronizarse puede consistir en una identidad de carpeta y una identidad de elemento de datos, p.e., la dirección de nodo “/Macy/03/05”, en donde “03” es la identidad del elemento de datos perteneciente a la base de datos “Macy” y “05” es la identidad del sub-elemento de datos (p.e., el documento adjunto) del nodo principal del elemento de datos “03”.

Modo 2: Un elemento para indicar una condición de filtro en el protocolo existente se extiende para indicar la carpeta (realmente, puede indicar cualquier nodo dentro de la estructura arborescente) a sincronizarse. La puesta en práctica específica incluye la extensión de la condición de filtro existente (p.e., orden de filtro Filter) para indicar la carpeta que se va a sincronizar. Por supuesto, si existe más de una carpeta a sincronizarse, estas carpetas pueden ser indicadas en la condición de filtro al mismo tiempo.

Estos dos modos para indicar la carpeta a sincronizarse se describen, a continuación, en detalle.

I. El Modo 1 es un esquema para negociar la carpeta a sincronizarse extendiendo el elemento para indicar la base de datos objetivo del protocolo existente, en donde la dirección de la carpeta a sincronizarse, contenida en el elemento para indicar la base de datos objetivo, se suele referir como un URI.

En las formas de realización de esta invención, mientras se negocia la carpeta a sincronizarse enviando una segunda orden de sincronización, un cliente y un servidor pueden negociar, además, el parámetro de negociación del tipo de sincronización. Cuando el cliente y el servidor negocian la carpeta a sincronizarse enviando la segunda orden de sincronización, el parámetro de negociación del tipo de sincronización puede contenerse, además, en la segunda orden de sincronización, según la que puede determinarse el tipo de sincronización actual. La segunda orden de sincronización puede ponerse en práctica utilizando un elemento de Alerta del protocolo de Lenguaje de Marcas de Sincronización (SyncML), en donde el elemento para contener la dirección de la carpeta, a sincronizarse, puede ser un sub-elemento del elemento de Alerta (p.e. un elemento de datos Item) y el elemento para contener el parámetro de negociación del tipo de sincronización puede ser un sub-elemento del elemento de Alerta (p.e. un elemento de datos Item). La segunda orden de sincronización puesta en práctica utilizando un elemento de Alerta puede denominarse también una orden de Sync Alert. Otros sub-elementos en el elemento de Alerta funcionan como la segunda orden de sincronización, tal como un elemento de Código, etc. puede utilizarse también para contener el parámetro de negociación del tipo de sincronización y la dirección de la carpeta a sincronizarse y un atributo del elemento de Alerta puede utilizarse también para contener el parámetro de negociación del tipo de sincronización y la dirección de la carpeta a sincronizarse. Las estructuras de órdenes en estos casos no se indicarán aquí una por una, pero deben cubrirse dentro del alcance de protección de esta invención. Y en las

especificaciones futuras, el nombre y la estructura de estos elementos, sub-elementos o atributos pueden cambiar, pero estos cambios no pueden considerarse como la limitación de la presente invención.

5 La orden de Alerta puede extenderse, además, para poner en práctica la sincronización de múltiples carpetas, de modo que pueda utilizarse para designar una o más carpetas a sincronizarse. El sistema para poner en práctica la función se proporciona a continuación.

10 En la fase de inicialización de sincronización, suponiendo que la negociación de la carpeta a sincronizarse se inicia por un cliente para un servidor, el proceso de negociación es como sigue.

15 El cliente envía una orden de Alerta que contiene la dirección de las carpetas a sincronizarse para el servidor, y la dirección puede ser el URI de la carpeta a sincronizarse. A continuación, el servidor puede reenviar una respuesta al cliente. Una realización, a modo de ejemplo, de la orden de Alerta se proporciona a continuación. A modo de ejemplo, la carpeta a sincronizarse que el cliente designa se denomina como folder1. El cliente envía una orden de Alerta al servidor, que contiene el tipo de sincronización y la dirección de la carpeta a sincronizarse (p.e., URI). Si la estructura de carpeta del cliente es la misma que la del servidor (si el nodo a sincronizarse solamente incluye al menos una carpeta, la estructura de la carpeta se refiere a la estructura jerárquica de datos), el cliente determina directamente la dirección local de la carpeta a sincronizarse como la dirección de la carpeta a sincronizarse en el servidor en conformidad con su estructura de carpeta local. Si la estructura de carpeta del cliente es diferente de la que tiene el servidor, el cliente puede necesitar determinar la dirección de la carpeta a sincronizarse en el servidor, en donde existen varios métodos de determinación como sigue: 1. El cliente adquiere primero la estructura de carpeta del servidor desde el servidor y determina la dirección de la carpeta a sincronizarse en el servidor en conformidad con la estructura de carpeta adquirida; 2. El servidor envía la dirección (p.e., UID o URI) de la carpeta a sincronizarse al cliente; 3. El cliente designa directamente la dirección de la carpeta a sincronizarse y si la dirección designada no existe en el servidor, el servidor crea la carpeta a sincronizarse en la dirección designada por el cliente; 4. El cliente determina la dirección de la carpeta a sincronizarse en función de la entrada de información por el usuario, esto es, el usuario designa la dirección de la carpeta a sincronizarse; 5. El cliente y el servidor establecen por anticipado la dirección de la carpeta a sincronizarse. Aún cuando la estructura de la carpeta del cliente sea la misma que la del servidor, el cliente puede utilizar también los métodos anteriores para determinar la carpeta a sincronizarse. A modo de ejemplo, el cliente y el servidor establecen un acuerdo de que si ha de realizarse una sincronización de reserva, debe utilizarse la carpeta denominada "backup" en la base de datos del servidor. La estructura de carpeta anterior puede memorizarse en el servidor después de su creación o no es memorizada en el servidor, sino creada en tiempo real para la base de datos designada por el servidor si el cliente así lo requiere. Si el servidor envía la orden de Alerta para iniciar el procedimiento de sincronización, la dirección de la carpeta a sincronizarse puede incluir: la dirección de la carpeta a sincronizarse en el cliente (esto es, la dirección origen) y la del servidor (esto es, la dirección de destino). Las dos direcciones pueden ser las mismas o diferentes. Con respecto al cliente como un origen de sincronización, la dirección local de la carpeta a sincronizarse (esto es, la dirección origen) puede obtenerse por múltiples medios, p.e., configuración previa de la dirección origen o designación de la dirección origen por el usuario o el envío de la dirección origen al cliente por el servidor, etc. Sin embargo, la dirección de la carpeta a sincronizarse en el servidor (esto es, la dirección de destino) ha de indicarse al servidor por el cliente cuando se inicie el procedimiento de sincronización. Es el método para indicar la dirección de destino al servidor por el cliente funcionando como el origen de la sincronización que aquí se describe mientras el método para determinar la dirección origen por el cliente no se examinará en esta descripción.

45 Puede realizarse, mediante una sesión de sincronización independiente, que un cliente adquiere una estructura de carpeta de un servidor, en donde un nuevo sub-elemento o un atributo debe extenderse o definirse en la orden de Alerta para incluir el parámetro de negociación del tipo de sincronización y un nuevo tipo de sincronización debe definirse también solamente para adquirir la estructura de carpeta de la base de datos, pero no para sincronizar la base de datos. Otro tipo de sincronización puede definirse también para adquirir la estructura de carpeta de la base de datos y los datos de sincronización. Un sistema de puesta en práctica para adquirir la estructura de carpeta se da a conocer a continuación.

55 1. Un cliente envía una orden para crear una sesión de sincronización a un servidor. Antes de enviar una orden de Alerta para determinar la dirección de la carpeta a sincronizarse, el cliente envía al servidor una orden de sincronización para adquirir la estructura de carpeta de la base de datos objetivo. Con el fin de distinguir esta orden de sincronización para adquirir la estructura de carpeta de la base de datos objetivo a partir de las dos órdenes de sincronización anteriores, la orden de sincronización puede denominarse una tercera orden de sincronización. La tercera orden de sincronización puede ponerse en práctica utilizando un elemento Get, p.e., la instancia operativa de orden Get ilustrada en la forma de realización siguiente, a modo de ejemplo. La orden Get existente solamente se utiliza para adquirir información de rendimiento de dispositivo, por lo que, con el fin de demandar la adquisición de la estructura de carpeta de la base de datos, debe extenderse la definición de la orden Get existente, en donde, un identificador para indicar una demanda de adquisición de la estructura de carpeta de la base de datos se añade en la definición.

65 <Get>

.....

```
<Target> <LocURI>/contacts?List = struct<LocURI></Target> ... ..
</Get>
```

5 En el elemento Target (Objetivo) ilustrado en la instancia operativa de la orden Get anterior, “/contacts” se refiere al URI de la base de datos objetivo, “?List=struct” se refiere al identificador para indicar la demanda de adquisición de la estructura de carpeta de base de datos. Y después de recibir la orden Get, el servidor encuentra la base de datos correspondiente al URI de la base de datos objetivo.

10 2. El servidor reenvía al cliente la estructura de carpeta de la base de datos objetivo mediante una respuesta de la orden Get, p.e., una orden Results. Se supone que la estructura de carpeta de la base de datos del servidor es que: bajo una carpeta raíz (esto es, el nodo de carpeta del primer nivel) de “/contacts” existen cuatro sub-nodos que incluyen A, B, C y D que se denominan el nodo de carpeta de segundo nivel y sub-nodos de A1 y A2 están bajo el nodo de carpeta A de segundo nivel y los dos sub-nodos se denominan el nodo de carpeta de tercer nivel. En este caso, la orden de Results reenviada puede organizarse como los diversos modos siguientes.

15 (1) Un elemento de datos Item se utiliza para indicar el URI de un nodo en la estructura de carpeta. Después de recibir la orden Results, el cliente construye la estructura de carpeta del servidor en función del URI indicado por cada elemento de datos Item. La forma de la orden de Results es como sigue:

```
<Results> .....
<Item> .....
25 <Origen><LocURI>/A</LocURI></Origen>.....
</Item>
30 <Item>.....
<Source><LocURI>/A/A1</LocURI></Source>.....
</Item>
35 </Results>
```

40 (2) Todos los datos de estructura de carpeta se encapsulan en un solo elemento. A modo de ejemplo, los datos de estructura de carpeta pueden encapsularse en un elemento de datos Data que es un elemento de datos Item, siendo el elemento Item un sub-elemento de un elemento Results, pudiendo los datos de estructura de carpeta, encapsulados en el elemento Data, existir como un fichero. En este caso, la forma de la orden Results es como sigue:

```
<Results> .....
45 <Item>.....
<Data> datos estructura carpeta </Data>
50 </Item>
</Results>
```

55 Otros sub-elementos o atributos pueden utilizarse también para indicar la demanda de adquisición de la estructura de carpeta en el elemento Get que funciona como la tercera orden de sincronización. Las estructura de órdenes del elemento Get y del elemento Results, bajo estos casos, no se ilustrarán una por una y sin importar qué clase de elemento Get y elemento Results se utilizan juntos, deben estar cubiertos también en el alcance de protección de esta invención. De modo similar, una orden Add (añadir) o una orden Replace (Sustituir) pueden extenderse también para contener los datos de estructura de jerarquía, cuyo modo de extensión es el mismo que el de la orden Results y debe estar también cubierto en el alcance de protección de esta invención.

60 Como alternativa, el cliente puede determinar la dirección de la carpeta a sincronizarse por medio del envío, por iniciativa del servidor, al cliente de la dirección de la carpeta a sincronizarse. El servidor que envía la dirección de la carpeta a sincronizarse puede ponerse en práctica enviando una notificación al cliente. El formato de la notificación puede ser el formato de notificación en el protocolo de sincronización de datos o la notificación puede enviarse por el servidor por intermedio de otros motores operativos, p.e., un servicio de mensajes cortos (SMS), un Servicio de

Empuje de Protocolo de Aplicación Inalámbrico (WAP Push), un Mensaje de Protocolo de Iniciación de Sesión (mensaje SIP), un Servicio de Mensajes Multimedia (MMS) y etc. El cliente puede captar el URI de cualquier clase de estructura jerárquica a sincronizarse.

5 Al determinar la dirección de la carpeta a sincronizarse, el cliente puede enviar, además, una orden de Alerta que contenga la dirección origen y objetivo determinada a sincronizarse y el parámetro de negociación del tipo de sincronización, la dirección de la carpeta a sincronizarse y el parámetro de negociación del tipo de sincronización puede contenerse en un elemento de datos Item y la forma de la orden de Alerta es como sigue:

```

10      <Alerta>
          <Sync type> sincronización bidireccional <Sync type>
          <Item>
15      <Target> <LocURI>/files/folder1 (el URI de la carpeta a sincronizarse en el servidor, denominada folder1)
          </LocURI></Target>
          <Source><LocURI>/files/folder1 (el URI de la carpeta a sincronizarse en el cliente, denominada folder1)
20 </LocURI></Source>.....
          </Item>.....
          </Alert>
25

```

Si existen múltiples carpetas a sincronizarse, un tipo de sincronización y múltiples elementos de Item pueden contenerse en una sola orden de Alerta y cada elemento de datos Item se utiliza para contener la dirección de una carpeta, en donde los tipos de sincronización de las múltiples carpetas a sincronizarse son los mismos.

30 Después de recibir la orden de Alerta enviada por el cliente, el servidor reenvía al cliente una respuesta que contiene el resultado de la negociación de la carpeta a sincronizarse, de modo que el cliente pueda determinar si continuar enviando los datos de sincronización al servidor o proceder a una renegociación. La respuesta puede ser una orden de sincronización para reenviar un código de estado operativo en el protocolo existente, a modo de ejemplo, una orden Status.

35 Si existen múltiples carpetas a sincronizarse, el modo para reenviar la respuesta al cliente por el servidor incluye: con respecto a todas las carpetas a sincronizarse, el servidor reenvía una sola respuesta que contiene los resultados de la negociación de todas las carpetas o, el servidor reenvía una sola respuesta para cada carpeta que contiene el resultado de la negociación de la carpeta, en donde múltiples órdenes de Status pueden utilizarse, respectivamente, para las múltiples carpetas para reenviar los resultados de la negociación de estas carpetas (el resultado de la negociación puede denominarse el código de estado operativo).

Para adoptar un tipo de sincronización diferente para una carpeta diferente, se pueden utilizar los sistemas siguientes.

45 1. Con respecto a múltiples carcasas a sincronizarse, el cliente envía al servidor una orden de Alerta que contiene las direcciones de todas las carpetas y los parámetros de negociación de los tipos de sincronización que corresponden, respectivamente, a cada carpeta. El servidor determina todas las carpetas a sincronizarse en función de las direcciones de todas las carpetas contenidas en la orden de Alerta y determina los tipos de sincronización correspondientes a las carpetas, respectivamente. En este caso, la orden Alerta contiene múltiples elementos (a modo de ejemplo, elementos Item) originalmente utilizados para contener la dirección de la base de datos de destino a sincronizarse, que se extienden para ser capaces de indicar una dirección a nivel de nodo y un tipo de sincronización y se utilizan para soportar la dirección de una carpeta (realmente, puede indicar cualquier nodo dentro de la estructura arborescente) a sincronizarse y el tipo de sincronización de la carpeta.

50 2. Para cada una de las carpetas a sincronizarse, el cliente envía al servidor una orden Alerta que contiene la dirección de la carpeta y el parámetro de negociación del tipo de sincronización de la carpeta. El servidor determina cada carpeta a sincronizarse y su tipo de sincronización en función de la dirección de carpeta y del parámetro de negociación del tipo de sincronización de la carpeta en cada orden Alerta. En este caso, la orden Alerta contiene un elemento originalmente utilizado para contener la dirección de la base de datos destino a sincronizarse y un elemento originalmente utilizado para contener el tipo de sincronización que se extiende para ser capaz de indicar una dirección a nivel de nodo y se utiliza para soportar la dirección de la carpeta (realmente, puede indicar cualquier nodo dentro de la estructura arborescente) a sincronizarse. El elemento originalmente utilizado para contener el tipo de sincronización se utiliza para soportar el parámetro de negociación del tipo de sincronización.

65 Sin embargo, si existen múltiples carpetas a sincronizarse y los tipos de sincronización de las múltiples carpetas son

los mismos, entonces, para todas las carpetas a sincronizarse, el cliente envía al servidor una orden Alerta que contiene las direcciones de todas las carpetas y el parámetro de negociación del tipo de sincronización de estas carpetas. El servidor determina los tipos de sincronización y todas las carpetas a sincronizarse en función de las direcciones de todas las carpetas y el parámetro de negociación en la orden Alerta. En este caso, la orden Alerta contiene múltiples elementos originalmente utilizados para contener la dirección de la base de datos de destino a sincronizarse (a modo de ejemplo, el elemento Item) y un elemento originalmente utilizado para contener el tipo de sincronización (a modo de ejemplo, el elemento Data), en donde cada elemento originalmente utilizado para contener la dirección de la base de datos de destino a sincronizarse se extiende a un elemento capaz de indicar una dirección a nivel de nodo y se utiliza para soportar la dirección de la carpeta a sincronizarse. El elemento originalmente utilizado para contener el tipo de sincronización soporta un parámetro de negociación del tipo de sincronización. El elemento de Alerta, que actúa como la segunda orden de sincronización puede utilizar también otros sub-elementos o atributos para contener el parámetro de negociación del tipo de sincronización, la dirección de la carpeta a sincronizarse y etc. Las estructuras de las órdenes, en estos casos, no se ilustrarán una por una, pero deben estar cubiertas dentro del alcance de protección de esta invención.

II. El modo 2 es un modo para determinar la dirección de la carpeta a sincronizarse utilizando un mecanismo de filtro.

El mecanismo de filtro existente solamente puede realizar el filtrado a nivel de ficheros y a nivel de elementos de datos, lo que depende principalmente del formato de los datos a sincronizarse. Y en una forma de realización de la presente invención, el mecanismo de filtrado existente se extiende al nivel de nodos (en este caso, el nivel de nodos se refiere al nivel de carpetas) y no está limitado por el formato de los datos a sincronizarse y puede utilizarse para todas las clases de formatos de datos, insertando un sub-elemento para contener una condición de filtrado a nivel de carpeta en el elemento de filtro correspondiente del protocolo SyncML, a modo de ejemplo, el elemento se puede denominar FolderLevel, en donde se puede utilizar la sintaxis de Interfaz de Pasarela Común (CGI) para indicar la carpeta a sincronizarse. Un atributo o un sub-elemento recientemente añadido del elemento de filtro puede utilizarse también para contener la condición de filtrado al nivel de carpeta y la estructura de orden de este caso no se indica a continuación pero debe estar cubierta dentro del alcance de protección de esta invención.

La forma de la orden Filter se proporciona como sigue, conteniendo dos carpetas a sincronizarse, respectivamente numeradas 112 y 113.

```

<Filter>.....
    <NodeLevel>
        <Item>
            <Meta><Type>syncml:filtertype-cgi</Type></Meta>
            <Data>&LUID&EQ; 112&AND;&LUID&EQ;113 (utilizando la sintaxis CGI del
            elemento Filter para indicar los números de las carpetas a sincronizarse que incluye 112 y 113) </Data>
        </Item>
    </NodeLevel>.....
</Filter>
    
```

En la fase de sincronización, el tipo de operación de la sincronización basada en cualquier nodo se indica enviando una primera orden de sincronización que puede indicar, además, la dirección de la carpeta a sincronizarse o incluir, además, el contenido de datos.

Si la dirección de la carpeta a sincronizarse se incluye en la primera orden de sincronización, el elemento utilizado para contener la dirección de los datos a sincronizarse, en la orden Sync existente, pueden extenderse, p.e., un elemento objetivo Target con el fin de ampliar la granularidad de sincronización desde el nivel de base de datos a cualquier nivel de nodos. A modo de ejemplo, el elemento Target capaz de indicar una dirección al nivel de nodos (p.e., el URI de la carpeta a sincronizarse) puede incluirse en la orden Sync. La orden Sync existente solamente puede incluir el elemento Target que solamente indica la dirección de la base de datos a sincronizarse. Sin embargo, la orden Sync, según una forma de realización de esta invención, puede indicar la dirección de cualquier nivel de nodos y contener los datos pertenecientes a la gama de las direcciones indicadas, pero no la totalidad de los datos de la base de datos completa.

En la solución anterior, una orden Sync puede contener la dirección de una carpeta a sincronizarse, si existen múltiples carpetas a sincronizarse, las direcciones de las múltiples carpetas pueden contenerse en múltiples órdenes Sync respectivamente o incluso en una sola orden Sync. A modo de ejemplo, las direcciones de dos carpetas a

sincronizarse, correspondientemente denominadas folder1 y folder2, se incluyen en dos órdenes Sync respectivamente. En el elemento de sincronización anterior que actúa como la primera orden de sincronización, otros sub-elementos o atributos pueden utilizarse para contener la dirección de la carpeta a sincronizarse y las estructuras de órdenes en estos casos no se ilustrarán, una a una, pero deben estar cubiertas dentro del alcance de protección de esta invención.

Cuando la primera orden de sincronización (a modo de ejemplo, una orden Sync) contiene un tipo de operación, el modo específico para realizar un tipo de operación es el mismo que el de la técnica anterior, que es la de utilizar un elemento de Add, (Añadir), Replace (Sustituir), Delete (Suprimir) o Move (Desplazar), etc. para realizar el tipo de operación. A modo de ejemplo, un sub-elemento puede incluirse en la orden de sincronización Sync, el sub-elemento puede ser un elemento de Add para indicar una operación de adición o un elemento de Replace para indicar una operación de sustitución o un elemento de Delete para indicar una operación de supresión o un elemento de Move para indicar una operación de desplazamiento, etc. De este modo, el lado receptor, que recibe la primera orden de sincronización, puede realizar la operación de sincronización designada por el tipo de operación para la carpeta a sincronizarse, en función del tipo de operación realizada en la primera orden de sincronización. Por ello, es posible seleccionar un elemento de adición Add, un elemento de sustitución Replace, un elemento de supresión Delete o un elemento de desplazamiento Move y contener el elemento seleccionado en la orden Sync para indicar varios tipos de operación en conformidad con la situación práctica.

La diversa información contenida en todas las primera, segunda y terceras órdenes de sincronización, antes citadas, de las formas de realización de esta invención, a modo de ejemplo, los datos del nodo a sincronizarse, la condición de filtrado de direcciones a nivel de nodos, el tipo de sincronización, los datos de estructuras jerárquicas, etc., no solamente pueden contenerse en el sub-elemento o atributo en las diversas órdenes anteriormente indicadas, sino que también pueden contenerse en otros sub-elementos o atributos. Considerando que existen numerosos casos de puesta en práctica de estas órdenes, las formas de estas órdenes no se ilustran, una a una en esta descripción, pero deben estar cubiertas dentro del alcance de protección de esta invención.

La fase de sincronización se describe, a continuación, haciendo referencia a algunas formas de realización específicas. Para el procedimiento de sincronización iniciado por un cliente que sea similar al iniciado por un servidor, la fase de sincronización se describe, a continuación, tomando como ejemplo el de un cliente que inicia el procedimiento de sincronización y un servidor que realiza la operación de sincronización de datos. En las formas de realización siguientes, la primera orden de sincronización se pone en práctica utilizando el elemento de sincronización del protocolo SyncML y el elemento de sincronización puede contener el tipo de operación de Añadir, Sustituir, Suprimir o Desplazar, etc.

Con el fin de permitir al usuario crear una carpeta física o lógica y designar cualquiera de las carpetas para ser sincronizadas de forma recursiva y sincronizadas de forma no recursiva, se pueden configurar tres tablas, respectivamente, en el cliente y en el servidor:

1. Una Tabla de Elementos de Datos, utilizada para memorizar información de todos los elementos de datos, incluyendo la relación de correspondencia del UID del elemento de datos y los contenidos de datos (Data), en donde el UID del elemento de datos se indica como Item LUID e Item GUID, respectivamente, en el cliente y en el servidor.

2. Una Tabla de Carpetas, utilizada para memorizar información de todas las carpetas, incluyendo la información de cada carpeta el número de la carpeta, la denominación de la carpeta (Name), la carpeta principal a la que pertenece la carpeta (Parent Source), el estado de la carpeta (Folder Status) y su relación de correspondencia, en donde el estado operativo de la carpeta incluye principalmente: Existente (identificada por E), Adición (identificada por N), Actualización (identificada por U), Supresión (identificada por D), Desplazamiento (identificada por M) y Copia (identificada por C). El estado operativo de Supresión puede incluir, además, dos estados operativos, Supresión Permanente (identificada por P-D, también denominada Supresión de Hardware Hard Deletion) y Supresión no Permanente (identificada por P-ND también denominada Supresión de Software Soft Deletion) y el UID de la carpeta se indica como Folder LUID y Folder GUID respectivamente, en el cliente y en el servidor. En la técnica anterior, LUID es diferente de GUID. Sin embargo, podrían ser los mismos. La evolución del mecanismo de LUID-GUID no afectaría al alcance de protección de esta invención.

3. Una Tabla de Índices de "elemento de datos –carpetas", utilizada para memorizar la afiliación de los elementos de datos, que incluye el UID del elemento de datos, el elemento de datos principal (Parent Source), el estado operativo del elemento de datos (Data Status) y su relación de correspondencia, en donde el UID del elemento de datos se indica como ítem LUID e ítem GUID, respectivamente, en el cliente y en el servidor.

Además, la tabla de la relación de correspondencia entre el UID de los datos en el cliente y la existente en el servidor debe memorizarse en el servidor, esto es, la relación de correspondencia entre GUID y LUID.

En la primera forma de realización de la presente invención, un usuario añade una nueva carpeta "bless" bajo la carpeta raíz (p.e., /sms) para mensajes cortos, añade dos sub-carpetas de "Spring Festival" y "Mid-autumn Festival" bajo la carpeta "bless" y respectivamente, añade datos bajo las carpetas, p.e., un dato de referencia N1 se añade

bajo la carpeta "bless", un dato de referencia N2 se añade bajo la carpeta "Spring Festival" y un dato de referencia N3 se añade bajo la carpeta "Mid-autumn Festival".

Según se ilustra en las Figuras 3a y 3b, la Figura 3a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario en la primera forma de realización de esta invención, en donde una trama cuadrada indica una carpeta, un círculo indica un elemento de datos, una línea continua indica el estado operativo de Existente y una línea de trazos indica el estado operativo de Nuevo. La Figura 3b es un diagrama esquemático que ilustra la memorización de datos de un cliente en la primera forma de realización de esta invención. En el cliente, se memoriza una Tabla de Elementos de Datos, una Tabla de Carpetas y una Tabla de Índices de "elementos de datos-carpetas". El estado operativo de datos añadidos en cada tabla se ilustra, correspondientemente, en la Figura 3b.

Cuando el usuario demanda sincronizar la carpeta "bless", el cliente crea las siguientes órdenes de sincronización de forma ordenada.

En primer lugar, después de determinar los datos a sincronizarse para una carpeta en función de la orden para sincronizar la carpeta "bless" desde el usuario, el cliente determina, en conformidad con la Tabla de Carpetas, que el estado operativo de la carpeta "bless" es N. Más adelante, el cliente construye una orden de sincronización que indica la adición de una carpeta, p.e., un sub-elemento de Add se añade en la orden Sync para construir una orden de sincronización Add que puede denominarse también una sub-orden de la orden Sync. En la orden Add construida, se utiliza un campo Meta para indicar que el tipo de datos es una carpeta. El tipo de datos se determina en conformidad con la Tabla de Carpetas en donde se utiliza un campo LUID para indicar que el UID de los datos a sincronizarse es 1006, se utiliza un campo Data para indicar que el contenido de datos es "bless" y un campo SourceParent se utiliza para indicar que la carpeta principal es la carpeta raíz.

En segundo lugar, el cliente determina el estado operativo del elemento de datos bajo la carpeta "bless" y puesto que el estado operativo correspondiente al elemento de datos 2001 en la Tabla de Índices es N, se construye una orden de sincronización Add. Después de determinar, sobre la base de la Tabla de Elementos de Datos que el contenido de datos correspondiente al elemento de datos 2001 es N1, en la orden de sincronización Add construida, se utiliza un campo Meta para indicar que el tipo de datos es vCard (que se conoce como una clase de elemento de datos), se utiliza un campo LUID para indicar que el UID de los datos a sincronizarse es 2001, se utiliza un campo Data para indicar que el contenido de datos es N1 y un campo SourceParent se utiliza para indicar que la carpeta principal es 1006.

Si el cliente ha determinado que no existe ningún elemento de datos recientemente añadido bajo la carpeta "bless", en tal caso, comprueba el estado operativo de la sub-carpeta bajo la carpeta "bless", cuya realización es la misma que la de la determinación de la carpeta "bless". Se construyen dos órdenes de sincronización Add, en una orden Add, existiendo un campo Meta que indica que el tipo de datos es de carpeta Folder, indicando un campo LUID que el UID de los datos a sincronizarse es 1007, un campo Data que indica que el contenido de datos es "Spring Festival" y un campo SourceParent que indica que la carpeta principal es 1006 y en la otra orden Add, existe un campo Meta que indica que el tipo de datos es de carpeta Folder, un campo LUID que indica que el UID de los datos a sincronizarse es 1008, un campo Data que indica que el contenido de datos es "Mid-autumn Festival" y un campo SourceParent que indica que la carpeta principal es 1006.

Después de determinar que no existe ningún elemento de datos recientemente añadido bajo la carpeta "bless", el cliente comprueba el estado de los elementos de datos bajo la carpeta "Spring Festival" y la carpeta "Mid-autumn Festival", cuya realización es la misma que la de la determinación de N1, esto es, el cliente construye, respectivamente, dos órdenes de sincronización Add (Añadir).

De este modo, se realiza la sincronización recursiva enviando la orden de sincronización Add para cada elemento de datos recientemente añadido. La sincronización no recursiva se refiere a los casos siguientes: 1. Solamente sincronizando una carpeta pero no el elemento de datos bajo la carpeta, p.e., modificación del nombre de la carpeta; 2. Solamente sincronizando una carpeta y el elemento de datos que contiene, pero no sincronizando las sub-carpetas.

Por último, todas las órdenes de sincronización Add construidas se envían al servidor. Si la cantidad de datos dentro de una orden Add es relativamente pequeña, múltiples órdenes Add pueden incluirse en un solo mensaje y enviarse al servidor mediante un intercambio de una sola vez y si la cantidad de datos de una orden Add es relativamente grande, se necesita más de un mensaje para transmitir múltiples órdenes Add y todas las órdenes Add pueden enviarse al servidor mediante múltiples intercambios. En la práctica, es posible enviar solamente una orden de sincronización Add que incluya múltiples carpetas y elementos de datos, mientras que la orden de sincronización Add única debe considerarse como múltiples órdenes Add lógicas.

El proceso para realizar la operación de sincronización después de que el servidor reciba la orden Add anterior se describe más adelante. La tabla relacionada con el proceso es similar a la tabla ilustrada en la Figura 3b.

Después de recibir la orden de sincronización para añadir la carpeta "bless", el servidor determina que el tipo de los datos a sincronizarse es de carpeta Folder en función del campo Meta incluido en la orden de sincronización

recibida, el UID de los datos a sincronizarse en el cliente es 1006 en función del campo LUID, el nombre de la carpeta a sincronizarse es "bless" en función del campo Data y la carpeta principal es la carpeta raíz en función del campo SourceParent. Más adelante, un UID local del servidor (Folder GUID) se asigna a los datos a sincronizarse, tal como 100006. A continuación, el elemento de datos correspondiente se añade en la Tabla de Carpetas configurada en local, esto es, el elemento de datos incluyendo 100006, "bless", la carpeta raíz, el estado operativo del elemento de datos y su relación de correspondencia, se añaden. Además, el UID de los datos a sincronizarse en el cliente (esto es, LUID), el UID de los datos a sincronizarse en el servidor (esto es, GUID) y su relación de correspondencia se memorizan en la tabla configurada de la relación de correspondencia entre LUID y GUID, esto es, 1006, 100006 y su relación de correspondencia son objeto de memorización.

Si el servidor ha recibido la orden de sincronización para añadir el elemento de datos N1, determina que el tipo de los datos a sincronizarse es el elemento de datos en función del campo Meta en la orden de sincronización recibida, el UID de los datos en el cliente es 2001 en función del campo LUID, el contenido de datos es N1 en función del campo Data y la carpeta principal es 1006 en función del campo SourceParent. En adelante, el elemento de datos N1 se memoriza en la base de datos local y luego, un UID local del servidor (Folder GUID), tal como, 200001, se asigna a los datos a sincronizarse, el elemento de datos correspondiente, tal como 200001, N1 y su relación de correspondencia se añaden en la Tabla de Elementos de Datos configurada en local; el elemento de datos correspondiente, tal como 2000001, 100006, el estado operativo del elemento de datos N1 y su relación de correspondencia se añaden en la Tabla de Índices; y el UID de los datos a sincronizarse en el cliente, el UID de los datos a sincronizarse en el servidor y su relación de correspondencia, esto es 2001, 200001 y su relación de correspondencia, se memorizan en la tabla configurada de la relación de correspondencia entre LUID y GUID.

El modo para añadir las carpetas "Spring Festival" y "Mid-autumn Festival" en el servidor es el mismo que el utilizado para añadir la carpeta "bless" y el modo para añadir los elementos de datos N2 y N3 es el mismo que para añadir el elemento de datos N1.

Además, conviene señalar que si el servidor envía órdenes de sincronización al cliente para realizar la operación de sincronización, el UID de los datos a sincronizarse en el servidor, se incluye en las órdenes de sincronización, después de realizar la operación de sincronización, el cliente reenvía al servidor, la relación de correspondencia entre el UID de los datos en el cliente y el del servidor, esto es, la relación de correspondencia entre LUID y GUID y el servidor memoriza la relación de correspondencia recibida en la tabla de la relación de correspondencia entre LUID y GUID configurada a nivel local.

De este modo, la operación de sincronización para añadir datos puede realizarse, en donde los datos pueden ser un elemento de datos específico o una carpeta creada por el usuario según se desee, lo que no está limitado por la estructura de datos físicos del sistema. Puede deducirse que las ventajas de una forma de realización de la invención es que, para un dato de referencia que pueda pertenecer a múltiples nodos en una estructura de datos, solamente una copia de referencia se transmite en la sincronización de datos de los múltiples nodos y solamente una copia del dato se memoriza por el lado que realiza la operación de sincronización, con lo que se ahorran, en gran medida, los recursos de la red y del dispositivo. A modo de ejemplo, se supone que N1 pertenece a las carpetas "bless", "Spring Festival" y "Mid-autumn Festival", durante la operación de sincronización en el servidor, debiendo añadirse dos entradas correspondientes en la Tabla de Índices, esto es, elemento 1 "200001, 100007, el estado operativo del dato N1 y su relación de correspondencia" y el elemento de datos 2 "200001, 100008, el estado operativo del dato N1 y su relación de correspondencia".

En la segunda forma de realización, el usuario actualiza el atributo de la carpeta "bless" bajo la carpeta raíz para mensajes cortos (/sms), el elemento de datos U1 en la carpeta "bless" y el elemento de datos U2 en la carpeta "Spring Festival". En esta forma de realización, U2 pertenece a las carpetas "Spring Festival" y "Mid-autumn Festival".

Según se indica en las Figuras 4a y 4b, la Figura 4a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario en la segunda forma de realización de esta invención, en donde una trama cuadrada indica una carpeta Folder, un círculo indica un elemento de datos Item, una línea continua que el estado operativo es Existente y una línea de puntos-trazos indica que el estado operativo es Update (Actualizar). La Figura 4b es un diagrama esquemático que ilustra la memorización de datos de un cliente en la segunda forma de realización de esta invención. Una Tabla de Elementos de Datos, una Tabla de Carpetas y una Tabla de Índices se memorizan en el cliente. El estado operativo de datos en cada tabla se ilustra correspondientemente en la Figura 4b.

Si el usuario demanda la sincronización de la carpeta "bless", el cliente crea las órdenes de sincronización siguientes de manera ordenada.

En primer lugar, al determinar los datos a sincronizarse en una carpeta en función de la orden recibida desde el usuario para sincronizar la carpeta "bless", el cliente determina el estado operativo de la carpeta "bless" en la Tabla de Carpetas que es U y construye una orden de sincronización para indicar la actualización, p.e., una orden de sincronización Replace que puede denominarse una sub-orden de la orden Sync. Y en la orden Replace construida, un campo Meta se utiliza para indicar que el tipo de datos es de carpeta Folder, que se determina en función de la

Tabla de Carpetas, se utiliza un campo LUID para indicar que el UID de los datos a sincronizarse es 1006, se utiliza un campo de datos Data para indicar que los datos específicos es "bless" y un campo SourceParent se utiliza para indicar que la carpeta principal es la carpeta raíz.

5 Más adelante, un cliente determina el estado operativo del elemento de datos en la carpeta "bless", como en la Tabla de Índices, el elemento de datos correspondiente al número 1006 es 2001, cuyo estado operativo es U, construyéndose una orden de sincronización Replace. Después del contenido de datos específico correspondiente al elemento de datos 2001, según se determina a partir de la Tabla de Elementos de Datos es N1, en la orden de sincronización Replace construida, se utiliza un campo Meta para indicar que el tipo de datos es vCard (conocido como la clase de elemento de datos), un campo LUID se utiliza para indicar que el UID de los datos a sincronizarse es 2001, un campo Data se utiliza para indicar que los datos específicos son U1 y el campo SourceParent se utiliza para indicar que la carpeta principal a la que pertenece el elemento de datos es 1006.

15 A continuación, después de determinar si no existen elementos de datos a sustituirse, en la carpeta "bless", el cliente comprueba el estado operativo de una o más sub-carpetas en la carpeta "bless", cuya realización es la misma que para determinar la carpeta "bless" y, en esta forma de realización, el estado operativo de la sub-carpeta de la carpeta "bless" no ha sido cambiado, por lo que no necesitan procesarse las sub-carpetas.

20 Por último, cuando se determina que no existen sub-carpetas a sincronizarse bajo la carpeta "bless", el cliente comprueba el estado operativo del elemento de datos en la sub-carpeta "Spring Festival", cuya realización es la misma que para determinar el elemento de datos U1. Es decir, el cliente construye una orden de sincronización Replace y después del contenido de datos específico correspondiente al número 2002, determinado a partir de la Tabla de Elementos de Datos es U2, en la orden de sincronización Replace construida, se utiliza un campo Meta para indicar que el tipo de datos es vCar (conocido como la clase de elementos de datos), un campo LUID se utiliza para indicar que el UID de los datos a sincronizarse es 2002, un campo Data se utiliza para indicar que los datos específicos son U2 y un campo SourceParent se utiliza para indicar que la carpeta principal a la que pertenece el elemento de datos es 1007.

30 De este modo, puede realizarse la sincronización recursiva enviando una orden de sincronización Replace para cada dato a sustituirse. Por supuesto, se puede realizar también una sincronización no recursiva, cuyo principio de puesta en práctica es similar al de la sincronización recursiva y por ello, no se proporciona aquí ninguna descripción adicional a este respecto. Si se realiza una sincronización recursiva, o no, puede determinarse durante la negociación de la carpeta a sincronizarse en la fase de inicialización de sincronización, en donde un identificador de sincronización recursiva se incluye, además, en la segunda orden de sincronización de las formas de realización de esta invención. Un identificador de sincronización recursiva con el valor 'verdadero' indica que debe realizarse una sincronización recursiva para la carpeta a sincronizarse, de modo que la sincronización de datos deba realizarse para el nodo raíz y todos los sub-nodos de la carpeta a sincronizarse. Si un identificador de sincronización recursiva, con el valor 'falso' se incluye en la segunda orden de sincronización, ello indica que debe realizarse una sincronización no recursiva para la carpeta a sincronizarse, de modo que solamente debe realizarse la sincronización de datos para el nodo raíz de la carpeta a sincronizarse. Si no está presente ningún identificador de sincronización recursiva con el valor 'verdadero', podría considerarse como sincronización recursiva siempre. Si se determina que debe realizarse una sincronización recursiva para la carpeta a sincronizarse en la fase de inicialización de sincronización, el contenido de datos incluido en la primera orden de sincronización, enviada por el cliente o el servidor en la fase de sincronización, incluye: los contenidos de datos de la carpeta raíz y de cada sub-carpeta de la carpeta a sincronizarse y el lado receptor de la primera orden de sincronización sincroniza el contenido de datos de la carpeta raíz y cada sub-carpeta de la carpeta a sincronizarse, por turno, en función de los contenidos de datos incluidos en la primera orden de sincronización. Y si se determina que debe realizarse una sincronización no recursiva para la carpeta a sincronizarse en la fase de inicialización de sincronización, los contenidos de datos incluidos en la primera orden de sincronización, enviada por el cliente o el servidor en la fase de sincronización, incluye: el contenido de datos de la carpeta raíz de la carpeta a sincronizarse y el lado receptor de la primera orden de sincronización solamente sincroniza el contenido de datos de la carpeta raíz de la carpeta a sincronizarse. En este caso, el contenido de datos descrito en la forma de realización de esta invención indica el contenido de la carpeta y del elemento de datos de la carpeta a sincronizarse, a modo de ejemplo, la denominación de la carpeta (Name) y los datos del elemento de datos (Data).

55 En adelante, todas las órdenes de sincronización Replace construidas se envían al servidor, el método de transmisión detallado es el mismo que para la transmisión de una orden de sincronización Add, por lo que no se describirá aquí con mayor detalle.

60 El proceso para realizar la operación de sincronización después de que el servidor reciba la orden Replace anterior se describe a continuación. La tabla relacionada con el proceso es similar a la tabla ilustrada en la Figura 4b por lo que no se describirá aquí con más detalle.

65 Después de recibir la orden de sincronización para sustituir la carpeta "bless", el servidor determina que el tipo de los datos a sincronizarse es el de carpeta Folder en función del campo Meta de la orden de sincronización recibida, el UID de los datos a sincronizarse en el cliente es 1006 en función del campo LUID, el nombre de los datos a sincronizarse es "bless" en función del campo Data y la carpeta principal de los datos a sincronizarse es a carpeta

raíz en función del campo SourceParent. En adelante, el número de datos a sincronizarse, que ha de sustituirse, en el servidor local, p.e., 100006, se adquiere a partir de la tabla configurada de la relación de correspondencia entre LUID y GUID. A continuación, el elemento de datos correspondiente se sustituye en la Tabla de Carpetas configurada local, esto es, la información del atributo de "bless" en el elemento de datos es sustituido, en donde, el elemento de datos incluye: 100006, "bless", la carpeta raíz, el estado operativo del elemento de datos "bless" y su relación de correspondencia.

Después de recibir la orden de sincronización para sustituir el elemento de datos U1, el servidor determina que el tipo de los datos a sincronizarse es de elemento de datos en función del campo Meta de la orden de sincronización recibida. El UID de los datos a sincronizarse en el cliente es 2001 en función del campo LUID, el contenido de datos específico es U1 en función del campo Data y la carpeta principal es 1006 en función del campo SourceParent. En adelante, el UID de los datos a sincronizarse, que han de sustituirse en este caso, en el servidor local (p.e., 200001) se adquiere a partir de la tabla configurada de la relación de correspondencia entre LUID y GUID y el elemento de datos correspondiente al UID local de los datos a sincronizarse se sustituye en la tabla de elementos datos configurada en local, esto es, la información de U1 en el elemento de datos de 200001, U1 y su relación de correspondencia son objeto de sustitución.

El método para sustituir U2 en el servidor es el mismo que el de para sustituir U1, por lo que no se describirá aquí con mayor detalle.

Conviene señalar que, en esta forma de realización, U2 pertenece a la carpeta "Spring Festival" y la carpeta "Mid-autumn Festival", pero solamente una vez la orden Replace necesita enviarse para sustituir U2 y U2 solamente necesita sustituirse una vez en el servidor, de modo que el U2 en ambas carpetas pueda sustituirse. Ésta es la razón por la que solamente una copia de los datos se memoriza en el servidor realmente y la descripción de los datos se indica en la Tabla de Índices. Puede deducirse que los datos de redundancia pueden disminuirse todavía más utilizando el método de las formas de realización de esta invención, de modo que puedan ahorrarse recursos.

En la tercera forma de realización, el usuario desplaza el elemento de datos "M1" desde la carpeta "música" a la carpeta "favorito" y desplaza la carpeta completa "mp3" a "favorito".

Según se ilustra en las Figuras 5a y 5b, la Figura 5a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario utilizando la tercera forma de realización de esta invención, en donde la trama cuadrada indica que el estado operativo es Existente y la línea de puntos y trazos indica que el estado operativo es Move (Desplazamiento). La Figura 5b es un diagrama esquemático que ilustra la memorización de datos de un cliente utilizando la tercera forma de realización de esta invención. La Tabla de Elementos de Datos, la Tabla de Carpetas y la Tabla de Índices se memorizan en un cliente. El estado operativo de datos, en cada tabla, tiene el reflejo correspondiente en la Figura 5b.

Cuando el usuario demanda sincronizar la carpeta raíz, el cliente crea la orden de sincronización siguiente de forma ordenada:

En primer lugar, un cliente comprueba todos los estados operativos de sub-carpetas de la carpeta raíz en función de la orden para sincronizar la carpeta raíz recibida desde el usuario. En esta forma de realización, todos los estados operativos de sub-carpetas de la carpeta raíz son invariables, por lo que no necesitan procesarse. A continuación, el cliente comprueba si se cambia el estado operativo del elemento de datos bajo la carpeta raíz. En esta forma de realización, el estado operativo del elemento de datos bajo la carpeta raíz no sufre cambio alguno, por lo que no necesita procesarse.

Más adelante, el cliente comprueba si se cambia, por turno, cada estado operativo de sub-carpeta. En esta forma de realización, el cliente determina que el estado operativo de la sub-carpeta "mp3" bajo la carpeta de música es M y luego construye una orden de sincronización Move. La orden de sincronización Move puede denominarse también una sub-orden de la orden Sync, que se utiliza para transmitir los datos de desplazamiento pendientes. En la orden Move construida, se utiliza un campo Meta para indicar que el tipo de datos es una carpeta Folder, el tipo de datos se determina en función de una Tabla de Carpetas, un campo LUID se utiliza para indicar que el número de datos a sincronizarse es 1006 y un campo SourceParent se utiliza para indicar que la carpeta principal después de la carpeta que se desplaza es 1004.

A continuación, el cliente determina el estado operativo del elemento de datos bajo la carpeta de música, puesto que el elemento de datos correspondiente a 1006 es 2001 en la Tabla de Índices y su estado operativo es M, se construye una orden de sincronización Move y en la orden construida, se utiliza un campo Meta para indicar que el tipo de datos es un elemento de datos, un campo LUID se utiliza para indicar que el número de datos a sincronizarse es 2001 y se utiliza un campo SourceParent para indicar que la carpeta principal después de que se desplace la carpeta es 1004.

De este modo, no existe ningún otro dato desplazado en esta forma de realización y por ello no se realiza ningún otro proceso.

Más adelante, todas las órdenes de sincronización Move construidas se envían al servidor. El modo de transmisión detallado es el mismo que el de transmisión de una orden de sincronización Add, por lo que no se describirá aquí con más detalle.

5 El proceso de realizar la operación de sincronización después de que el servidor reciba la orden Move anterior se describe a continuación. La tabla relacionada en el proceso es similar a la tabla ilustrada en la Figura 5b y la tabla no se ilustra en esta descripción.

10 Después de recibir la orden de sincronización Move que indica el desplazamiento de la carpeta mp3, el servidor determina que el tipo de datos a sincronizarse es una carpeta Folder en función del campo Meta recibido de la orden de sincronización Move y determina que el número de datos a sincronizarse en el cliente es 1006 en función del campo LUID y determina que la carpeta principal después de que la carpeta mp3 se desplace es 1004 en función del campo SourceParent. Más adelante, el número de datos de desplazamiento pendientes en el servidor (a modo de ejemplo: 100006) se adquiere a partir de la tabla configurada de la relación de correspondencia entre LUID y GUID y la carpeta principal se cambia en la carpeta principal incluida en la orden de sincronización Move recibida, esto es, la carpeta principal correspondiente a 100006 en la tabla se cambia desde 1005 a 1004, en el elemento de datos correspondiente al número de datos locales a sincronizarse en la Tabla de Carpetas configuradas a nivel local.

20 Después de recibir la orden de sincronización Move que indica el desplazamiento del elemento de datos M1, el servidor determina que el tipo de datos a sincronizarse es un elemento de datos en función del campo Meta recibido de la orden de sincronización Move y determina que el número de datos a sincronizarse en el cliente es 2001 en función del campo LUID y determina que la carpeta principal después de que el elemento de datos M1 se desplace es 1004 en función del campo SourceParent, más adelante, el número sustituido de datos a sincronizarse en el servidor local, a modo de ejemplo: 200001 se adquiere a partir de la tabla configurada de la relación de correspondencia entre LUID y GUID, la carpeta principal se cambia en la carpeta principal incluida en la orden de sincronización recibida, esto es, la carpeta principal correspondiente a 200001 en la tabla se cambia desde 1005 a 1004, en el elemento de datos correspondiente al número de datos locales a sincronizarse en la Tabla de Índices configurada a nivel local.

30 Puede deducirse que, el método de las formas de realización de esta invención solamente necesita modificar la relación de correspondencia de la tabla de datos correspondiente y no necesita desplazar los datos reales cuando se realiza la operación de sincronización de desplazamiento, de modo que se ahorran recursos limitados en la medida máxima.

35 Conviene señalar que cuando una carpeta así como su sub-carpeta y el elemento de datos se desplazan, a modo de ejemplo, cuando se desplaza una carpeta de mp3, solamente una orden Move necesita enviarse en relación con la carpeta mp3 y otras órdenes Move, en relación con la sub-carpeta y el elemento de datos bajo la carpeta mp3 no necesitan enviarse, porque su sub-carpeta y la carpeta principal a la que pertenece el elemento de datos, no sufrieron cambios.

40 Cuando el tipo de operación realizada en la primera orden de sincronización es la de supresión, y la etapa de sincronización del elemento de datos bajo la carpeta a sincronizarse incluye, además: decidir si el contenido del elemento de datos de la carpeta a sincronizarse solamente se memoriza bajo la carpeta a sincronizarse y si es así, la inclusión adicional de un identificador de supresión permanente con el valor 'verdadero' en la primera orden de sincronización; de no ser así, la inclusión adicional de un identificador de supresión permanente con el valor 'falso' en la primera orden de sincronización; la operación de sincronización de datos realizada para la carpeta a sincronizarse por la primera orden de sincronización incluye: decidir si un identificador de supresión permanente con el valor 'verdadero' se incluye en la primera orden de sincronización y si es así, la supresión del contenido de datos del elemento de datos bajo la carpeta a sincronizarse; de no ser así, la cancelación de la relación de correspondencia entre el elemento de datos y la carpeta a sincronizarse. Si no está presente ningún identificador de sincronización permanente con el valor 'verdadero', podría considerarse como supresión permanente siempre.

50 En la cuarta forma de realización, el usuario suprime el elemento de datos "D1" bajo la carpeta "bless", y selecciona una supresión permanente con respecto a los datos "U2" bajo la carpeta "Spring Festival" y selecciona una supresión no permanente con respecto a los datos "D3" bajo la carpeta "Spring Festival". En esta forma de realización, solamente se describe la supresión del elemento de datos.

60 Según se indica en las Figuras 6a y 6b, la Figura 6a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario utilizando la cuarta forma de realización de esta invención en donde la trama cuadrada indica una carpeta Folder y el círculo indica un elemento de datos Item; la línea continua indica que el estado operativo es Existente y la línea de puntos indica que el estado operativo es Supresión. La Figura 6b es un diagrama esquemático que ilustra la memorización de datos de un cliente utilizando la cuarta forma de realización de esta invención. La Tabla de Elementos de Datos, la Tabla de Carpetas y la Tabla de Índices se memorizan en un cliente. El estado operativo de datos en cada tabla tiene el reflejo correspondiente en la Figura 6b.

Cuando el usuario demanda sincronizar la carpeta "bless", el cliente crea la orden de sincronización siguiente de forma ordenada:

Un cliente determina el estado operativo del elemento de datos bajo la carpeta "bless". Cuando el estado operativo de un elemento de datos 2001 es P-D, se construye una orden de sincronización Delete. La orden de sincronización Delete puede denominarse también una sub-orden de la orden Sync que se utiliza para realizar la supresión de algunos datos y en la orden de sincronización Delete construida, se utiliza un campo Meta para indicar que el tipo de datos es un elemento de datos Item, un campo LUID se utiliza para indicar que el número de datos a sincronizarse es 2001 y un identificador que indica la supresión permanente necesita también incluirse en la orden de sincronización Delete.

Después de determinar que no existe ningún elemento de datos suprimido bajo la carpeta "bless", un cliente comprueba el estado de sub-carpeta bajo la carpeta "bless". En esta forma de realización, el estado operativo de la sub-carpeta bajo la carpeta "bless" no sufre cambio alguno, por lo que no necesita procesarse.

Después de determinar que no existe ninguna sub-carpeta suprimida bajo la carpeta "bless", el cliente determina el estado del elemento de datos bajo la sub-carpeta Spring Festival. El método detallado es el mismo que el de la determinación D1. Es decir, el resultado final es construir dos órdenes de sincronización Delete en una orden de sincronización Delete única, se utiliza un campo Meta para indicar que el tipo de datos es una tarjeta vCard (conocida como la clase de elemento de datos), un campo LUID se utiliza para indicar que el número de datos a sincronizarse es 2002 y un identificador que indica la supresión permanente necesita incluirse también en la orden, a modo de ejemplo: P-D. y en la otra orden Delete, se utiliza un campo Meta para indicar que el tipo de datos es una tarjeta vCard (conocida como la clase de elemento de datos), un campo LUID se utiliza para indicar que el número de datos a sincronizarse es 2003 y un identificador que indica la supresión no permanente necesita incluirse en la orden, a modo de ejemplo: NP-D.

La orden de sincronización Delete construida no necesita incluir los datos de supresión pendientes y solamente necesita indicar el tipo y el UID de los datos de supresión pendientes en los escenarios operativos de supresión permanente o de supresión no permanente. Lo que antecede es un modo de puesta en práctica de la orden Delete, esto es, la orden incluye tres clases de información de tipo, UID e identificador utilizado para realizar la supresión permanente o la supresión no permanente; por supuesto, existen otros modos de puesta en práctica, a modo de ejemplo, la orden Delete está dividida en dos órdenes, una es una orden P-Delete y la otra es una orden NP-Delete, por lo que cada orden de supresión solamente necesita incluir el tipo y el número de los datos de supresión pendientes.

Por último, todas las órdenes de sincronización construidas, utilizadas para realizar la supresión, se envían al servidor.

El proceso de realizar la operación de sincronización después de que el servidor reciba la orden Delete anterior se describe a continuación.

Después de recibir la orden de sincronización que indica la supresión del elemento de datos D1, el servidor determina que el tipo de datos a sincronizarse es un elemento de datos Item en función del campo Meta recibido de la orden de sincronización y determina que el número de datos a sincronizarse en el cliente es 2001 en función del campo LUID y determina que la supresión es la supresión permanente y en adelante, el número de datos de supresión pendiente en el servidor local, a modo de ejemplo: 200001, se adquiere desde la tabla configurada de relación de correspondencia entre LUID y GUID, el elemento de datos correspondiente al número local de datos a sincronizarse se suprime respectivamente, esto es, el elemento de datos completo numerado 200001 se suprime, en la Tabla de Elementos de Datos local y en la Tabla de Índices. Al mismo tiempo, se suprimen los datos D1 en la base de datos local.

Después de recibir la orden de sincronización Delete utilizada para realizar la supresión del elemento de datos D2, el servidor suprime el elemento de datos completo de la tabla de datos correspondiente, el método de suprimir D2 es el mismo que para suprimir D1, por lo que no se describirá aquí con más detalle.

Después de recibir la orden de sincronización Delete utilizada para realizar la supresión del elemento de datos D3, el servidor determina que el tipo de datos a sincronizarse es un elemento de datos Item en función del campo Meta recibido de la orden de sincronización Delete y determina que el número de datos a sincronizarse en el cliente es 2003 en función del campo LUID y determina que la supresión es una supresión no permanente, y en adelante, el número de datos de la supresión pendiente en el servidor local, a modo de ejemplo: 200003, se adquiere a partir de la tabla configurada de la relación de correspondencia entre LUID y GUID, con lo que se solamente se suprime el elemento de datos correspondiente al número de datos locales a sincronizarse, esto es, el elemento de datos completo numerado 200003 se suprime y los datos D3 no se suprimen en la base de datos local, en la Tabla de Índices.

Puede deducirse que el método de las formas de realización de esta invención solamente necesita enviar un

identificador entre un cliente y un servidor y no necesitan enviar el contenido de datos detallado cuando se realiza la operación de sincronización de supresión, de modo que los recursos limitados se ahorran en la medida máxima.

5 En la quinta forma de realización, el usuario suprime la carpeta "bless" completa. Equivale a suprimir todas las sub-carpetas y los elementos de datos de la carpeta "bless". En esta forma de realización, D1 y D2 solamente existen bajo la carpeta "bless", D3 existe bajo la carpeta "bless" y "joke" y esta forma de realización solamente es la descripción de la supresión de la carpeta.

10 Según se ilustra en las Figuras 7a y 7b, la Figura 7a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario utilizando la quinta forma de realización de esta invención, en donde la trama cuadrada indica una carpeta Folder, el círculo indica un elemento de datos Item; la línea continua indica que el estado operativo es Existente, la línea de puntos indica que el estado operativo es Delete. La Figura 7b es un diagrama esquemático que ilustra la memorización de datos de un cliente que utiliza la quinta forma de realización de esta invención. La Tabla de Elementos de Datos, la Tabla de Carpetas y la Tabla de Índices se memorizan en un cliente. El estado operativo de datos en cada tabla tiene el reflejo correspondiente en la Figura 7b.

15 Cuando el usuario demanda sincronizar la carpeta raíz, el cliente crea la orden de sincronización siguiente de forma ordenada:

20 En primer lugar, el cliente comprueba todos los estados operativos de sub-carpetas bajo la carpeta raíz en conformidad con la orden para sincronizar la carpeta raíz recibida desde el usuario. En esta forma de realización, el estado operativo denominado "bless" se determina como D desde la Tabla de Carpetas, el cliente realizará, además, las etapas de: determinar si el elemento de datos de la carpeta de supresión pendiente y el elemento de datos de la sub-carpeta de la carpeta solamente existen bajo la carpeta de supresión pendiente; si es así, se construye una
25 orden de sincronización Delete y se incluye la información que indica la supresión permanente en la orden de sincronización Delete; de no ser así, se construye, respectivamente, una orden de sincronización Delete en relación con cada elemento de datos y carpeta y se incluye la información que indica que la supresión permanente en la orden de sincronización Delete correspondiente al elemento de datos o la carpeta solamente existe bajo la carpeta de supresión pendiente y se incluye la información que indica la supresión no permanente en la orden de
30 sincronización Delete correspondiente al elemento de datos o carpeta indicando que no solamente existe bajo la carpeta de supresión pendiente. Es decir, si un elemento de datos o carpeta todavía existe bajo otras carpetas (en este caso, otras carpetas no incluyen la sub-carpeta "bless") y la información de supresión no permanente está incluida en la orden de sincronización Delete correspondiente a dichos datos, si ninguno de dichos datos y la información de supresión permanente se incluye en la orden de sincronización Delete correspondiente. En adelante,
35 todas las órdenes de sincronización Delete construidas se envían al servidor. En este caso, una orden de supresión se construye respectivamente en relación con cada elemento de datos y carpeta, que realmente es la sincronización recursiva.

40 El proceso de realizar la operación de sincronización después de que el servidor reciba la orden de sincronización Delete anterior se describe a continuación.

45 Si la orden recibida por el servidor es la orden de sincronización Delete en relación con el elemento de datos, el modo de procesamiento es el mismo que el de la cuarta forma de realización, por lo que no se describirá aquí con más detalle.

50 Si la orden recibida por el servidor es la orden de sincronización Delete en relación con la carpeta, el número de datos de supresión pendientes en el servidor local se adquiere a partir de la tabla configurada de relación de correspondencia entre LUID y GUID, por lo que, entonces, sin importar la supresión permanente o la supresión no permanente, se suprimirá el elemento de datos correspondiente al número de datos locales a sincronizarse a partir de la Tabla de Carpetas configurada local.

55 Conviene señalar, en relación con la operación de supresión de carpeta que, como un iniciador de sincronización, cuando un cliente suprime una carpeta, a modo de ejemplo, suprime la carpeta "bless", solamente puede construir una orden de sincronización Delete en relación con la carpeta y otras operaciones realizadas por ella, a modo de ejemplo, "determinación de si el elemento de datos de la carpeta de supresión pendiente y el elemento de datos de la sub-carpeta de la carpeta solamente existen bajo la carpeta de supresión pendiente" y así sucesivamente, todas ellas realizadas por el servidor con el fin de simplificar la operación del cliente. Por supuesto, también se realiza en
60 viceversa.

60 En las aplicaciones reales, la cuarta forma de realización se suele utilizar incorporada con la quinta forma de realización.

65 Además, para la operación de supresión, después de que el servidor acabe la operación de sincronización, el cliente suprimirá el elemento de datos de su propia tabla de datos correspondiente.

En la sexta forma de realización, el usuario copia el elemento de datos "M1" desde la carpeta "música" a la carpeta

“favorito” y copia la carpeta “mp3” a la carpeta “favorito”.

Según se ilustra en las Figuras 8a y 8b, la Figura 8a es un diagrama esquemático que ilustra la estructura de datos definida por el usuario utilizando la sexta forma de realización de esta invención, en donde la trama cuadrada indica una carpeta Folder, el círculo indica un elemento de datos Item; la línea continua indica que el estado operativo es Existente, la línea gruesa indica que el estado operativo es Copy. La Figura 8b es un diagrama esquemático que ilustra la memorización de datos de un cliente utilizando la sexta forma de realización de esta invención. La Tabla de Elementos de Datos, la Tabla de Carpetas y la Tabla de Índices se memorizan en un cliente. El estado operativo de datos añadidos en cada tabla tiene el reflejo correspondiente en la Figura 3b.

Conviene señalar que, en las formas de realización anteriores, una tabla de relación de correspondencia entre el número en el cliente y el número en el servidor, a modo de ejemplo: una tabla de relación de correspondencia entre LUID y GUID, la finalidad de tener el mecanismo de LUID-GUID es considerar que el hecho de que existan dispositivos con capacidad limitada que no puedan soportar un identificador UID largo. En el caso de que todos los dispositivos puedan soportar el UID largo, el mecanismo de LUID-GUID evoluciona como el cliente y el servidor utilizando el mismo UID generado por el cliente o el servidor y luego, el proceso puede realizarse directamente utilizando el mismo UID y el mapeado de LUID-GUID ya no es necesario. De este modo, en este caso, la puesta en práctica de las formas de realización de esta invención no está limitada.

Cuando el usuario necesita sincronizar la carpeta raíz, la operación del cliente y del servidor es la misma que en la primera forma de realización. La diferencia de ellas es: en la primera forma de realización, el cliente envía una orden de sincronización Add en relación con cada elemento de datos y carpeta y en esta forma de realización, si el cliente envía una orden de sincronización Copy en relación con una carpeta (la orden de sincronización Copy es también una sub-orden de la orden Sync que se utiliza para transmitir datos de copias pendientes), no hay necesidad de enviar órdenes de sincronización Copy en relación con sub-carpetas y elementos de datos bajo la carpeta, por lo que se ahorra la cantidad de transmisión de datos con lo que se disminuye, todavía más, dicha cantidad y se ahorran recursos de la red. El proceso del servidor, en esta forma de realización, es el mismo que la primera forma de realización y se realiza también, uno por uno, en relación con cada carpeta y elemento de datos.

Además, cuando se realiza la sincronización de Copy, el usuario determina, en función de la demanda, si necesita copiarse una copia de datos reales y si es así, la operación de sincronización de datos del lado que realiza la operación de sincronización incluye, además: copia de los datos en la base de datos local y la adición del elemento de datos correspondiente en la Tabla de Carpetas de datos configurada local.

Si la operación de modificación del cliente está en conflicto con la del servidor, a modo de ejemplo: adición, sustitución o supresión de algunas entradas en la carpeta desplazada, las formas de realización de esta invención, pueden garantizar la sincronización de datos absoluta entre el cliente y el servidor extendiendo el mecanismo de resolución de conflictos existente. La puesta en práctica detallada es: extensión del mecanismo de resolución de conflictos existente para añadir una nueva resolución denominada “Win-Win” para la fusión del contenido de datos del cliente y del servidor sobre la base de alguna configuración y asegurar los mismos datos absolutos entre el cliente y el servidor por el modo ‘Win-Win’ además de la resolución de ‘Cliente-Win’ o ‘Servidor-Win’ actual. Cuando la operación de datos del cliente está en conflicto con la del servidor, el servidor detecta el conflicto y realiza la resolución ‘Win-Win’. A continuación, el servidor reenvía la operación de sincronización en función del resultado de resolución del conflicto y el cliente realiza la operación de sincronización en función de la operación de datos del servidor, incluyendo la operación de datos: operación de adición, operación de sustitución, operación de desplazamiento, operación de supresión, operación de copia o cualquiera de sus combinaciones. A modo de ejemplo, el usuario desplaza la carpeta A y permite que sea una sub-carpeta de la carpeta B en el cliente y añade un elemento de datos en la carpeta A en el servidor, este caso, el servidor desplazará la carpeta A y permitirá que sea una sub-carpeta de la carpeta B y el cliente añade también un elemento de datos en la carpeta A con el fin de garantizar los datos idénticos absolutos entre el cliente y el servidor.

Las formas de realización de esta invención dan a conocer, además, un sistema para la sincronización de datos basado en el método anterior, incluyendo dicho sistema: un cliente y un servidor, que se comunican entre sí intercambiando órdenes de sincronización. El cliente y el servidor se utilizan, además, para determinar una carpeta a sincronizarse y para realizar la sincronización de datos de la carpeta a sincronizarse.

La Figura 9 es un diagrama esquemático que ilustra la arquitectura del sistema para la sincronización de datos en conformidad con las formas de realización de esta invención. Según se ilustra en la Figura 9, el cliente incluye un primer módulo para procesar una dirección de nodo y un primer módulo para la sincronización de datos y el servidor incluye un segundo módulo para procesar la dirección de nodo y un segundo módulo para la sincronización de datos.

En el cliente, el primer módulo para procesar la dirección de nodos se utiliza para determinar y transmitir la dirección de un nodo a sincronizarse al primer módulo para sincronización de datos, la recepción de la dirección del nodo a sincronizarse desde el primer módulo para sincronización de datos, la determinación del nodo a sincronizarse en función de la dirección recibida y proporcionar sincronización entre el cliente y servidor para los datos bajo el nodo a sincronizarse; el primer módulo para la sincronización de datos se utiliza para recibir la dirección del nodo a

5 sincronizarse desde el primer módulo para procesar la dirección del nodo, construir una orden de sincronización que contiene la dirección del nodo a sincronizarse, la transmisión de la orden de sincronización al servidor, la recepción de una orden de sincronización desde el servidor, la adquisición de la dirección del nodo a sincronizarse desde la orden de sincronización y transmitir la dirección adquirida al primer módulo para procesar la dirección del nodo. El primer módulo para procesar la dirección del nodo puede utilizar, además, para recibir la dirección del nodo a sincronizarse introducida por una orden de configuración desde el usuario de modo que el usuario pueda establecer la dirección del nodo a sincronizarse para el cliente.

10 En el servidor, el segundo módulo para procesar la dirección del nodo se utiliza para determinar y transmitir la dirección del nodo a sincronizarse con el segundo módulo para sincronización de datos, la recepción de la dirección del nodo a sincronizarse desde el segundo módulo para la sincronización de datos, la determinación del nodo actualmente a sincronizarse en función de la dirección recibida y proporcionar la sincronización entre el cliente y el servidor para los datos bajo el nodo a sincronizarse; el segundo módulo para sincronización de datos se utiliza para recibir la dirección del nodo a sincronizarse desde el segundo módulo para procesar la dirección del nodo, construir una orden de sincronización que contenga la dirección del nodo a sincronizarse, la transmisión de la orden de sincronización al servidor, la recepción de una orden de sincronización desde el servidor y la adquisición de la dirección del nodo a sincronizarse a partir de la orden de sincronización mediante análisis y la transmisión de la dirección adquirida al segundo módulo para procesar la dirección del nodo.

20 Puede deducirse del sistema anterior para la sincronización de datos que las formas de realización de esta invención dan a conocer, además, un cliente para sincronización de datos y un servidor para sincronización de datos. El principio para poner en práctica el cliente y el servidor es el mismo que para el cliente y el servidor en el sistema anterior, por lo que el principio operativo y su estructura interna no se describen aquí con más detalle, pero deben estar cubiertos bajo el alcance de protección de esta invención.

25 Lo que antecede es solamente formas de realización preferidas de esta invención y no se utilizan para limitar esta invención. Cualquier modificación, sustitución equivalente o mejora realizada bajo el principio de esta invención debe incluirse en el alcance de protección de la presente invención.

30

REIVINDICACIONES

1. Un método de sincronización de una pluralidad de dispositivos, cuyo método comprende:

5 obtener, por un primer dispositivo, la dirección de un nodo informacional a sincronizarse entre una pluralidad de nodos informacionales dispuestos en una estructura arborescente jerárquica en un segundo dispositivo, en donde dicha dirección comprende una posición jerárquica que se utiliza para localizar dicho nodo informacional entre la estructura arborescente y en donde el nodo informacional es un elemento de datos o una carpeta en una base de datos;

10 estando el método caracterizado por cuanto que comprende, además:
enviar, por el primer dispositivo, al segundo dispositivo, una orden de sincronización durante una fase de inicialización de sincronización que permite diseñar el nodo informacional a sincronizarse, en donde la orden de sincronización comprende la dirección del nodo informacional;

15 utilizar, por el primer dispositivo, una orden de sincronización, durante una fase de sincronización, para sincronizar el nodo informacional designado, en donde la orden de sincronización contiene datos que pertenecen a la gama de direcciones indicadas de dicha pluralidad de nodos informacionales, pero no la totalidad de los datos de la base de datos completa.

20 **2.** El método según la reivindicación 1, en donde dicha etapa consiste en obtener dicha dirección que comprende:
25 recibir, por dicho primer dispositivo, una descripción de dicha estructura arborescente jerárquica desde dicho segundo dispositivo y
determinar, por dicho primer dispositivo, dicha dirección a partir de dicha descripción.

30 **3.** El método según la reivindicación 1, en donde dicha obtención de dicha dirección comprende:
recibir, por dicho primer dispositivo, dicha dirección desde dicho segundo dispositivo.

4. El método según la reivindicación 3, que comprende, además:
35 enviar una orden, por dicho primer dispositivo, para recuperar dicha dirección desde dicho segundo dispositivo.

5. El método según la reivindicación 1, en donde dicha obtención de dicha dirección comprende:
40 designar, por dicho primer dispositivo, una dirección como dicha dirección.

6. El método según la reivindicación 5, en donde dicha obtención de dicha dirección comprende:
crear, si dicha dirección no existe, por el segundo dispositivo, un nodo en dicha dirección.

45 **7.** El método según la reivindicación 1, en donde dicha obtención de dicha dirección comprende:
determinar, por dicho primer dispositivo, dicha dirección a partir de información introducida por un usuario.

50 **8.** El método según la reivindicación 1, en donde dicha obtención comprende:
determinar dicha dirección por referencia a una dirección designada con anterioridad.

9. El método según la reivindicación 1, en donde la orden de sincronización es una orden de Alerta.

55 **10.** El método según la reivindicación 1, en donde la orden de sincronización indica que dicha sincronización es una sincronización recursiva.

11. El método según la reivindicación 1, en donde dicha orden de sincronización comprende múltiples direcciones si hay múltiples nodos informacionales que han de sincronizarse.

60 **12.** El método según la reivindicación 1, en donde utilizar una orden de sincronización, durante una fase de sincronización, para sincronizar el nodo informacional, comprende:

65 enviar, por el primer dispositivo, una segunda orden de sincronización durante una fase de sincronización al segundo dispositivo, en donde la segunda orden de sincronización comprende el tipo de operación para el nodo informacional;

realizar, por el segundo dispositivo, una operación de sincronización en cada uno de los nodos informacionales después de recibir la segunda orden de sincronización, en donde la operación de sincronización se designa por el tipo de operación incluida en la segunda orden de sincronización.

- 5 **13.** El método según la reivindicación 12, en donde el tipo de operación es una de entre Añadir, Sustituir, Suprimir y Desplazar.
- 14.** El método según la reivindicación 13, en donde la operación de Suprimir es una de entre una supresión permanente o una supresión no permanente.
- 10 **15.** El método según la reivindicación 1, en donde la dirección es una entre un Identificador de Recurso Uniforme (URI), un Identificador Único (UID) y una condición de filtro.
- 16.** Un aparato para sincronización, que comprende:
- 15 un primer módulo, adaptado para obtener la dirección de un nodo informacional a sincronizarse entre una pluralidad de nodos informacionales dispuestos en una estructura arborescente jerárquica en un segundo dispositivo y para proporcionar la dirección a un segundo módulo, en donde dicha dirección comprende una posición jerárquica que se utiliza para localizar dicho nodo informacional entre la estructura arborescente;
- 20 caracterizado por cuanto que comprende, además:
- el segundo módulo, adaptado para enviar una orden de sincronización, durante una fase de inicialización de sincronización, al segundo dispositivo para designar el nodo informacional a sincronizarse, y para utilizar una
- 25 segunda orden de sincronización para sincronizar el nodo informacional durante una fase de sincronización, en donde la primera orden de sincronización comprende la dirección del nodo informacional, el nodo informacional es un elemento de datos o una carpeta en una base de datos y la segunda orden de sincronización contiene datos que pertenecen a la gama de direcciones indicadas de dicha pluralidad de nodos informacionales obtenidas por el primer módulo, pero no la totalidad de los datos de la base de datos completa.
- 30 **17.** El aparato según la reivindicación 16, en donde dicho primer módulo está adaptado para obtener dicha dirección recibiendo, desde dicho segundo dispositivo, una descripción de dicha estructura arborescente jerárquica y para determinar dicha dirección a partir de dicha descripción.
- 35 **18.** El aparato según la reivindicación 16, en donde dicho primer módulo está adaptado para obtener dicha dirección recibiendo, desde dicho segundo dispositivo, dicha dirección.
- 19.** El aparato según la reivindicación 16, en donde dicho primer módulo está adaptado para obtener dicha dirección designando una dirección como dicha dirección.
- 40 **20.** El aparato según la reivindicación 16, en donde dicho primer módulo está adaptado para obtener dicha dirección determinando una dirección en función de la información introducida por un usuario.
- 21.** El aparato según la reivindicación 16, en donde dicho primer módulo está adaptado para obtener dicha dirección haciendo referencia a una dirección designada con anterioridad.
- 45 **22.** El aparato según la reivindicación 16, en donde dicha orden de sincronización es una orden de Alerta.
- 23.** El aparato según la reivindicación 16, en donde el segundo módulo está adaptado, además, para enviar una
- 50 segunda orden de sincronización durante una fase de sincronización al segundo dispositivo, en donde la segunda orden de sincronización comprende un tipo de operación para el nodo informacional.

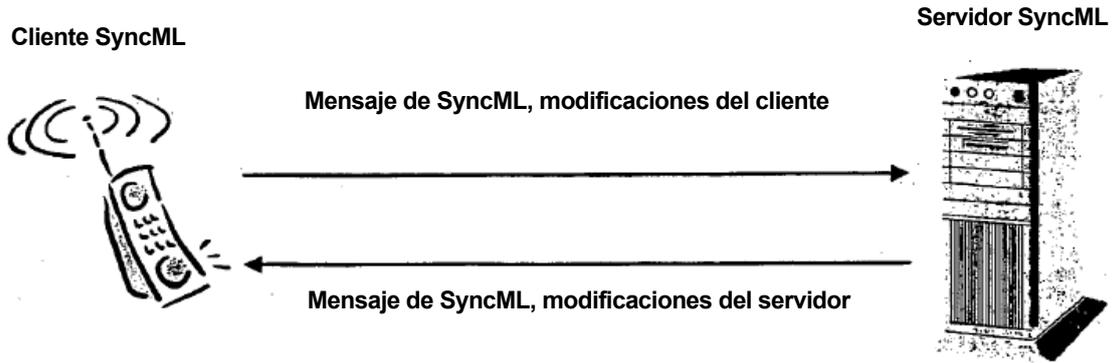


Fig. 1

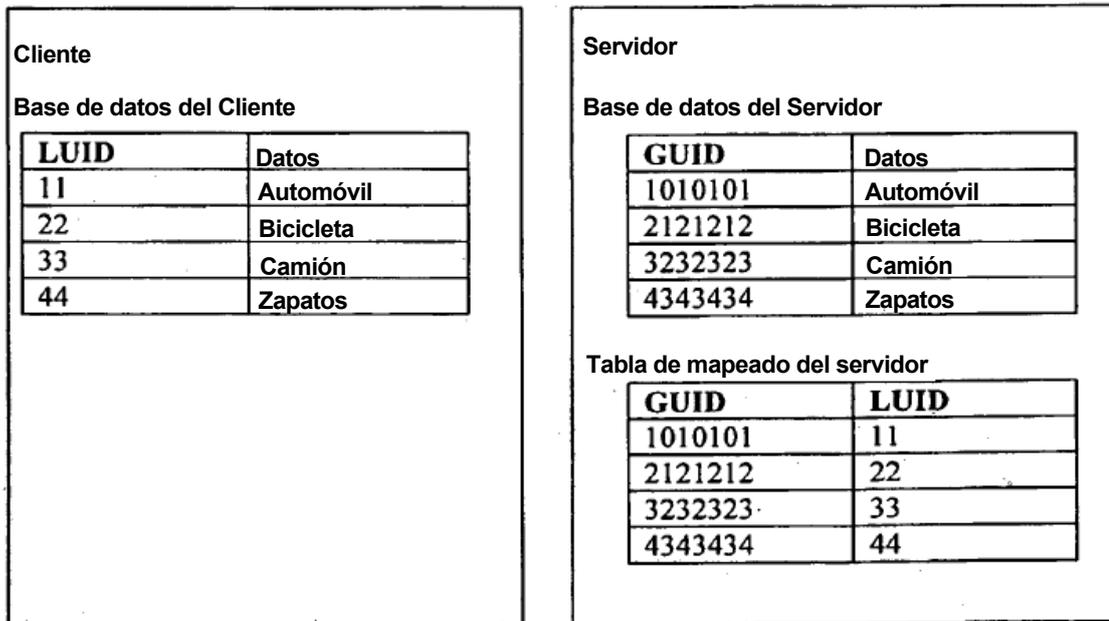


Fig. 2

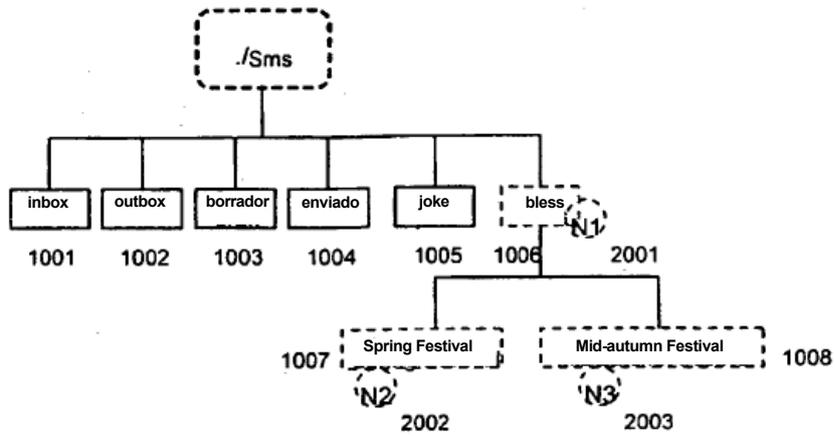


Fig. 3a

Elementos LUID	Datos
2001	N1
2002	N2
2003	N3

Carpeta LUID	Denominación	Parent Source	estado
1007	Spring Festival	1006	N
1008	Mid-autumn Festival	1006	N
1006	bless	/	N
1005	joke	/	E

Elemento LUID	Parent Source	estado
2001	1006	N
2002	1007	N
2003	1008	N

Fig. 3b

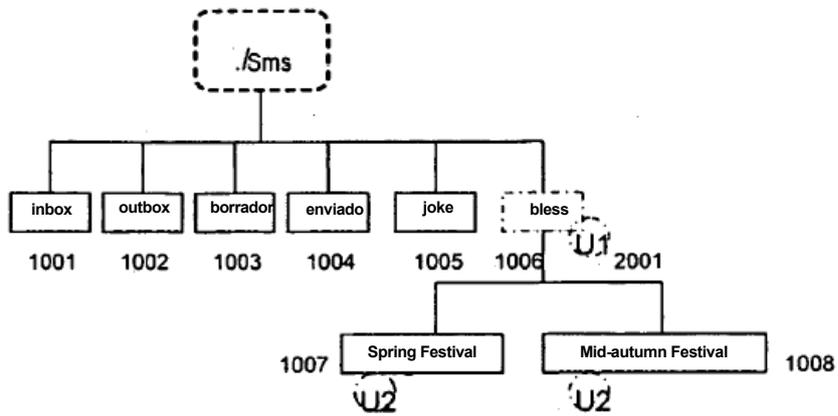


Fig. 4a

Elementos LUID	Datos
...	...
2001	U1
2002	U2

Carpeta LUID	Denominación	Parent Source	estado
...
1007	Spring Festival	1006	E
1008	Mid-autumn Festival	1006	E
1006	bless	/	U
1005	joke	/	E

Elemento LUID	Parent Source	estado
...
2001	1006	U
2002	1007	U
2002	1008	U

Fig. 4b

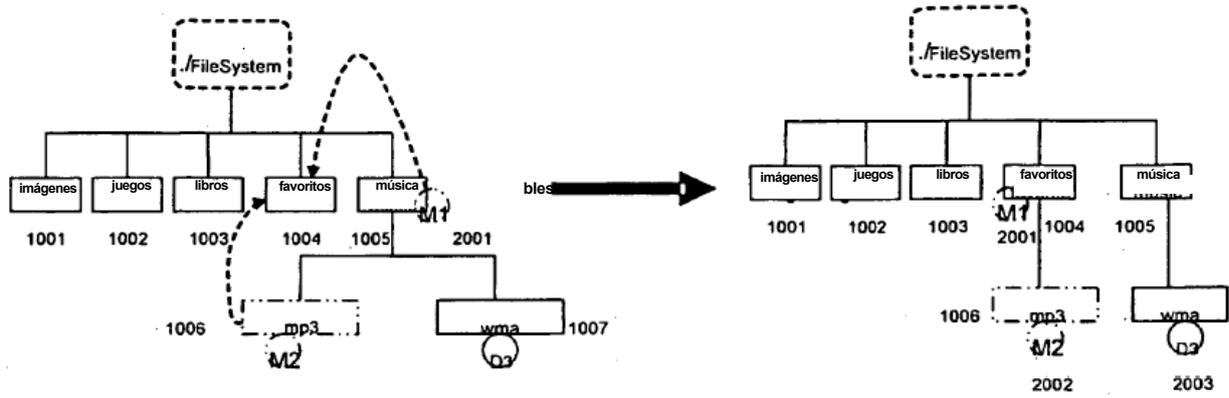


Fig. 5a

Elementos LUID	Datos
...	...
2001	M1
2002	M2
2003	M3

Carpeta LUID	Denominación	Parent Source	estado
...
1006	mp3	1005> 1004	M
1007	wma	1005	E
1004	favorito	/	E
1005	música	/	E

Elemento LUID	Parent Source	estado
...
2001	1005> 1004	M
2002	1006	E
2003	1007	E

Fig. 5b

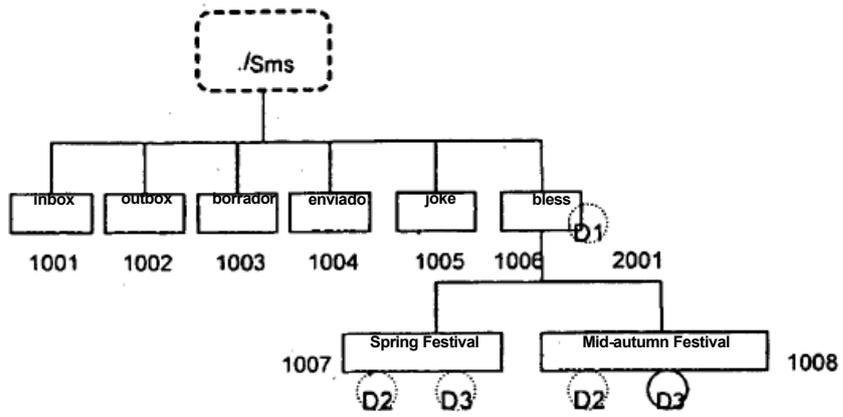


Fig. 6a

LUID	Datos
2001	D1
2002	D2
2003	D3

Carpeta LUID	Denominación	Parent Source	estado
1007	Spring Festival	1006	E
1008	Mid-autumn Festival	1006	E
1006	bless	/	E
1005	joke	/	E

Elemento LUID	Parent Source	estado
2001	1006	P-D
2002	1007	P-D
2002	1008	P-D
2003	1007	NP-D
2003	1008	E

Fig. 6b

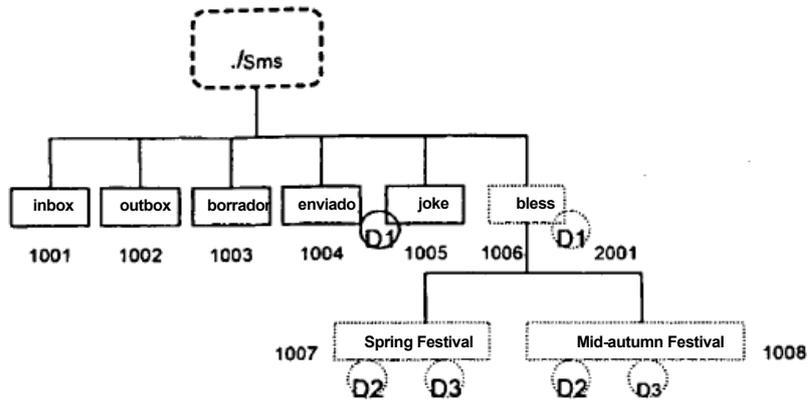


Fig. 7a

Elementos LUID	Datos
2001	D1
2002	D2
2003	D3

Carpeta LUID	Denominación	Parent Source	estado
1007	Spring Festival	1006	E
1008	Mid-autumn Festival	1006	E
1006	bless	/	D
1005	joke	/	E

Elemento LUID	Parent Source	estado
2001	1006	NPD
2002	1007	P-D
2002	1008	P-D
2003	1007	P-D
2003	1008	P-D
2001	1005	E

Fig. 7b

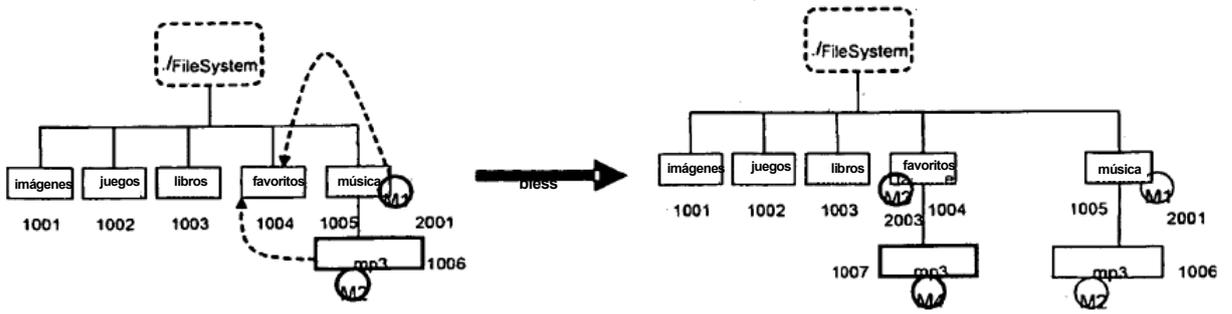


Fig. 8a

Elementos LUID	Datos
2001	M1
2002	M2
2003	M3
2004	M4

Carpeta LUID	Denominación	Parent Source	estado
1006	mp3	1005	E
1004	favoritos	/	E
1005	música	/	E
1007	mp3	1004	C

Elemento LUID	Parent Source	estado
2001	1005	E
2002	1006	E
2003	1004	C
2004	1007	C

Fig. 8b

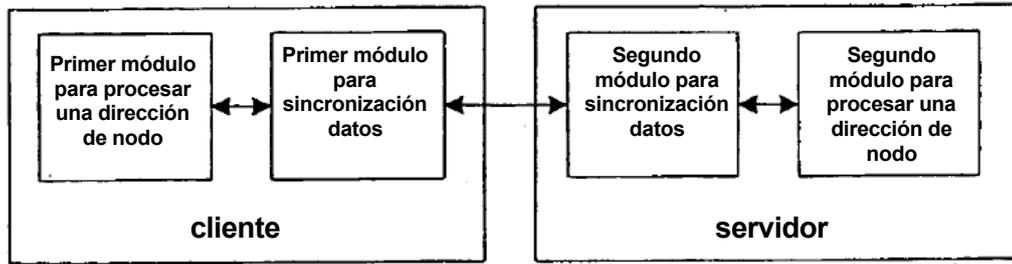


Fig. 9