



Industria y Comercio
SUPERINTENDENCIA

SOLICITUD
PATENTE DE INVENCION

21. EXPEDIENTE N°

04-16523

54. TÍTULO

METODO PARA EL MANEJO DE REPLICACION DE ARCHIVOS EN
APLICACIONES

51. CLASIFICACIÓN INTERNACIONAL

7. E06F 17/24

71. SOLICITANTE

MICROSOFT CORPORATION

DOMICILIO

REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA

74. APODERADO

DARIO CARDENAS NAVAS

22. BOGOTÁ, D. C.

2004 - II - 25

202038

PETITORIO

Industria y Comercio
SUPERINTENDENCIA

04 016523 00

SUPERINTENDENCIA Y COMERCIO
Radicación : 04016523 00000000 Folios: 30
Fecha (AMB): 2004-02-25 14:24:58
Trámite : 002 PRETENCIONES 1 RESOLUCION/ 411 PRESENTACION
Dependencia: 2000 DIVISION DE NUEVAS EMPRESAS

FORMULARIO UNICO DE SOLICITUD DE PATENTE 2000-3

1 SOLICITUD DE:			
☒ Patente de Invención		☐ Patente de Modelo de Utilidad	
2 SOLICITANTE (71)	Nombre: MICROSOFT CORPORATION Dirección: One Microsoft Way Redmond, Washington 98052, Estados Unidos de America Teléfono: 3123600 Fax: 3122410 E-Mail: general@cardenasycardenas.com Domicilio: REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA	IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número _____	
3 REPRESENTANTE O APODERADO (74)	Nombre: DARIO CARDENAS NAVAS Dirección: Carrera 7 No 71-52 Torre B Piso 8 Teléfono: 3123600 Fax: 3122410 E-Mail: general@cardenasycardenas.com	IDENTIFICACION C.C. ☒ NIT ☐ C.E. ☐ Otro ☐ Número 17.066.629 BTA TP 1.250 C.S.J.	
4 INVENTOR(ES) (72)	Nombre: - Noah Edelstein - Hani Saliba - Hai Liu - Steven James Rayson Dirección: 380 North 26 th Street Seattle, Washington 98103 524 12 th Avenue East # 302 Seattle, Washington 98102 4800 156 th Avenue Apartment # 280 Redmond, Washington 98052 609 Federal Avenue East Seattle, Washington 98102 Teléfono: 3123600 Fax: 3123600 E-Mail: general@cardenasycardenas.com Domicilio: - WASHINGTON, ESTADOS UNIDOS DE AMERICA - WASHINGTON, ESTADOS UNIDOS DE AMERICA - WASHINGTON, ESTADOS UNIDOS DE AMERICA - WASHINGTON, ESTADOS UNIDOS DE AMERICA	IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número _____	
5 Título(54): METODO PARA EL MANEJO DE REPLICACION DE ARCHIVOS EN APLICACIONES			
6 Prioridad SI ☒ NO ☐		(33) País de Origen ESTADOS UNIDOS DE AMERICA	(32) Fecha Febrero 28, 2003
		(31) N° de Solicitud 10/376,738	
7 Para publicar a partir de la fecha de la presente solicitud a los: 6 meses ☐ 12 meses ☐ 18 meses ☐ Otro ☐			
8 Comprobante de Pago No.:		04-10.513 / 04-10.507	Fecha: Febrero 10, 2004

9

ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud.
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☒ Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (No 04-10.513)
- ☒ Poderes, si fuere el caso. (Protocolo No 16.069)
- ☒ Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- ☐ Copia certificada de la primera solicitud, si se reivindica prioridad.
- ☒ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☒ Documento de cesión del inventor al solicitante o a su causante.
- ☒ Descripción de la Invención.
- ☒ Una o más reivindicaciones.
- ☒ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☒ Arte final 12x12 y 6x6 cm.
- ☒ Resumen y extracto para publicación.

10

NOMBRE:

DARIO CARDENAS NAVAS

FIRMA:

G.C. 17.066.629 BTA

TP. 1.250 C.SJ.

SUMATORIA Y CORRECCION
 Radicacion : 00016321
 Fecha (MM/AA) : 2004-02-25 14:34:58
 Transite : 002 PATENTES O I REGISTRO, al M. EXAM.
 Dependencia: 2029 DIVISION DE NUEVAS CREACIONES

NIT : 000176089-2

RECIBO OFICIAL DE CAJA : 04 - 10,307
 FECHA : FEBRERO 10 DE 2004

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I. CONSIGNACION		BANCO POPULAR	030-00110-6	21079192	10/02/2004	3,431,000.00

***** CONCEPTO *****

CANT. RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	30003-01-01 SOLICITUDES	
1	TRAMITES DE SOL. DE PATENTE DE IN	422,000.00
	TOTAL :	422,000.00

BON: CUATROCIENTOS VEINTIDOS MIL PESOS

Método para el manejo
 de replicación de archivos

en Aplicaciones

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

SUPERINTENDENCIA Y COMERCIO
Radicación : 04016523 00000000 folios: 98
Fecha (DM): 2004-02-25 14:04:38
Trámite : 002 PATENTES 1 REGISTRO/ 41 PRESENCIA:
Dependencia: 0020 DIVISION DE NUEVAS CREACIONES

NIT : 000176087-2

RECIBO OFICIAL DE CAJA : 04 - 10,513
FECHA : FEBRERO 10 DE 2004

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	21079195	10/02/2004	3,451,000.00

***** CONCEPTO *****

CANT. REMISTICO	CONCEPTO	TOTAL CONCEPTO
1 50005-01-01 SOLICITUDES	02 INVOCACION DE UNA PRIORIDAD EN MI	118,000.00
	TOTAL :	118,000.00

SON: CIENTO DIEZ Y OCHO MIL PESOS

Método para el manejo de
Replicación de archivos
en Aplicaciones.

RESPONSABLE : _____

CARDENAS Y CARDENAS
ABOGADOS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

**METODO PARA EL MANEJO DE REPLICACIÓN DE ARCHIVOS EN
APLICACIONES**

SOLICITUDES RELACIONADAS

Esta solicitud de patente se relaciona con una Solicitud Norteamericana de Patente No.AA/BBB,CCC, co-pendiente de aprobación y cedida en forma común, presentada en febrero 27 de 2003 bajo el número de sumario de abogado MSFT-1586 y titulada "Método y Sistema para el Manejo del Estado de Múltiples Archivos."

CAMPO DE LA INVENCION

La presente invención se relaciona con métodos para manejar múltiples versiones de un archivo electrónico. Más particularmente, la presente invención se relaciona con métodos para usar una funcionalidad de aplicación relevante para reconciliar versiones de un archivo durante un evento de sincronización.

ANTECEDENTES DE LA INVENCION

Es una práctica común en un ambiente en red de computadores para que un usuario de computador, colabore sobre un documento electrónico con otro usuario en la red. Tal red puede ser una Red de Área Local (LAN por sus siglas en inglés), Red de Área Extendida (WAN por sus siglas en inglés), la Internet u otras del mismo tipo. En tal colaboración, cada usuario puede crear y editar un archivo electrónico usando una aplicación de programa tal como un programa de aplicación general, o un programa diseñado específicamente para el tipo de archivo que es el objeto de la colaboración, u otros del mismo tipo.

La colaboración puede hacerse posible manteniendo una versión maestra del archivo electrónico en una ubicación que sea accesible para todos los usuarios que estarán colaborando sobre el archivo. En un ambiente convencional de computadores en red, podrían existir un computador servidor y uno o más computadores usuarios. El servidor mantiene enlaces de comunicación entre, y es accesible a, cada computador usuario y por lo tanto puede almacenar y controlar la versión maestra del archivo que está siendo colaborado por lo usuarios. Cuando hay colaboración sobre un archivo electrónico, un usuario puede acceder primero

a la versión maestra del archivo sobre el servidor, y luego copiar el archivo al computador del usuario. El usuario puede entonces trabajar sobre el archivo alterándolo en cualquier forma que se halle disponible para el usuario por la aplicación. Por ejemplo, si la aplicación es un procesador de palabra, hoja de cálculo, base de datos, y/o similares, la aplicación puede proveer medios para realizar tareas tales como edición, copiado, borrado y/o similares. Al completar las modificaciones, el usuario puede salvar los cambios de la versión del archivo guardado en el computador del usuario, si éste desea conservar tales modificaciones del mismo. En otros sistemas, tales como en uno en donde no exista un computador servidor, la versión maestra puede ser almacenada en el computador del usuario.

Con el propósito de colaborar con otros usuarios a través del servidor, un usuario con una versión cambiada y almacenada en el computador del usuario, debe sincronizar la versión del archivo sobre el computador con la versión maestra del mismo sobre el servidor. La sincronización de una diversidad de versiones de un archivo electrónico, en su forma más simple, asegura que todas las versiones, una vez sincronizadas, son idénticas. De esta forma, el

usuario llevando a cabo la sincronización permitirá a otros usuarios que participan en la colaboración, acceder al nuevo archivo modificado. Convencionalmente, una sincronización es llevada a cabo por una función de sincronización. Una función de sincronización está típicamente separada de la aplicación, y en muchos casos es parte del sistema operativo del computador del usuario. La función de sincronización convencional realiza la sincronización a tiempos predeterminados, o a solicitud por parte del sistema operativo.

En una sincronización convencional, la función de sincronización comparará la versión del usuario del archivo con la versión maestra del mismo utilizando cualquier información referente a las copias que la función sea capaz de abarcar. Típicamente, tal información está limitada a la fecha y/o hora de la última actualización de la última versión. Si la fecha y/o hora son idénticos, la función de sincronización determina que ambas versiones son idénticas, y por lo tanto sincronizadas, y que no se requiere ninguna acción adicional. Si existe una discrepancia entre la fecha y/o la hora, de cualquier modo, la función de sincronización provee al usuario asociado con el computador que está

manejando el evento de la sincronización con una interfaz de usuario. La interfaz de usuario provee convencionalmente al usuario ~~con~~ una breve notificación de que hay una discrepancia, la fecha y/o la hora de esa última actualización para cada versión del archivo, y una solicitud de una instrucción por parte del usuario de cual copia del archivo el usuario desea conservar. Si el usuario elige conservar la versión de usuario, la función de sincronización escribe sobre la versión maestra la versión del usuario, de tal forma que convierte la versión del usuario en la nueva versión maestra sobre el servidor. Si el usuario desea conservar la versión maestra, ninguna acción adicional es tomada, o la versión del usuario es sobrescrita con la versión maestra. En algunas interfaces de usuario, se le presenta al usuario una opción para abortar la sincronización, en vez de escoger cuál versión será la versión maestra.

La sincronización convencional como se discutió antes, tiene desventajas significativas. Debido a que una función de sincronización convencional tiene una funcionalidad limitada ~~con~~ respecto a un archivo electrónico, una interfaz de usuario generada por la función convencional presenta solamente una cantidad

limitada de información a un usuario cuando lo incita a seleccionar un curso de acción. Esa cantidad limitada de información puede por lo tanto causar que ese usuario tome una decisión no uniforme y posiblemente incorrecta acerca de cual curso de acción debe seguir. Por ejemplo, un primer usuario puede alterar una versión de usuario previamente sincronizada del archivo, y luego, en un determinado momento, la función de sincronización convencional inicia un evento de sincronización. Cuando es notificado por la interfaz de usuario convencional, el usuario puede ver que la versión de usuario fue actualizada después que la versión maestra y por lo tanto decide sobrescribir dicha versión maestra. Sin embargo, la información presentada por la interfaz de usuario convencional no informa al usuario si la versión maestra ha permanecido inalterada desde que el usuario comenzó a alterar dicha versión de usuario. Esto conduce a una situación problemática, donde un segundo usuario puede haber hecho alteraciones a otra versión de usuario, y luego sincronizó esa versión de usuario con la versión maestra antes de la primera sincronización de usuario. La primera versión de usuario es más reciente-habiendo sido actualizada posteriormente- pero la versión maestra ha sufrido alteraciones por parte del segundo usuario que no

7

11

estaba presente cuando el primer usuario copió originalmente el archivo del servidor. Como resultado, las alteraciones del segundo usuario a la versión maestra se pierden, una vez que el primer usuario complete la sincronización.

La anterior situación es causada por la carencia del rango completo de funcionalidad de la función de la sincronización convencional, inherente en la aplicación relevante al archivo, de forma tal que la función es incapaz de usar esa funcionalidad durante un evento de sincronización. Por lo tanto, la función solamente puede proveer al usuario una muy limitada cantidad de información (en relación con) las diferentes versiones del archivo electrónico, tales como, por ejemplo, la fecha y/o la hora de la última actualización de cada versión del archivo como se discutió antes.

Otra desventaja causada por la funcionalidad limitada de las funciones de sincronización convencional es la falta de múltiples vías de acción disponibles para el usuario durante una sincronización. Por ejemplo, en el ejemplo discutido antes, la interfaz de usuario solamente presenta tres opciones: actualizar la copia maestra con

la copia de usuario, actualizar la copia de usuario con la copia maestra, o abortar.

Una desventaja adicional de una función de sincronización convencional radica en una falta de integración de tal función con tal aplicación. Por ejemplo, y como se discutió antes, una función de sincronización convencional, es parte de un sistema operativo de un computador de usuario o de servidor. Por lo tanto, el proceso de sincronización, y cualquier interfaz de usuario aplicable, es conducida convencionalmente fuera de la experiencia de aplicación del usuario. Esto puede llevar a confusión entre usuarios quienes-mientras que tales usuarios pueden ser expertos con la aplicación-no son expertos con la función de un sistema operativo. Adicionalmente, y como un resultado de la falta de integración, una sincronización puede ser iniciada por la función de sincronización en un momento que no es óptimo para que tal evento tenga lugar. Una desventaja adicional de las funciones de sincronización convencionales es la incapacidad de tales funciones para ser usadas como un componente de una aplicación de programa. Por ejemplo, usando los métodos convencionales, un programador de computadores es incapaz de crear una primera aplicación.

de programa que use una función de sincronización que como la habilidad para usar una segunda aplicación apropiada para el archivo electrónico de reconciliar múltiples versiones del mismo. Por lo tanto, lo que se necesita es un medio eficiente de usar una funcionalidad de aplicación de programa que sea relevante para resolver conflictos entre las copias de un archivo durante una sincronización.

RESUMEN DE LA INVENCION

En vista de lo anterior, la presente invención provee métodos de integración de una interfaz de programa de aplicación con una aplicación de programa que permita a la aplicación solicitar y controlar un proceso de sincronización de archivo mientras mantiene en la aplicación la experiencia del usuario. La invención provee un método para manejar una primera y una segunda versión de un archivo electrónico. En el método, la primera y segunda versiones de un archivo electrónico son reconocidas y un evento de sincronización es iniciado usando una primera aplicación de programa operando sobre un primer computador. Una interfaz de programa de aplicación accesa la primera y segunda versiones,

compara la información asociada con cada una, genera un resultado y lo reporta a la primera aplicación de programa. Una segunda aplicación de programa genera una interfaz de usuario de acuerdo con una característica de la segunda aplicación de programa, donde el archivo electrónico está asociado con la segunda aplicación de programa, y la interfaz de usuario es mostrada en un dispositivo de pantalla.

La invención también provee para la segunda aplicación de programa, la generación de una opción de sincronización y presentar la opción con la interfaz de usuario y mostrar la interfaz de usuario en el dispositivo de pantalla. La segunda aplicación de programa puede también recibir una selección de opción y sincroniza en forma correspondiente la primera y segunda versiones del archivo electrónico. La invención además prevé que la primera y segunda aplicaciones de programa pueden ser la misma aplicación de programa. La aplicación puede proveer cualquier cantidad de información al usuario relativas a la primera y segunda versiones, y el usuario puede ingresar cualquier número o tipo de selecciones de sincronización. En una modalidad preferida, el usuario puede crear una tercera versión que incluya cualquier porción, o ninguna,

ya sea de la primera o de la segunda versión, o de ambas. La aplicación puede iniciar la sincronización en cualquier momento, tal como a solicitud del usuario. Rasgos adicionales y ventajas de la invención son expuestas en la descripción que viene a continuación.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

El anterior resumen, así como la siguiente descripción detallada de modalidades preferidas, son mejor entendidos cuando se leen en conjunto con los dibujos anexados. Para el propósito de ilustrar la invención, se muestran en los dibujos modalidades ejemplares de la invención; sin embargo, la invención no está limitada a los métodos específicos ni a los medios descritos. En los dibujos:

La Figura 1 es un diagrama de bloques mostrando una modalidad de computo ejemplar en la cual, aspectos de la invención pueden ser implementadas;

La Figura 2 es un diagrama funcional ilustrando un ejemplo de una relación entre un primer computador, una red, y un segundo computador;

La Figura 3 es una carta de flujo mostrando un proceso de sincronización de acuerdo con una modalidad de la presente invención; y

La Figura 4 es una carta de flujo ilustrando un método de iniciar un proceso de sincronización de acuerdo con una modalidad de la presente invención.

DESCRIPCIÓN DETALLADA DE MODALIDADES ILUSTRATIVAS

Vistazo General

La presente invención provee métodos para integrar una interfaz de programa de aplicación con una aplicación de programa que permite la aplicación para solicitar y controlar un proceso de sincronización de archivo mientras permanece en la aplicación la experiencia del usuario. La invención provee un método para manejar una primera y segunda versiones de un archivo electrónico. En el método, son reconocidas la primera y segunda versiones de un archivo electrónico, y se inicia un evento de sincronización para sincronizar la primera versión sobre el primer computador con la segunda versión, usando una primera aplicación de programa que opera en el primer

computador. Una interfaz del programa de aplicación accesa la primera y segunda versiones; la información asociada con cada una de estas primera y segunda versiones se compara, y se genera un resultado. El resultado de la comparación se reporta a la primera aplicación de programa. A una segunda aplicación de programa se la hace generar una interfaz de usuario de acuerdo con las características de la segunda aplicación de programa, en donde el archivo electrónico está asociado con la segunda aplicación de programa, y la interfaz de usuario es mostrada sobre un dispositivo de pantalla.

La invención también le permite a la segunda aplicación de programa, generar una opción de sincronización, y hacer que esta opción de sincronización sea presentada por la interfaz de usuario y mostrada sobre el dispositivo de pantalla. La segunda aplicación de programa también recibe una opción de selección y sincroniza la primera y segunda versiones del archivo electrónico de acuerdo con esta opción de selección. La invención permita además que la primera y la segunda aplicaciones de programa puedan ser la misma aplicación

de programa, y por lo tanto, todos los pasos pueden ser realizados en una aplicación única.

La aplicación puede suministrar cualquier cantidad de información al usuario respecto de la primera y segunda versiones, y el usuario puede ingresar cualquier número o tipo de selecciones de sincronización. En una modalidad, el usuario puede crear una tercera versión que incluya cualquier porción, o ninguna, ya sea de la primera o de la segunda versión, o de ambas. La aplicación puede iniciar la sincronización en cualquier momento, tal como a solicitud del usuario.

Medio de Computación Ejemplar

La Figura 1 ilustra un ejemplo de un sistema de computación adecuado de ambiente 100, en el cual la invención puede ser implementada. El sistema de computación en ambiente 100 es solamente un ejemplo de un medio de computación adecuado, y no pretende sugerir ninguna limitación al alcance en el uso o funcionalidad de la invención. Ningún medio de computación, 100, debería ser interpretado como teniendo alguna dependencia o requisito relacionado con cualquier otro, o combinación

de componentes ilustrados en el medio de operación ejemplar, 100.

La invención es operacional con muchos otros propósitos generales o especiales de ambientes de sistemas de computo o configuraciones. Ejemplos de sistemas de computación bien conocidos, medio ambientes, y/o configuraciones que pueden ser adecuados para el uso con la invención incluyen, pero no se limitan a, computadores personales, computadores servidores, dispositivos de mano o portátiles, sistemas multiprocesadores, sistemas basados en microprocesadores, set top boxes, artículos electrónicos programables, red de computadores, mini computadores, computadores mainframe, medios de computación distribuidos que incluyen cualquiera de lo sistemas o dispositivos anteriores, y similares.

La invención puede ser descrita en el contexto general de instrucciones de computador ejecutables, tales como módulos de programa, que son ejecutados por un computador. Generalmente, los módulos del programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que llevan a cabo tareas particulares o implementan tipos abstractos de datos

particulares. La invención también puede ser llevada a la práctica en ambientes de computación distribuidos donde las tareas son realizadas por dispositivos remotos de procesamiento que están unidos a través de redes de comunicación u otros medios de transmisión de datos. En un ambiente de computación distribuida, los módulos de programa y otros datos pueden ser localizados tanto en los medios de almacenamiento de computadores locales como remotos, incluidos los dispositivos de almacenamiento de memoria.

Con referencia a la Figura 1, un sistema ejemplar para implementar la invención incluye un dispositivo de computación de propósito general en la forma de un computador 110. Los componentes del computador 110 pueden incluir, pero no están limitados a, una unidad de procesamiento 120, un sistema de memoria 130, y un sistema de bus 121 que acopla los componentes de los diferentes sistemas, incluyendo el sistema de memoria de la unidad de procesamiento 120. El sistema de bus 121 puede ser cualquiera de los diferentes tipos de estructuras de bus incluyendo un bus de memoria o controlador de memoria, un bus periférico, y un bus local que usa una cualquiera de las variedades de arquitectura

de buses. A manera de ejemplo, y no como una limitación, tales arquitecturas incluyen bus de Industry Standard Architecture (ISA), bus de Micro Channel Architecture (MCA), bus de Enhanced ISA (EISA), bus local de Video Electronics Standards Association (VESA), y bus de Peripheral Component Interconnect (PCI) (también conocida como Mezzanine bus).

El computador 110, incluye típicamente una variedad de medios leíbles para él. Los medios leíbles para el computador pueden ser cualquier medio disponible que pueda ser accedido por el computador 110, e incluye tanto medios volátiles como no volátiles, removibles y no removibles. A manera de ejemplo y no como una limitación, los medios leíbles por un computador pueden incluir medios de almacenamiento en computador y medios de comunicación. Los medios de almacenamiento en computador incluyen tanto medios volátiles como no volátiles, removibles como no removibles implementados en cualquier método o tecnología para almacenamiento de información, tales como las instrucciones leíbles por el computador, estructuras de datos, módulos de programa u otros datos. Los medios de almacenamiento en computador incluyen, pero no están limitados a, RAM, ROM, EEPROM, memoria

instantánea u otra tecnología de memoria, CD-ROM, discos digitales versátiles (DVD), u otros almacenamientos en disco óptico, casetes magnéticos, cinta magnética, almacenamiento en disco magnético, u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda ser usado para almacenar la información deseada y al que pueda tener acceso el computador 110. Los medios de comunicación típicamente incluyen instrucciones leíbles por un computador, estructuras de datos, módulos de programa u otros datos en una señal modulada tal como una onda transportadora u otro mecanismo de transporte, e incluye cualquier medio de envío de información. El término "señal de datos modulada" significa una señal que tiene una o más del juego de sus características o sea modificada de tal manera que codifique información en la señal. A manera de ejemplo, y no como una limitación, los medios de comunicación incluyen medios inalámbricos tales como acústicos, RF, infrarrojos y otros. Las combinaciones de cualquiera de los anteriores debería también ser incluida dentro del alcance de los medios leíbles por un computador.

El sistema de memoria 130 incluye medios de almacenamiento en computador en la forma de memorias

volátiles y/o no volátiles tales como ROM 131 y RAM 132. Un sistema básico de entrada / salida 133 (BIOS por sus siglas en inglés), contiene las rutinas básicas que ayudan a transferir información entre los elementos dentro del computador 110, tales como durante el encendido, su almacenamiento típico en ROM 131. RAM 132 contiene típicamente datos y/o módulos de programas que son inmediatamente accesibles y/o están siendo operados en el momento por una unidad de procesamiento 120. A manera de ejemplo, y no como una limitación, la Figura 1 ilustra un sistema operativo 134, programas de aplicación 135, otros módulos de programa 136, y datos de programa 137.

El computador 110 puede también incluir otros medios de almacenamiento removibles/no removibles, volátiles/no volátiles. A manera de ejemplo solamente, la Figura 1 ilustra un mecanismo de disco duro 140 que lee desde o escribe a un medio magnético no removible, no volátil, un mecanismo de disco magnético 151 que lee desde o escribe a un disco magnético 152 removible, no volátil, y un mecanismo de disco óptico 155 que lee desde o escribe a un disco óptico 156 removible, no volátil, tal como un CD-ROM u otros medios ópticos. Otros medios de

almacenamiento en computador removibles/no removibles, volátiles/no volátiles, que pueden ser utilizados en el medio ambiente de operación ejemplar, incluyen, pero no están limitados a, casetes de cinta magnética, tarjetas de memoria instantánea, discos digitales versátiles, cintas de video digital, RAM y ROM, ambos de estado sólido, y similares. El mecanismo de disco duro 141, está conectado típicamente al sistema de bus 121 a través de una interfaz de memoria no removible, tal como la interfaz 140, y el mecanismo de disco magnético 151 y el de disco óptico 155, están típicamente conectados al sistema de bus 121 por una interfaz de memoria removible, tal como la interfaz 150.

Los mecanismos y sus medios asociados de almacenamiento en computador, discutidos antes e ilustrados en la Figura 1, proveen almacenamiento para las instrucciones leíbles por el computador, las estructuras de datos, los módulos de programa y otros datos para el computador 110. En la Figura 1, por ejemplo, el mecanismo de disco duro 141 es ilustrado como almacenando un sistema operativo 144, programas de aplicación 145, otros módulos de programa 146, y datos de programa 147. Note que esos componentes pueden ser, ya sea el mismo o diferentes del sistema

operativo 134, los programas de aplicación 135, otros módulos de programa 136, y datos de programa 137. El sistema operativo 144, , los programas de aplicación 145, otros módulos de programa 146, y datos de programa 147 se presentan con diferentes números aquí para ilustrar que, como mínimo, ellos son copias diferentes. Un usuario puede ingresar comandos e información dentro del computador 110 a través de dispositivos de entrada tales como un teclado 162 y un dispositivo de punteo 161, comúnmente conocido como ratón, bola de pista o almohadilla de contacto. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, palanca de control, almohadilla de juego, disco satelital, explorador, o similares. Estos y otros dispositivos de ingreso están a menudo conectados a la unidad de procesamiento 120 a través de un interfaz de ingreso de usuario 160 que está acoplado al sistema de bus, pero pueden estar conectados por otra interfaz y estructuras de bus, tales como un puerto paralelo, puerto de juegos o un bus serial universal (USB por sus siglas en inglés). Un monitor 191 u otro tipo de dispositivos de pantalla también está conectado al sistema de bus 121 a través de un interfaz, tal como una interfaz de video 190. Además del monitor, los computadores también pueden incluir

otros dispositivos de salida periféricos tales como parlantes 197 e impresora 196, los cuales pueden estar conectados a través de una interfaz periférica de salida 190.

El computador 110 puede operar en un medio de red usando conexiones lógicas para uno o más computadores remotos, tales como un computador remoto 180. El computador remoto 180 puede ser un computador personal, un servidor, un enrutador, una red PC, un dispositivo para escudriñar, u otros nodos de red comunes, y típicamente incluyen muchos o todos los elementos descritos antes relacionados con el computador 110, aunque solamente un dispositivo de almacenamiento de memoria 181 ha sido ilustrado en la Figura 1. Las conexiones lógicas descritas incluyen una red de área local (LAN por sus siglas en inglés) 171 y una red de área extendida (WAN por sus siglas en inglés) 173, pero también pueden incluirse otras redes. Tales medios en red son lugares comunes en las oficinas, grandes redes de computo de empresas, redes internas y la Internet.

Cuando se usa en un medio de red LAN, el computador 110 está conectado a la LAN 171 a través de una interfaz de

red o adaptador 170. Cuando se usa en un medio de red WAN, el computador 110 típicamente incluye un modem 172 u otros medios para establecer comunicaciones sobre la WAN 173, tales como la Internet. El MODEM 172, el cual puede ser interno o externo, puede estar conectado al sistema de bus 121 a través de una interfaz de entrada de usuario 160, u otros mecanismos apropiados. En un ambiente de red, los módulos de programa descritos relativos a l computador 110, o porciones del mismo, pueden estar almacenados en un dispositivo remoto de almacenamiento de memoria. A manera de ejemplo, y no limitado por ello, la Figura 1 ilustra programas de aplicación remota 185 residentes sobre el dispositivo de memoria 181. Se apreciará que las conexiones de red mostradas son ejemplares y que otros medios para establecer un enlace de comunicaciones, pueden ser usados entre los computadores.

Arquitecturas o Armazones de Computación Distribuidos en Forma Ejemplar

Varios armazones distribuidos de computo han sido y están siendo desarrollados en vista de la convergencia de computación personal y la Internet. A los usuarios

individuales y de negocios por igual, se los provee con una interfaz habilitada para la red e interoperable sin remiendos para aplicaciones y dispositivos de computo, haciendo actividades de computo incrementando los buscadores de red u orientados hacia la red.

Por ejemplo, la plataforma MICROSOFT[®]s.NET incluye servidores, servicios de construcción en bloque, tales como el almacenamiento de datos basado en la red, y el programa de dispositivo descargable. Hablando en general, la plataforma .NET brinda (1) la posibilidad de hacer el rango completo del trabajo de dispositivos de computación juntos y de tener actualizada automáticamente la información de usuario y sincronizada en todos ellos, (2) capacidad interactiva incrementada para los sitios web, posibilitada para mayor uso de XML en vez de HTML, (3) servicios en línea que dan importancia al acceso personalizado y el envío de productos y servicios al usuario desde un punto central de inicio para el manejo de varias aplicaciones, tales como correo electrónico, por ejemplo, o programas, tales como Office.NET, (4) almacén de datos centralizado, que incrementará la eficiencia y facilidad de acceso a la información, así como a la sincronización de información entre

dispositivos y usuarios, (5) la posibilidad de integrar varios medios de comunicación, tales como el correo electrónico, facsímiles y teléfonos, (6) para desarrolladores, la capacidad de crear módulos reusables, incrementando por lo tanto la productividad y reduciendo el número de errores programación, y (7) la integración así mismo de las características de muchas otras plataformas cruzadas.

Mientras que las modalidades ejemplares son descritas aquí en conexión con un programa residente sobre un dispositivo de computación, una o más secciones de la invención pueden ser implementadas también a través de un sistema operacional, API, o una objeción de "hombre medio" entre un co-procesador y el objeto solicitado, tales servicios pueden ser realizados por, soportados en, o accedidos a través de todos los lenguajes y servicios de .NET, y también en otros armazones distribuidos de computo.

Modalidades Ejemplares

Cuando un archivo electrónico ha sido creado en más de una versión, de acuerdo con la invención, una interfaz de

programa de aplicación puede ser configurada para permitir a una aplicación de programa asociada con el archivo, controlar un proceso de sincronización a las diferentes versiones. En una modalidad, la aplicación puede iniciar automáticamente la sincronización, mientras que en otra modalidad, la aplicación puede iniciar la sincronización con base en la entrada del usuario. En otra modalidad, la aplicación presenta opciones de sincronización proveídas por la funcionalidad de aplicaciones del usuario, y que le permiten a éste seleccionar las opciones y llevar a cabo tareas usando la funcionalidad de la aplicación. En otra modalidad aun, una primera aplicación inicia la sincronización, mientras que una segunda aplicación presenta opciones de sincronización al usuario.

Refiriéndonos ahora a la Figura 2, un primer computador 210 incluye un primer procesador 220 en comunicación operativa con un dispositivo de almacenamiento 218. El primer computador 210 puede ser un computador de mesa o portátil para propósitos generales, una estación de trabajo, un dispositivo de computo especializado y/o similares. Además, el primer procesador 220 puede ser un procesador de computador para propósitos generales, un

procesador especializado, una red de procesadores o similares. Operativamente asociado con el primer computador 210 está el dispositivo de pantalla 230. El dispositivo de pantalla 230 puede ser cualquier medio de presentar opciones y similares para un usuario, tales como por ejemplo un monitor, parlantes o similares. Tal asociación operativa del dispositivo de pantalla 230 con el primer computador 210 puede ser hecha por medio de conexión inalámbrica o por cables. Mientras que el dispositivo de pantalla 230 se lo muestra dentro del primer computador 210 en la Figura 2, el dispositivo de pantalla 230 puede estar localizado en cualquier parte que esté en comunicación operativa con el primer procesador 220, para incluir una ubicación remota o similar.



El dispositivo de almacenamiento 218 puede ser un mecanismo duro, CD-ROM, disco magnético, DVD, RAM y/o similares. Al igual que el dispositivo de pantalla 230, el dispositivo de almacenamiento 218 puede estar localizado en cualquier parte que esté en comunicación operativa con el primer procesador 220. la comunicación operativa en tal situación puede ser por cable, inalámbrica o cualquier combinación de las mismas.

Contenida dentro del dispositivo de almacenamiento 218 está una primera versión 212 de un archivo electrónico. Tal primera versión 212 puede ser cualquier tipo de archivo electrónico, tal como por ejemplo un procesador de palabra de documentos, una hoja de cálculo, una base de datos y/o similares.

La primera aplicación 208 y la segunda aplicación 214 pueden estar almacenadas en el dispositivo de almacenamiento 218. La primera aplicación 208 y la segunda aplicación 214 pueden ser cualquier tipo de instrucciones ejecutables por un computador, tales como por ejemplo un procesador de palabra, hoja de calculo, base de datos, tipo de letra programada, programa especializado o similares. En las modalidades en donde tanto la primera aplicación 208 como la segunda 214 están presentes, como se discutió más abajo, solamente la segunda aplicación 214 necesita ser compatible con el archivo electrónico. Alternativamente, y como también será discutido más abajo, la primera aplicación 208 y la segunda aplicación 214 pueden ser la misma aplicación. Una o ambas de la primera aplicación 208 y la segunda aplicación 214, pueden también estar localizadas afuera del dispositivo de almacenamiento 218, ya que cualquiera

de las dos aplicaciones solamente necesitan estar en comunicación operativa con el primer computador 210.

La interfaz del programa de aplicación 216 es cualquier tipo de computador que ejecute instrucciones adaptado para chequear y comparar dos o más copias de un archivo electrónico para propósitos de sincronización, y puede ser el único programa que esté específicamente diseñado para la sincronización de dos o más copias de un documento electrónico, una función de programa que opera como un componente de otra aplicación, un componente integrado de la primera o segunda aplicación 208, 214 y/o similares. La interfaz de programa de aplicación 216 puede también realizar tareas adicionales que pueden o no estar relacionadas con la sincronización. La primera versión 212, primera aplicación 208, segunda aplicación 214 e interfaz de programa de aplicación 216 pueden residir en el mismo dispositivo de almacenamiento 218, o dentro de o a través de cualquier diferente número de dispositivos de almacenamiento 218.

Un segundo computador 222 incluye un segundo procesador 224 y un dispositivo de almacenamiento 228 el cual contiene una segunda versión 226 del archivo electrónico

discutido antes en conexión con la primera versión 212. Como se discutió antes, el dispositivo de almacenamiento 228 puede estar localizado en cualquier parte que esté en comunicación operativa con un segundo computador 222, que incluye una ubicación remota o similar. También, el dispositivo de almacenamiento 228 y el 218 pueden, en una de las modalidades, ser el mismo dispositivo de almacenamiento. Conectando operativamente el primero y segundo computadores 210, 222, está la red 218. La red 218 puede ser cualquier red de comunicaciones adaptada para tales comunicaciones operativas, tal como por ejemplo una LAN, WAN, red inalámbrica o similares. Como puede ser apreciado, cualquier número de primeros computadores 210 y segundos computadores 222 pueden ser operativamente conectados a la red 218.

Volviendo ahora a la Figura 3, en la etapa 311 se inicia un proceso de sincronización. En una modalidad, la primera aplicación 208 inicia un proceso de sincronización, y una segunda aplicación 214 lleva a cabo el proceso de sincronización. En otra modalidad, una primera aplicación 208 y una segunda aplicación 214 son una aplicación simple, en cuyo caso la aplicación simple, inicia y lleva a cabo la sincronización. Mientras la

discusión a continuación de la Figura 3 describe un proceso que involucra tanto a la primera aplicación 208 como a la segunda aplicación 214, puede apreciarse que una aplicación simple puede por si sola llevar a cabo todas las etapas realizadas por la primera aplicación 208 y la segunda aplicación 214.

Cualquier número de situaciones pueden causar que la primera aplicación 208 inicie un proceso de sincronización tales como por ejemplo: abrir, cerrar, salvar o similares de la primera versión 212; por el reconocimiento de una primera y una segunda versiones del archivo electrónico; iniciación manual por solicitud de un usuario; iniciación en intervalos de tiempo; por registro de usuario dentro de un sistema y/o similares. Adicionalmente, la primera aplicación 208 puede imponer cualesquiera condiciones adicionales o puede usar condiciones sustitutas tales como, por ejemplo, solamente iniciar una sincronización cuando un cierto número de versiones estén presentes. Mas detalles con respecto al inicio de un evento de sincronización en una modalidad se presentaran abajo, en conexión con la Figura 4. Para iniciar el proceso, la primera aplicación 208 ocasiona que la interfaz del programa de aplicación 216, acceda a

la primera versión 212 y a la segunda versión 226. El acceso a la primera versión 212 y a la segunda versión 226 puede ser llevada a cabo por cualquier medio de computación, tal como la inspección de la información asociada con cada versión, encontrando cada versión por medio del uso de la opción de búsqueda, localización de cada versión usando información de identificación suministrada por la primera aplicación 208 o similar.

La primera aplicación 208 puede proveer la interfaz 216 con cualquier cantidad de información de identificación acerca de la segunda versión 226, tal como por ejemplo un nombre de archivo, una localización, dirección y/o similares. La interfaz 216 puede ser adaptada para localizar activamente la segunda versión 226, o puede tener una funcionalidad limitada que permita ver solamente una localización especificada por la primera aplicación 208, primera versión 212 o similares. También, la interfaz 216 puede ser configurada para realizar cualquier número adicional de tareas que pueden inclusive no estar relacionadas con el proceso de sincronización: En la etapa 313, la interfaz del programa de aplicación 216 localiza la segunda versión 226 y chequea el estatus de la sincronización. La interfaz del programa de

aplicación 216 determina el estatus comparando la información, tal como por ejemplo, la hora y/o la fecha de una última actualización o similar, para la primera versión 212 contra la segunda versión 226. El estatus de la sincronización puede ser determinado también por cualquier criterio adicional a, o sustituido por la mencionada información anterior. Si la información iguala a la información y/o criterios especificados por la primera aplicación 208, la primera y segunda versiones 212, 226 presentan un estatus sincronizado. Si la información es inconsistente, la primera y segunda versiones 212, 226 no presentan un estatus sincronizado y tal inconsistencia indica la posibilidad de que se hayan podido hacer cambios en una de tales versiones 212, 226. Alternativamente si la interfaz del programa de aplicación 216 detecta un error, una condición no prevista, una condición que la primera aplicación 208 ha especificado como causa para terminar la sincronización o similar, entonces la interfaz 216 procede hacia la etapa 330. En la etapa 330, la interfaz 216 informa a la primera aplicación 208 de esa condición, y primera aplicación 208 procede en consecuencia, lo cual puede incluir cualquier tipo de acción o de inacción.

En la etapa 315, la interfaz del programa de aplicación 216 puede tomar uno de dos caminos de acción, dependiendo del resultado de la determinación de sincronización de la etapa 313. Si la interfaz del programa de aplicación 216 ha determinado que la primera versión 212 y la segunda versión 216 están sincronizadas la interfaz 216 procede hacia la etapa 325. En la etapa 325, la interfaz del programa de aplicación reporta a la primera aplicación 208 que la primera versión 208 y la segunda versión 226 están sincronizadas. Tal reporte le indica a la primera aplicación 208 que ninguna actividad adicional de sincronización se requiere en ese momento, ya que la primera versión 212 incluye la más reciente versión del archivo electrónico. El reporte también puede indicar que un usuario puede utilizar la primera aplicación 208 o la segunda aplicación 214 para alterar la primera versión 212.

En la etapa 323, termina el proceso de sincronización. En esta etapa la primera aplicación 208 puede dejar de funcionar, como por ejemplo si tal proceso de sincronización tiene lugar cuando la primera aplicación 208 estaba cerrando la primera versión 212 del documento electrónico. En tal situación, una vez que la

sincronización ha ocurrido, entonces, la primera aplicación 208 no tiene más funciones que realizar y puede ser terminado manualmente por un usuario, o automáticamente. En otra modalidad, la primera aplicación 208 inicia el proceso de sincronización cuando se abre una primera versión 212. En tal situación, la aplicación no terminaría en su totalidad tal evento de sincronización, y en vez de eso permitiría al usuario iniciar la alteración de la primera versión 212 utilizando la función de la primera aplicación 208, si está disponible.

Sin embargo, si en la etapa 315, la interfaz del programa de aplicación 216 ha determinado que la primera versión 212 y la segunda versión 226, no están sincronizadas, la interfaz 216 procede hasta la etapa 317. En la etapa 317, la interfaz del programa de aplicación 216 reporta a la primera aplicación 208 y/o segunda aplicación 214, que la primera versión 212 y la segunda versión 226 no están sincronizadas. Tal reporte causa que la segunda aplicación 214 comience la sincronización de la primera versión 212 y la segunda versión 226 con la entrada del usuario como se discutió antes en conexión con las etapas 319 y 321.

En la etapa 319, la segunda aplicación 214 presenta una interfaz de usuario al usuario, mostrando la interfaz de usuario sobre un dispositivo de pantalla 230. Tal interfaz de usuario puede estar en la forma de un texto y/o ventana grafica, apunte de audio y/o similares, y cualquiera de tales configuraciones de una interfaz de usuario es igualmente consistente con una modalidad de la presente invención. En la presentación de tal interfaz de usuario, la segunda aplicación 214 puede presentar una o más opciones a un usuario utilizando una opción de la segunda aplicación 214 o similar. Por ejemplo si la segunda aplicación 214 es un procesador de palabra, la segunda aplicación 214 puede presentar una opción de sincronización a un usuario dentro del ambiente de usuario del procesamiento de palabra de la segunda aplicación 214, tal como por ejemplo una ventana de edición con una vista de la página de tal primera versión 212 y/o similar. La segunda aplicación 214 también puede resaltar las diferencias entre la primera versión 212 y la segunda versión 226 usando color, sonidos, gráficos y similares para permitirle en mejor forma al usuario tomar las decisiones de sincronización. La segunda aplicación 214 puede también proporcionar información adicional, tal

como por ejemplo, la identificación de cada usuario que altero cualquiera de la s versiones 212,226, y puede permitir solamente a un usuario, alterar una versión 212, 226, si tal usuario tiene permiso, en la forma de un beneficio de seguridad o similar para hacerlo. La segunda aplicación 214 puede permitir al usuario, por ejemplo combinar las versiones 212, 226 en una tercera copia para incorporar los cambios en ambas versiones 212, 226. Una opción adicional también puede estar en la forma de una guía derivada de una opción de la segunda aplicación 214, tal como por ejemplo un curso de acción recomendado y/o similares. Como puede ser apreciado, una opción puede ser cualquier cosa que pueda ser generada por la segunda aplicación 214 usando una opción de la segunda aplicación 214, con respecto a la primera versión 212 y a la segunda versión 226.

En la etapa 321, la segunda aplicación 214, recibe y procesa la entrada del usuario. Tal entrada de usuario puede incluir cualquier cantidad de información relacionada con tal información por ejemplo, la entrada de usuario puede seleccionar una alteración particular ya sea de la primera versión 212, o de la segunda versión 226 del archivo electrónico que el usuario desea

conservar. En otra modalidad, la entrada de usuario puede incluir una selección ya sea de la primera versión 212 o de la segunda versión 226 que el usuario desea conservar en su totalidad. En otra modalidad aún, la entrada de usuario puede incluir editar la primera y/o segunda versiones 212, 226 utilizando la herramienta de la segunda aplicación 214 que, cuando se completa resulta en la sincronización de la primera versión 212 y la segunda versión 226.

Como puede ser apreciado, la entrada de usuario puede incluir cualquier entrada que este disponible por la segunda aplicación 214 para selección o uso por un usuario, y cualquiera de esas entradas es consistente con una modalidad de la presente invención.

Como también puede ser apreciado, las etapas 319 y 321 pueden repetir, como tal interfaz de usuario puede ser mostrada nuevamente y/o refrescada dependiendo de la entrada de usuario. Por ejemplo, la interfaz de usuario puede ser reconfigurada por la segunda aplicación 214 basada en la entrada de usuario. Tal reconfiguración puede ocurrir, si, por ejemplo, el usuario selecciona una de una variedad de diferencias entre la primera versión

212 y la segunda versión 226, y la segunda aplicación 214 vuelve a mostrar la interfaz de usuario al usuario, para reflejar tal selección. Las etapas 319 y 321 pueden repetirse cualquier número de veces o no en absoluto, y cualquiera de tales situaciones es igualmente consistente con una modalidad de la presente invención. La segunda aplicación 214 también puede reportar cualquier resultado de la sincronización a la primera aplicación 208 o a la interfaz 216. Los resultados reportados por la segunda aplicación 214 puede ser cualquier información asociada con la sincronización tal como la(s) opción(es) de selección, resultado(s) de sincronización o similares.

La segunda aplicación 214 puede proceder hacia la etapa 223, el final de la sincronización, cuando ocurre un evento activador. Tal evento de activación puede ocurrir en cualquier momento, como por ejemplo cuando el usuario ha completado la reconciliación de todas las diferencias entre la primera versión 212 y la segunda versión 226, cuando el usuario se dirige a la segunda aplicación 214 para abortar o terminar el proceso de sincronización, sin haber completado el mismo, a un tiempo predeterminado o similar. En la etapa 323, ¹termina la sincronización. La primera aplicación 208 o la segunda aplicación 214 pueden

entonces, como se menciona antes en conexión con la etapa 325, tomar cualquier acción en la etapa 323 que sea consistente con el contexto en el cual la sincronización tuvo lugar.

Volviendo ahora a la Figura 4, una vista detallada de la etapa 311 es mostrada de acuerdo con una modalidad de la presente invención. Como se discutió antes en relación con la Figura 3, en la etapa 311 la primera aplicación 208 inicia un proceso de sincronización. La Figura 4 ilustra muy particularmente una modalidad de tal etapa de iniciación, donde la primera aplicación 208 inicia tal proceso de sincronización por la apertura de la primera versión 212 de un archivo electrónico. Como se discutió antes, la primera aplicación 208 puede iniciar tal proceso de sincronización en cualquier momento, como por ejemplo cuando se abre, salva, o cierra tal primera versión 212, un juego de intervalos variables, por ingreso del usuario y/o similar.

En la etapa 411, la primera aplicación 208 abre la primera versión 212 de un archivo electrónico. Tal apertura deberá comúnmente ocurrir por entrada del usuario para abrir tal primera versión 212, pero

cualquier medio para iniciar tal apertura de la primera versión 212 es consistente con una modalidad de la presente invención. En la etapa 413, la primera aplicación 208 determina si la primera versión 212 tiene asociada una segunda versión 226. La primera aplicación 208 puede proceder hacia la etapa 413 hasta una apertura total d la versión 212 en la etapa 411, o puede proceder hacia la etapa 413 a cualquier punto durante tal proceso de apertura incluyendo, por ejemplo, antes de abrir la primera versión 212 en la etapa 411. Cualquier método puede ser también empleado para permitir la primera aplicación 208 para tomar tal determinación. En contraste, la primera aplicación 208 puede explorar alguno o todos los dispositivos de computo operativamente conectados a la red 218, como por ejemplo el segundo computador 222, para tal determinación. Como puede ser apreciado, la primera aplicación 208 puede usar cualquier método para determinar la existencia de la segunda versión 226 mientras que el resto es consistente con una modalidad de la presente invención.

Si, en la etapa 413, la primera aplicación 208 no encuentra la segunda versión 226, la primera aplicación 208 determina que la primera versión 212 es la única

copia de tal archivo electrónico y procede hasta la etapa 417.

En la etapa 417, la primera aplicación 208 procesa normalmente la primera versión 212. Tal procesamiento normal puede ser cualquier tipo de procesamiento que la primera aplicación 208 típicamente realiza sobre el archivo electrónico del mismo formato como la primera versión 212, posiblemente de acuerdo con cualquier entrada de usuario. Por ejemplo, en la Figura 4, la primera aplicación 208 estaba en el proceso de abrir la primera versión 212 cuando la primera aplicación 208 inició tal proceso de sincronización de acuerdo con la etapa 311. Al no tener necesidad de una sincronización- debido a la ausencia de la segunda versión 226- la primera aplicación 208 completa su apertura de la primera versión 212 y procede en consecuencia.

Si, en la etapa 413, la primera aplicación 208 encuentra la segunda versión 226 primera aplicación 208, procede hacia la etapa 415. En la etapa 415, la primera aplicación 208 contacta la interfaz del programa de aplicación 216 para comenzar la sincronización de la primera versión 212 y la segunda versión 226. Por

contacto de la interfaz del programa de aplicación 216, tal proceso de sincronización procede hacia la etapa 313 como se discutió antes en conexión con la Figura 3.

Así, han sido presentados los métodos y aparatos para manejar la replicación de archivos en aplicaciones. Mientras que la presente invención ha sido descrita en conexión con las modalidades preferidas de las diferentes figuras, debe ser entendido que otras modalidades similares pueden ser usadas o las modificaciones y adiciones pueden ser hechas a partir de la modalidad descrita para realizar la misma función de la presente invención sin desviarse de la misma. Por ejemplo, una persona entrenada en la técnica reconocerá que la presente invención tal como se describió en la presente solicitud, puede aplicarse a cualquier dispositivo de computo o ambiente ya sea con cables o inalámbrico, y puede ser aplicado a cualquier número de tales dispositivos de computo conectados a través de una red de comunicaciones, e interactuando a través de la red. Además debería enfatizarse que una variedad de plataformas de computador, incluyendo los sistemas operativos de dispositivos de mano y otras aplicaciones con sistemas operativos específicos son contempladas,

especialmente si el número de dispositivos inalámbricos conectados a la red continua proliferando. Aún además, la presente invención puede ser implementada en, o a través de una variedad de chips de procesamiento o dispositivos, y el almacenamiento puede verse afectado de forma similar a través de una variedad de dispositivos. Por lo tanto, la presente invención no debe estar limitada a cualquier simple modalidad sino que en vez de eso debería ser construida con la libertad y alcance acordes con las reivindicaciones anexas.

Lo que se Reivindica es:

1. Un método para manejar una primera y una segunda versión de un archivo electrónico, donde el método comprende:

iniciar, usando una primera aplicación de programa operando sobre un primer computador, un evento de sincronización para sincronizar la primera versión sobre el primer computador con la segunda versión;

originar una interfaz de programa de aplicación para acceder a dichas primera y segunda versiones;

comparar la información asociada con cada una de dichas primera y segunda versiones y generar un resultado;

reportar el resultado de la comparación a la primera aplicación de programa;

Originar una segunda aplicación de programa para generar una interfaz de usuario de acuerdo con una característica de la segunda aplicación de programa, en donde el archivo

electrónico esta asociado con la segunda aplicación de programa; y

Desplegar la interfaz de usuario sobre un dispositivo de pantalla.

2. El método de la reivindicación 1, en donde la primera y la segunda aplicaciones de programa son la misma aplicación.

3. El método de la reivindicación 1, en donde el resultado de dicha etapa de comparación es un estatus de sincronización de la primera y segunda versiones.

4. El método de la reivindicación 1, en donde la segunda versión esta almacenada en un segundo computador, y dichos primero y segundo computadores están operativamente conectados por una red de comunicaciones.

5. El método de la reivindicación 1, comprendiendo además la etapa cuando el resultado de la comparación indica que la primera y segunda versiones del archivo electrónico no están sincronizadas, dicha etapa de generación comprende

además, presentar, usar la segunda aplicación del programa, opciones de sincronización.

6. El método de la reivindicación 6, en donde dicha etapa de presentar las opciones de sincronización comprende además:

generar una opción de sincronización por medio de la segunda aplicación de programa;

originar la opción de sincronización para ser presentada por la interfaz de usuario y desplegarla sobre el dispositivo de pantalla,

recibir una selección de opción; y

sincronizar la primera y segunda versiones del archivo electrónico de acuerdo con la selección de opción.

7. El método de la reivindicación 6, en donde la etapa de sincronización de la primera y segunda versiones comprende sobrescribir las segundas versiones del archivo electrónico con la segunda versión del archivo electrónico.

8. El método de la reivindicación 6, en donde la etapa de sincronización de la primera y segunda versiones comprende sobrescribir la primera y segunda versiones con una tercera versión del archivo electrónico, dicha tercera versión comprendiendo una porción de la primera versión del archivo electrónico.

9. El método de la reivindicación 6, en donde la etapa de sincronización de la primera y segunda versiones comprende sobrescribir la primera y segunda versiones con una tercera versión del archivo electrónico, dicha tercera versión comprendiendo una porción de la segunda versión del archivo electrónico.

10. El método de la reivindicación 6, en donde la etapa de sincronización de la primera y segunda versiones comprende fusionar la primera y segunda versiones en una tercera versión de acuerdo con la entrada recibida por la interfaz de usuario y sobrescribir la primera y segunda versiones con la tercera versión.

11. El método de la reivindicación 1, comprendiendo además:

abrir la primera y segunda versiones usando la segunda aplicación de programa;

presentar una opción de sincronización utilizando la interfaz de usuario, la opción de sincronización generada por la segunda aplicación de programa;

recibir la entrada de usuario de la interfaz de usuario; procesar la entrada de usuario utilizando la segunda aplicación de programa; y

sincronizar la primera y segunda versiones de acuerdo con la entrada de usuario.

12. El método de la reivindicación 11, en donde dicha etapa de presentar una opción de sincronización comprende además desplegar la primera y segunda copias en la interfaz de usuario.

13. El método de la reivindicación 11, en donde la etapa de sincronización de la primera y segunda versiones comprende sobrescribir la primera y segunda versiones de acuerdo con la entrada de usuario.

14. El método de la reivindicación 11, en donde la etapa de iniciar el evento de sincronización ocurre sobre la primera aplicación de programa detectando la información asociada con la primera versión del archivo electrónico indicativa de la segunda versión del archivo electrónico.

15. El método de la reivindicación 1, en donde la etapa de iniciación del evento de sincronización ocurre sobre la primera aplicación de programa abriendo la primera versión del archivo electrónico.

16. El método de la reivindicación 1, en donde la etapa de iniciación del evento de sincronización ocurre sobre la primera aplicación de programa salvando la primera versión del archivo electrónico en un dispositivo de almacenamiento del primer computador.

17. El método de la reivindicación 1, en donde la etapa de iniciación del evento de sincronización ocurre sobre la primera aplicación de programa reconociendo la primera y segunda versiones.

51
?

18. Un método para manejar una primera y segunda versiones de un archivo electrónico, el método comprendiendo:

recibir una solicitud de una primera solicitud de una primera aplicación de programa para acceder y comparar dichas primera y segunda versiones;

acceder a dichas primera y segunda versiones;

comparar la información asociada con cada una de dichas primera y segunda versiones;

determinar si dichas primera y segunda versiones están sincronizadas y generar un resultado; y

reportar el resultado de la comparación a la primera aplicación de programa.

19. El método de la reivindicación 18, en donde dicha etapa de recibir una solicitud comprende además recibir información de identificación correspondiente a dichas primera y segunda versiones.

20. El método de la reivindicación 18, en donde la segunda versión es almacenada en un segundo computador, y dichos primero y segundo computadores están operativamente conectados por una red de comunicaciones.

21. El método de la reivindicación 18, comprendiendo

además:

recibir una segunda solicitud de la primera aplicación del programa, dicha segunda solicitud para acceder a dichas primera y segunda versiones y realizar una tarea relacionada con dichas primera y segunda versiones; y acceder a dichas primera y segunda versiones y realizar dicha tarea.

22. El método de la reivindicación 18, comprendiendo

además:

recibir una solicitud de la segunda aplicación de programa, la segunda aplicación de programa asociada con el archivo electrónico, para sobrescribir la primera y segunda versiones de acuerdo con la entrada de usuario recibida por la segunda aplicación de programa; y

sobrescribir la primera y segunda versiones.

23. El método de la reivindicación 18, en donde la primera y segunda aplicaciones de programa son la misma aplicación.

24. Un medio leíble por computador teniendo el computador instrucciones ejecutables para realizar un método para manejar una primera y segunda versiones de un archivo electrónico, el método comprendiendo:

iniciar, usando una primera aplicación de programa operando sobre un primer computador, un evento de sincronización para sincronizar la primera versión sobre el primer computador con la segunda versión;

originar una interfaz de programa de aplicación para acceder a dichas primera y segunda versiones;

comparar la información asociada con cada una de dichas primera y segunda versiones y generar un resultado;

reportar el resultado de la comparación a la primera aplicación de programa;

Originar una segunda aplicación de programa para generar una interfaz de usuario de acuerdo con una característica de la segunda aplicación de programa, en donde el archivo electrónico esta asociado con la segunda aplicación de programa; y

Desplegar la interfaz de usuario sobre un dispositivo de pantalla.

25. El medio legible por computador de la reivindicación 24 en donde la primera y segunda aplicaciones de programa son la misma aplicación.

26. El medio legible por computador de la reivindicación 24 en donde el resultado de dicha etapa de comparación es un estatus de sincronización de la primera y segunda versiones.

27. El medio legible por computador de la reivindicación 24, en donde la segunda versión es almacenada sobre un segundo computador, y dichos primero y segundo computadores están operativamente conectados por una red de comunicaciones.

28. El medio leíble por computador de la reivindicación 24, en donde el método comprende además la etapa de que cuando el resultado de la comparación indica que la primera y segunda versiones del archivo electrónico no están sincronizadas, dicha etapa de generación comprende además, presentar, usar la segunda aplicación de programa, opciones de sincronización. = 5

29. El medio leíble por computador de la reivindicación 27, en donde, dicha etapa de presentar opciones de sincronización comprende además: ?

generar una opción de sincronización por medio de la segunda aplicación de programa; = 6

originar que la opción de sincronización sea presentada por la interfaz de usuario y desplegada sobre el dispositivo de pantalla;

recibir una opción de selección; y

sincronizar la primera y segunda versiones del archivo electrónico de acuerdo con la opción de selección.

30. El medio leíble por computador de la reivindicación 29, en donde la etapa de sincronizar la primera y segunda versiones comprende sobre escribir las segundas versiones del archivo electrónico con la segunda versión del archivo electrónico. = 7

31. El medio leíble por computador de la reivindicación 29, en donde la etapa de sincronizar la primera y segunda versiones comprende sobrescribir la primera y segunda versiones con una tercera versión del archivo electrónico, dicha tercera versión comprendiendo una porción de la primera versión del archivo electrónico.

32. El medio leíble por computador de la reivindicación 29, en donde la etapa de sincronizar la primera y segunda versiones comprende sobrescribir la primera y segunda versiones con una tercera versión del archivo electrónico, dicha tercera versión comprendiendo una porción de la segunda versión del archivo electrónico. = 9

33. El medio leíble por computador de la reivindicación 29, en donde la etapa de sincronizar la primera y segunda versiones comprende fusionar la primera y segunda

versiones dentro de una tercera versión de acuerdo con la entrada recibida por la interfaz de usuario y sobrescribir la primera y segunda versiones con la tercera versión. =10

34. El medio leíble por computador de la reivindicación 24, en donde el método comprende además:

Abrir la primera y segunda versiones usando la segunda aplicación de programa;

Presentar una opción de sincronización usando la interfaz de usuario, la opción de sincronización generada por la segunda aplicación de programa; =11

recibir la entrada de usuario de la interfaz de usuario;
procesar la entrada de usuario usando la segunda aplicación de programa; y

sincronizar la primera y segunda versiones de acuerdo con la entrada del usuario.

35. El medio leíble por computador de la reivindicación 34, en donde dicha etapa de presentar una opción de

?

sincronización comprende además desplegar la primera y segunda copias en la interfaz d usuario. = 12

36. El medio leíble por computador de la reivindicación 34, en donde la etapa de sincronizar la primera y segunda versiones comprende sobrescribir la primera y segunda versiones de acuerdo con la interfaz de usuario. 13

37. El medio leíble por computador de la reivindicación 24, en donde la etapa de iniciar el evento de sincronización ocurre sobre la primera aplicación de programa detectando la información asociada con la primera versión del archivo electrónico indicativa de la segunda versión del archivo electrónico. 14

38. El medio leíble por computador de la reivindicación 24, en donde la etapa de iniciar el evento de sincronización ocurre sobre ~~la primera~~ aplicación de programa abriendo la primera versión del archivo electrónico. = 15

39. El medio leíble por computador de la reivindicación 24, en donde la etapa de iniciar el evento de sincronización ocurre sobre la primera aplicación de 17

programa salvando la primera versión del archivo electrónico a un dispositivo de almacenamiento del primer computador. = 16

40. El medio leíble por computador de la reivindicación 24, en donde la etapa de iniciar el evento de sincronización ocurre sobre la primera aplicación de programa reconociendo la primera y la segunda versiones. = 17

41. El medio leíble por computador teniendo instrucciones ejecutables por computador para manejar una primera y segunda versiones de un archivo electrónico el método comprendiendo: *obscuro*

recibir una solicitud de una primera aplicación de programa para acceder y comparar dichas primera y segunda versiones; = 18

acceder a dichas primera y segunda versiones;

comparar la información asociada con cada una de dichas primera y segunda versiones; determinar si dichas primera y segunda versiones están sincronizadas y generar un resultado; y

reportar el resultado de la comparación a la primera aplicación de programa.

42. El medio ~~leíble~~ por computador de la reivindicación 41, en donde dicha etapa de recibir una solicitud comprende además recibir información de identificación correspondiente a dichas primera y segunda versiones. = 19

43. El medio ~~leíble~~ por computador de la reivindicación 41, en donde la segunda versión es almacenada sobre un segundo computador, y dichos primero y segundo computadores están operativamente conectados por una red de comunicaciones. = 20

44. El medio ~~leíble~~ por computador de la reivindicación 41, en donde el método comprende además:

recibir una segunda solicitud de la primera aplicación de programa, dicha segunda solicitud para acceder a dichas primera y segunda versiones y realizar una tarea relacionada con dichas primera y segunda versiones; y acceder a dichas primera y segunda versiones y realizar dicha tarea. = 21

45. El medio leíble por computador de la reivindicación 41, en donde el método comprende además:

recibir una solicitud de la segunda aplicación de programa, la segunda aplicación de programa asociada con el archivo electrónico, sobrescribir la primera y segunda versiones de acuerdo con la entrada de usuario recibida por la segunda aplicación de programa; y

=22

sobrescribir la primera y segunda versiones.

46. El medio leíble por computador de la reivindicación 41, en donde la primera y segunda aplicaciones de programa son la misma aplicación.

=23

RESUMEN

Se describe un método para manejar una primera y segunda versiones de un archivo electrónico. En el método, la primera y segunda versiones de un archivo electrónico son reconocidas y se inicia un evento de sincronización. Una interfaz de programa de aplicación accesa a la primera y segunda versiones, compara la información asociada con cada una y genera un resultado y reporta el resultado a la primera aplicación de programa. Una segunda aplicación de programa genera una interfaz de usuario de acuerdo con una característica de la segunda aplicación de programa, en donde el archivo electrónico esta asociado con la segunda aplicación de programa y la interfaz de usuario es desplegada sobre un dispositivo de pantalla.

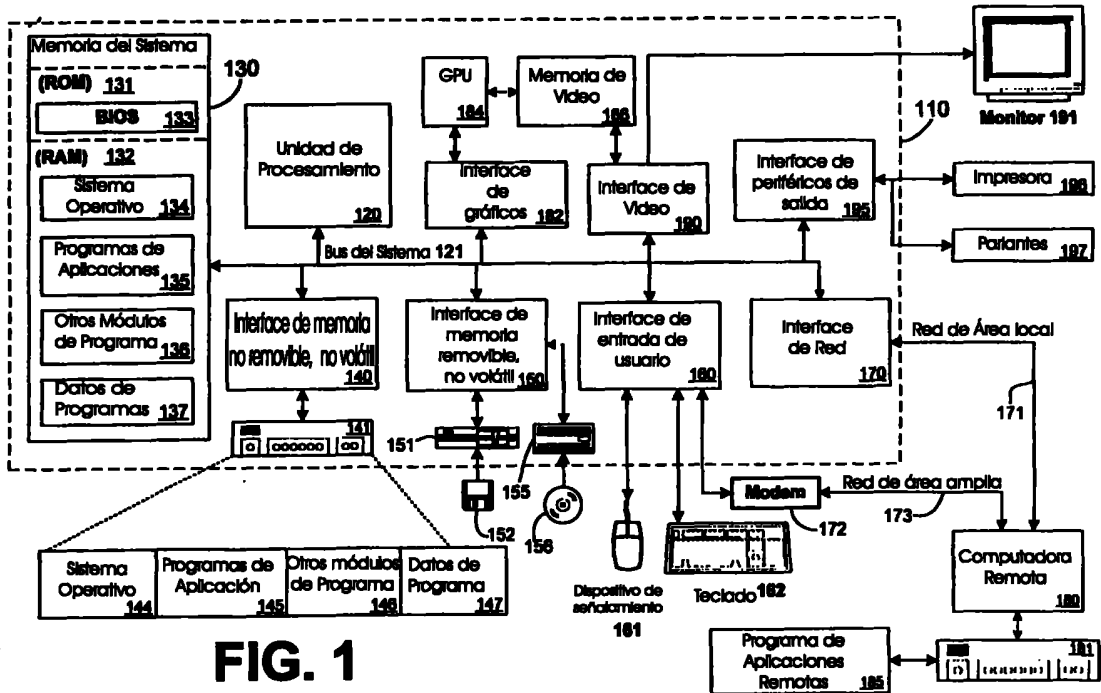


FIG. 1

2/4

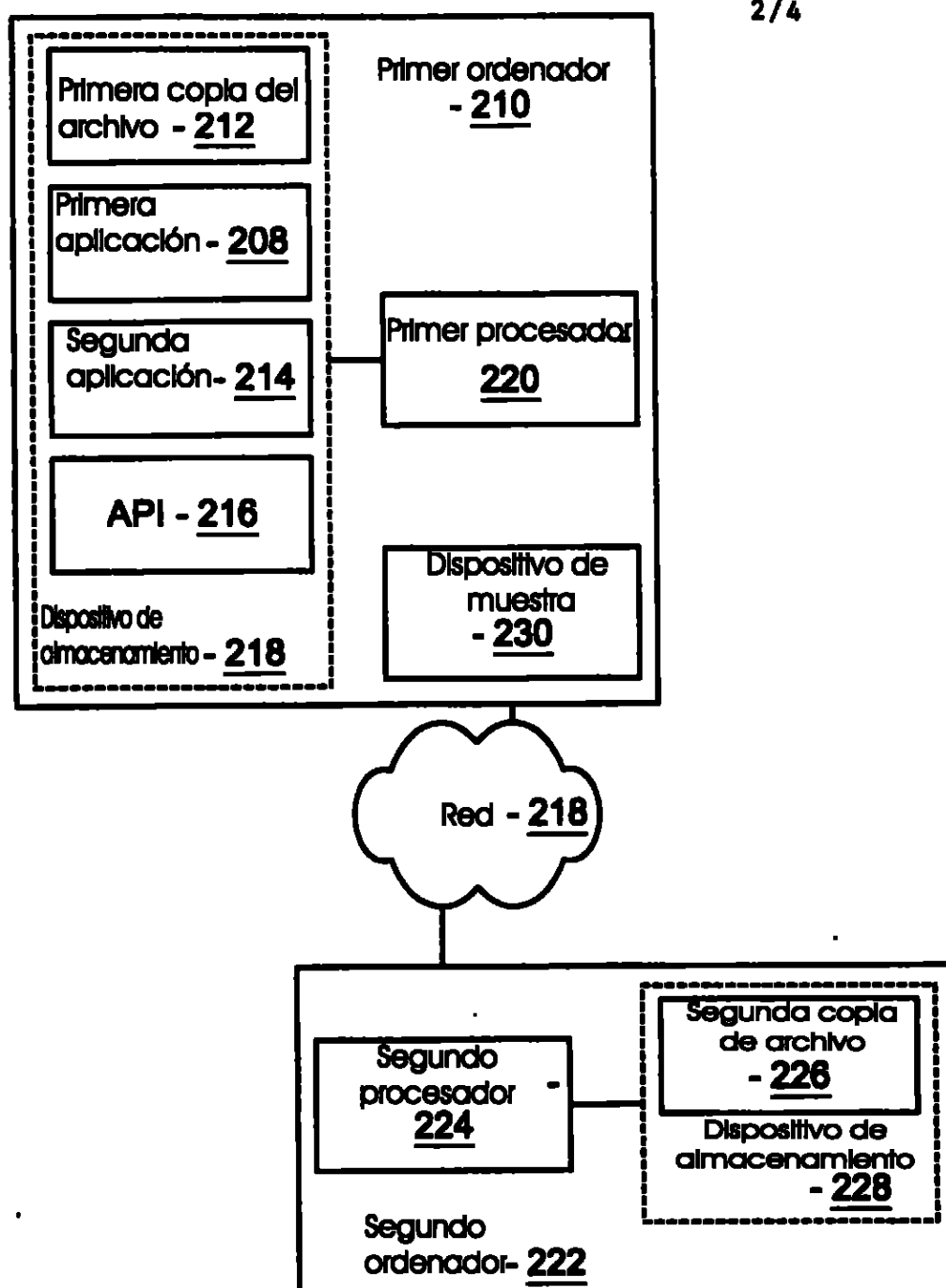


FIG. 2

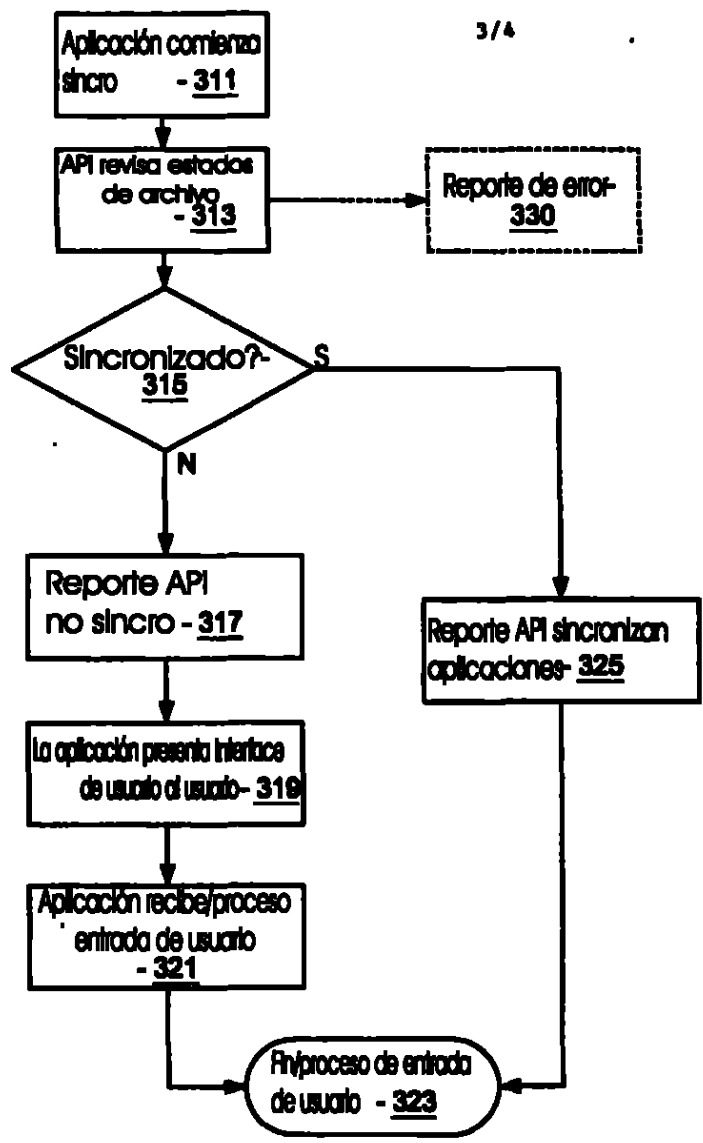


FIG. 3

70

4/4

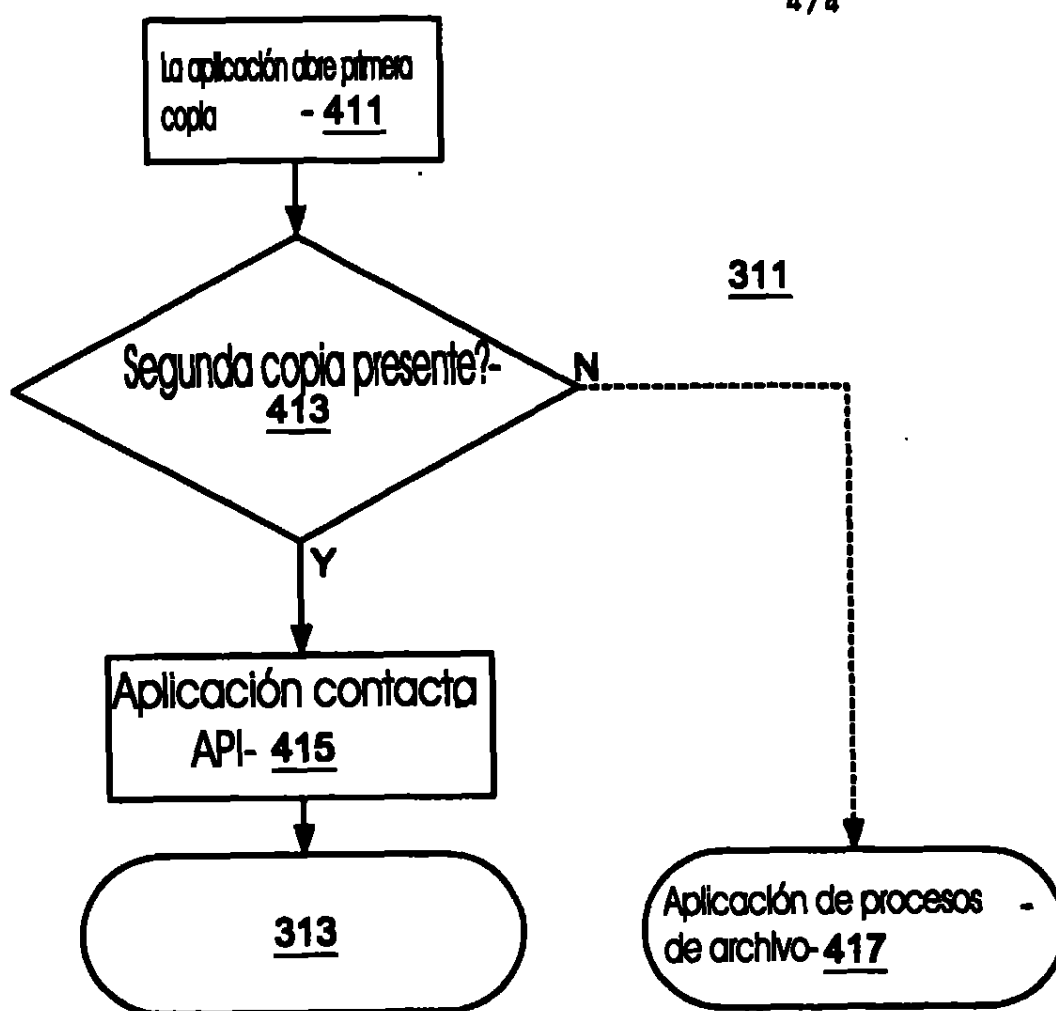


FIG. 4

CA12

Feb 19

12/22/03

METHOD FOR MANAGING FILE REPLICATION IN APPLICATIONS

RELATED APPLICATIONS

This patent application relates to commonly assigned copending U.S. Patent Application No. AA/BBB,CCC, filed February 27, 2003 under attorney docket number MSFT-1586 and entitled "Method and System for Managing Multiple File States."

FIELD OF THE INVENTION

The present invention is directed to methods for managing multiple versions of an electronic file. More particularly, the present invention relates to methods for using relevant application functionality to reconcile versions of a file during a synchronization event.

BACKGROUND OF THE INVENTION

It is common practice in a networked computer environment for a computer user to collaborate on an electronic document with another user on the network. Such a network may be a Local Area Network (LAN), Wide Area Network (WAN), the Internet or the like. In such a collaboration, each user may create and edit an electronic file by using a software application such as general application software, or software designed specifically for the type of file that is the object of the collaboration, or the like.

The collaboration may be made possible by maintaining a master version of the electronic file in a location that is accessible to all users who will be collaborating on the file. In a conventional networked computer environment, there may exist a server computer and one or

more user computers. The server maintains communications links between, and is accessible to, each user computer and therefore may store and control the master version of the file being collaborated on by the users. When collaborating on an electronic file, a user may first access the master version of the file on the server, and then copy the file to the user's computer. The user may then work on the file by altering the file in any way made available to the user by the application. For example, if the application is a word processor, spreadsheet, database, and/or the like, the application may provide means for performing tasks such as editing, copying, deleting and/or the like. At the completion of any alterations, the user may save changes to the version of the file stored on the user's computer if the user wishes to retain such alterations to the file. In other systems, such as a system where no server computer exists, the master version may be stored on a user computer.

In order to collaborate with other users via the server, a user with a changed version stored on the user's computer must synchronize the version of the file on the computer with the master version of the file on the server. Synchronizing a plurality of versions of an electronic file, in its simplest form, ensures that all versions, once synchronized, are identical. In this way, the user performing the synchronization will enable other users who are participating in the collaboration to access the newly-altered file. Conventionally, a synchronization is performed by a synchronization utility. A synchronization utility is typically separate from the application, and in many cases is part of the operating system of the user computer. The conventional synchronization utility performs synchronizations at predetermined times, or as requested by the operating system.

In a conventional synchronization, a synchronization utility will compare the user version of the file to the master version of the file using whatever information about the copies the utility is able to comprehend. Typically, such information is limited to the date and/or time of the last update of each version. If the date and/or time is identical, the synchronization utility determines that both versions are identical, and therefore synchronized, and that no further action is required. If a discrepancy exists between the date and/or time, however, the synchronization utility provides the user associated with the computer that is driving the synchronization event with a user interface. The user interface conventionally provides a user with a brief notification that there is a discrepancy, the date and/or time of such last update for each version of the file, and a request for an instruction from the user as to which copy of the file the user wishes to keep. If the user elects to keep the user version, the synchronization utility overwrites the master version with the user version, thereby making the user version the new master version on the server. If the user desires to keep the master version, no further action is taken, or the user

version is overwritten with the master version. In some user interfaces, a user is presented with an option to abort the synchronization, rather than choosing which version will be the master version.

The conventional synchronization as discussed above has significant disadvantages. Because a conventional synchronization utility has limited functionality with respect to an electronic file, a user interface generated by the conventional utility presents only a limited amount of information to a user when prompting a user to select a course of action. Such a limited amount of information may therefore cause such user to make an uninformed and possibly incorrect decision about which course of action to take. For example, a first user may alter a previously-synchronized user version of the file and then at some point the conventional synchronization utility initiates a synchronization event. When notified by the conventional user interface, the user may see that the user version was updated later than the master version and therefore decide to overwrite the master version. However, the information presented by the conventional user interface does not inform the user whether the master version has remained unchanged since the user began altering the user version. This leads to a problematic situation, where a second user may have made alterations to another user version, and then synchronized that user version with the master version before the first user's synchronization. The first user's version is more recent – having been updated later – but the master version has undergone alterations by the second user that were not present when the first user originally copied the file from the server. As a result, the second user's alterations to the master version are lost once the first user completes the synchronization.

The above situation is caused by a conventional synchronization utility's lack of the full range of functionality inherent in the application relevant to the file, so the utility is unable to use that functionality during a synchronization event. Therefore, the utility can only provide to a user a very limited amount of information regarding the different versions of the electronic file, such as, for example, the date and/or time of the last update to each version of the file as discussed above.

Another disadvantage caused by the limited functionality of conventional synchronization utilities is the lack of multiple courses of action available to a user during a synchronization. For example, in the example discussed above, the user interface only presents three options: update the master copy with the user copy, update the user copy with the master copy, or abort.

A further disadvantage of a conventional synchronization utility stems from a lack of integration of such utility with such application. For example, and as discussed above, a

conventional synchronization utility is part of an operating system of a user or server computer. Therefore, the synchronization process, and any applicable user interface, is conventionally conducted outside of the application user experience. This may lead to confusion among users who – while such users may be proficient with the application – are not proficient with an operating system utility. Additionally, and as a result of the lack of integration, a synchronization may be initiated by the synchronization utility at a time that is not optimal for such event to take place. An additional disadvantage of conventional synchronization utilities is the inability of such a utility to be used as a component of a software application. For example, using conventional methods, a computer programmer is unable to create a first software application that uses a synchronization utility that as the ability to use a second application appropriate to the electronic file to reconcile multiple versions of the electronic file. Therefore, what is needed is an efficient means of using relevant software application functionality to resolve conflicts between copies of a file during a synchronization.

SUMMARY OF THE INVENTION

In view of the foregoing, the present invention provides methods of integrating an application program interface with a software application that allows the application to request and control a file synchronization process while remaining in the application user experience. The invention provides for a method for managing a first and second version of an electronic file. In the method, first and second versions of an electronic file are recognized and a synchronization event is initiated using a first software application operating on a first computer. An application program interface accesses the first and second versions, compares information associated with each, generates a result and reports the result to the first software application. A second software application generates a user interface in accordance with a feature of the second software application, where the electronic file is associated with the second software application, and the user interface is displayed on a display device.

The invention also provides for the second software application to generate a synchronization option and to present the option with the user interface and to display the user interface the display device. The second software application may also receive an option selection and synchronizes the first and second versions of the electronic file accordingly. The invention further provides that the first software application and the second software application may be the same software application. The application may provide any amount of information to the user regarding the first and second versions, and the user may input any number or type of synchronization selections. In one embodiment, the user may create a third version comprising

any portion, or none, of either first or second version, or both. The application may initiate the synchronization at any time, such as at the user's request. Additional features and advantages of the invention are set forth in the description below.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing summary, as well as the following detailed description of preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings exemplary embodiments of the invention; however, the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

Figure 1 is a block diagram showing an exemplary computing environment in which aspects of the invention may be implemented;

Figure 2 is a functional diagram illustrating an example of a relationship between a first computer, a network and a second computer;

Figure 3 is a flowchart showing a synchronization process in accordance with one embodiment of the present invention; and

Figure 4 is a flowchart illustrating a method of initiating a synchronization process in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Overview

The present invention provides methods of integrating an application program interface with a software application that allows the application to request and control a file synchronization process while remaining in the application user experience. The invention provides for a method for managing a first and second version of an electronic file. In the method, first and second versions of an electronic file are recognized and a synchronization event to synchronize the first version on the first computer with the second version is initiated using a first software application operating on a first computer. An application program interface accesses the first and second versions, information associated with each of the first and second versions is compared and a result is generated. The result of the comparison is reported to the first software application. A second software application is caused to generate a user interface in accordance with a feature of the second software application, wherein the electronic

file is associated with the second software application, and the user interface is displayed on a display device.

The invention also provides for the second software application to generate a synchronization option and to cause the synchronization option to be presented by the user interface and displayed on the display device. The second software application also receives an option selection and synchronizes the first and second versions of the electronic file according to the option selection. The invention further provides that the first software application and the second software application may be the same software application, and therefore all steps may be performed by a single application.

The application may provide any amount of information to the user regarding the first and second versions, and the user may input any number or type of synchronization selections. In one embodiment, the user may create a third version comprising any portion, or none, of either first or second version, or both. The application may initiate the synchronization at any time, such as at the user's request.

Exemplary Computing Environment

Figure 1 illustrates an example of a suitable computing system environment 100 in which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices

that are linked through a communications network or other data transmission medium. In a distributed computing environment, program modules and other data may be located in both local and remote computer storage media including memory storage devices.

With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus (also known as Mezzanine bus).

Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and non-volatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer storage media in the form of volatile and/or non-volatile memory such as ROM 131 and RAM 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Figure 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

The computer 110 may also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example only, Figure 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, non-volatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, non-volatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, non-volatile optical disk 156, such as a CD-ROM or other optical media. Other removable/non-removable, volatile/non-volatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

The drives and their associated computer storage media, discussed above and illustrated in Figure 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Figure 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 110 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or

a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in Figure 1. The logical connections depicted include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Figure 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Exemplary Distributed Computing Frameworks Or Architectures

Various distributed computing frameworks have been and are being developed in light of the convergence of personal computing and the Internet. Individuals and business users alike are provided with a seamlessly interoperable and web-enabled interface for applications and computing devices, making computing activities increasingly web browser or network-oriented.

For example, MICROSOFT®'s .NET platform includes servers, building-block services, such as web-based data storage, and downloadable device software. Generally speaking, the .NET platform provides (1) the ability to make the entire range of computing devices work together and to have user information automatically updated and synchronized on all of them, (2) increased interactive capability for web sites, enabled by greater use of XML rather than HTML,

(3) online services that feature customized access and delivery of products and services to the user from a central starting point for the management of various applications, such as e-mail, for example, or software, such as Office .NET, (4) centralized data storage, which will increase efficiency and ease of access to information, as well as synchronization of information among users and devices, (5) the ability to integrate various communications media, such as e-mail, faxes, and telephones, (6) for developers, the ability to create reusable modules, thereby increasing productivity and reducing the number of programming errors, and (7) many other cross-platform integration features as well.

While exemplary embodiments herein are described in connection with software residing on a computing device, one or more portions of the invention may also be implemented via an operating system, API, or a "middle man" object between a coprocessor and requesting object, such that services may be performed by, supported in, or accessed via all of .NET's languages and services, and in other distributed computing frameworks as well.

Exemplary Embodiments

When an electronic file has been formed into more than one version, according to the invention, an application program interface may be configured to allow a software application associated with the file to control a process of synchronizing the plurality of versions. In one embodiment, the application may initiate the synchronization automatically, while in another embodiment, the application may initiate the synchronization based on user input. In another embodiment, the application presents synchronization options provided by application functionality to the user, and permits the user to select options and perform tasks using the application functionality. In yet another embodiment, a first application initiates the synchronization, while a second application presents synchronization options to a user.

Referring now to Figure 2, a first computer 210 comprises a first processor 220 in operative communication with a storage device 218. First computer 210 may be a general purpose desktop or portable computer, a workstation, a specialized computing device and/or the like. Likewise, first processor 220 may be a general purpose computer processor, a specialized processor, a network of processors or the like. Operatively associated with first computer 210 is display device 230. Display device 230 may be any means of presenting options and the like to a user, such as for example a monitor, speakers or the like. Such operative association of display device 230 to first computer 210 may be made by any wired or wireless connective means. While display device 230 is shown to be within first computer 210 in Figure 2, display device

230 may be located anywhere that is in operative communications with first processor 220, to include a remote location or the like.

Storage device 218 may be a hard drive, CD-ROM, magnetic disk, DVD, RAM and/or the like. Like display device 230, storage device 218 may be located anywhere that is in operative communications with first processor 220. Operative communications in such a situation may be wired, wireless or any combination thereof. Contained within storage device 218 is a first version 212 of an electronic file. Such first version 212 may be any type of electronic file, such as for example a word processing document, spreadsheet, database and/or the like.

First application 208 and second application 214 may be stored in storage device 218. First application 208 and second application 214 may be any type of computer-executable instructions, such as for example a word processor, spreadsheet, database, programmed script, specialized software or the like. In embodiments where both first application 208 and second application 214 are present, as discussed below, only second application 214 need be compatible with the electronic file. Alternatively, and as will also be discussed below, first application 208 and second application 214 may be the same application. One or both of first application 208 and second application 214 may also be located outside of storage device 218, as either application need only be in operative communications with first computer 210.

Application program interface 216 is any type of computer-executable instructions adapted to check and compare two or more copies of an electronic file for synchronization purposes, and may be stand alone software that is specifically designed for synchronizing two or more copies of an electronic document, a software utility that operates as a component of another application, an integrated component of first or second application 208, 214 and/or the like. Application program interface 216 may also perform additional tasks that may or may not be related to synchronization. First version 212, first application 208, second application 214 and application program interface 216 may reside in the same storage device 218, or within or across any number different storage devices 218.

Second computer 222 comprises a second processor 224 and storage device 228 which contains a second version 226 of the electronic file discussed above in connection with first version 212. As discussed above, storage device 228 may be located anywhere that is in operative communications with second computer 222, to include a remote location or the like. Also, storage device 228 and storage device 218 may, in an embodiment, be the same storage device. Operatively connecting first and second computers 210, 222 is network 218. Network 218 may be any communications network adapted for such operative communications, such as

for example a LAN, WAN, wireless network or the like. As may be appreciated, any number of first computers 210 and second computers 222 may be operatively connected to network 218.

Turning now to Figure 3, at step 311 a synchronization process begins. In one embodiment, first application 208 initiates the synchronization process, and second application 214 carries out the synchronization process. In another embodiment, first application 208 and second application 214 are a single application, in which case the single application both initiates and carries out the synchronization. While the discussion of Figure 3 that follows describes a process involving both first application 208 and second application 214, it may be appreciated that a single application may alone perform all of the steps performed by first application 208 and second application 214.

Any number of situations may cause first application 208 to initiate a synchronization process, such as for example: opening, closing, saving or the like of first version 212; upon recognizing a first and second version of the electronic file; manually initiating upon a user request; initiating at timed intervals; upon a user logging into a system and/or the like. In addition, first application 208 may impose any additional conditions or may use substitute conditions such as, for example, only initiating a synchronization when a certain number of versions are present. More details with respect to initiating a synchronization event in one embodiment follows below in connection with Figure 4. To initiate the process, first application 208 causes application program interface 216 to access first version 212 and second version 226. Accessing first version 212 and second version 226 may be performed by any computing means, such as by viewing information associated with each version, finding each version by using searching functionality, locating each version by using identifying information provided by first application 208 or the like.

First application 208 may provide interface 216 with any amount of identifying information about second version 226, such as for example a file name, location, address and/or the like. Interface 216 may be adapted to actively locate second version 226, or may have limited functionality that permits only looking at a location specified by first application 208, first version 212 or the like. Also, interface 216 may be configured to perform any number of additional tasks that may even be unrelated to the synchronization process. At step 313, application program interface 216 locates second version 226 and checks for a synchronization status. The application program interface 216 determines the status by comparing information, such as for example time and/or date of a last update or the like, for first version 212 against second version 226. Synchronization status may also be determined by any criteria additional to or substituted for the information mentioned above. If the information matches information

and/or criteria specified by first application 208, first and second versions 212, 226 have a synchronized status. If the information is inconsistent, first and second versions 212, 226 do not have a synchronized status, as such an inconsistency indicates the possibility that changes may have been made to one of such versions 212, 226. Alternatively, if application program interface 216 detects an error, a condition not provided for, a condition that first application 208 has specified as cause for terminating the synchronization, or the like, then interface 216 proceeds to step 330. At step 330, interface 216 informs first application 208 of the condition and first application 208 proceeds accordingly, which may include any type of action or inaction.

At step 315, application program interface 216 may take one of two courses of action depending on the outcome of the synchronization determination of step 313. If application program interface 216 has determined that first version 212 and second version 226 are synchronized, interface 216 proceeds to step 325. At step 325, application program interface reports to first application 208 that first version 212 and second version 226 are synchronized. Such a report indicates to first application 208 that no further synchronization activities are presently necessary, as first version 212 comprises the most recent version of the electronic file. The report may also indicate that a user may use first application 208 or second application 214 to alter first version 212.

At step 323, the synchronization process ends. At this step, first application 208 may cease functioning, such as for example if such synchronization process took place when first application 208 was closing first version 212 of the electronic document. In such a situation, once synchronization has occurred, therefore, first application 208 has no more functions to perform and may terminate manually by user or automatically. In another embodiment, first application 208 initiates the synchronization process when opening a first version 212. In such a situation, application would not terminate at the completion of such synchronization event, and rather would permit user to begin altering first version 212 using first application 208 functionality, if available.

If, at step 315, however, application program interface 216 has determined that first version 212 and second version 226 are not synchronized, interface 216 proceeds to step 317. At step 317, application program interface 216 reports to first application 208 and/or second application 214 that first version 212 and second version 226 are not synchronized. Such a report causes second application 214 to begin synchronizing first version 212 and second version 226 with user input as discussed below in connection with steps 319 and 321.

At step 319, second application 214 presents a user interface to a user by displaying the user interface on a display device 230. Such user interface may be in the form of a text and/or

graphics window, audio prompt and/or the like, and any such configuration of a user interface is equally consistent with an embodiment of the present invention. In presenting such user interface, second application 214 may present one or more options to a user using second application 214 functionality or the like. For example, if second application 214 is a word processor, second application 214 may present a synchronization option to a user within a word processing user environment of second application 214, such as for example an editing window with a page view of such first version 212 and/or the like. Also, second application 214 may highlight differences between first version 212 and second version 226 using color, sounds, graphics and the like to better enable user to make synchronization decisions. Second application 214 may also provide additional information, such as for example the identity of each user who altered either version 212, 226, and may only permit a user to alter a version 212, 226 if such user has permission, in the form of a security clearance or the like, to do so. Second application 214 may enable user to, for example, merge versions 212, 226 into a third copy, so as to incorporate changes in both versions 212, 226. An additional option may also be in the form of guidance derived from second application 214 functionality such as for example a recommended course of action and/or the like. As may be appreciated, an option may be anything that may be generated by second application 214 using second application 214 functionality with respect to first version 212 and second version 226.

At step 321, second application 214 receives and processes user input. Such user input may comprise any amount of information regarding such synchronization. For example, user input may select a particular alteration of either first version 212 or second version 226 of the electronic file that the user wishes to retain. In another embodiment, user input may comprise a selection of either first version 212 or second version 226 that user wishes to retain in its entirety. In still another embodiment, user input may comprise edits of first and/or second version 212, 226 using the functionality of second application 214 that, when completed, results in synchronizing first version 212 and second version 226. As may be appreciated, user input may comprise any input made available by second application 214 for selection or use by a user, and any such input is consistent with an embodiment of the present invention.

As also may be appreciated, steps 319 and 321 may repeat, as such user interface may be redisplayed and/or refreshed depending on user input. For example, user interface may be reconfigured by second application 214 based on user input. Such a reconfiguration may occur if, for example, user selects one of a plurality of differences between first version 212 and second version 226 and second application 214 redisplay user interface to user to reflect such a selection. Steps 319 and 321 may repeat any number of times, or not at all, and either such

situation is equally consistent with an embodiment of the present invention. Also, second application 214 may report any results of the synchronization to first application 208 or to interface 216. The results reported by second application 214 may be any information associated with the synchronization, such as the option selection(s), synchronization result(s) or the like.

Second application 214 may proceed to step 323, the end of the synchronization, when a triggering event occurs. Such triggering event may occur at any time, such as for example when user has completed reconciling all differences between first version 212 and second version 226, when user directs second application 214 to abort or terminate the synchronization process without completing synchronization, at a predefined time, or the like. At step 323, the synchronization ends. First application 208 or second application 214 may then, as mentioned above in connection with step 325, take any action at step 323 that is consistent with the context in which the synchronization took place.

Turning now to Figure 4, a detailed view of step 311 is shown in accordance with one embodiment of the present invention. As discussed above in connection with Figure 3, in step 311 first application 208 initiates a synchronization process. Figure 4 illustrates with greater particularity one embodiment of such an initiation step, where first application 208 initiates such synchronization process upon opening first version 212 of an electronic file. As discussed above, first application 208 may initiate such synchronization process at any time, such as for example when opening, saving or closing such first version 212, at set or variable intervals, upon user inputs and/or the like.

At step 411, first application 208 opens first version 212 of an electronic file. Such opening will most commonly occur upon user input to open such first version 212, but any means for initiating such opening of first version 212 is consistent with an embodiment of the present invention. At step 413, first application 208 determines whether first version 212 has an associated second version 226. first application 208 may proceed to step 413 upon completely opening first version 212 in step 411, or may proceed to step 413 at any point during such opening process including, for example, before opening first version 212 at step 411. Also, any method may be employed to enable first application 208 to make such a determination. For example, information such as an electronic stamp or the like may be associated with such first version 212 thereby enabling first application 208 to make such determination. In contrast, first application 208 may search any or all computing devices operatively connected to network 218, such as for example second computer 222, for such determination. As may be appreciated, first application 208 may use any method of determining the existence of second version 226 while remaining consistent with an embodiment of the present invention.

If, at step 413, first application 208 does not find second version 226 first application 208 determines that first version 212 is the only copy of such electronic file and proceeds to step 417. At step 417 first application 208 processes first version 212 normally. Such normal processing may be any type of processing that first application 208 typically performs on electronic files of the same format as first version 212, possibly in accordance with any user inputs. For example, in Figure 4 first application 208 was in the process of opening first version 212 when first application 208 initiated such synchronization process according to step 311. Upon finding no need for a synchronization – because of the absence of second version 226 – first application 208 completes opening of first version 212 and proceeds accordingly.

If, at step 413, first application 208 finds second version 226 first application 208 proceeds to step 415. At step 415, first application 208 contacts application program interface 216 to commence synchronization of first version 212 and second version 226. Upon contacting application program interface 216, such synchronization process proceeds to step 313 as discussed above in connection with Figure 3.

Thus, methods and apparatus for managing file replication in applications have been presented. While the present invention has been described in connection with the preferred embodiments of the various figures, it is to be understood that other similar embodiments may be used or modifications and additions may be made to the described embodiment for performing the same function of the present invention without deviating therefrom. For example, one skilled in the art will recognize that the present invention as described in the present application may apply to any computing device or environment, whether wired or wireless, and may be applied to any number of such computing devices connected via a communications network, and interacting across the network. Furthermore, it should be emphasized that a variety of computer platforms, including handheld device operating systems and other application specific operating systems are contemplated, especially as the number of wireless networked devices continues to proliferate. Still further, the present invention may be implemented in or across a plurality of processing chips or devices, and storage may similarly be affected across a plurality of devices. Therefore, the present invention should not be limited to any single embodiment, but rather should be construed in breadth and scope in accordance with the appended claims.

What is Claimed:

1. A method for managing a first and second version of an electronic file, the method comprising:

initiating, using a first software application operating on a first computer, a synchronization event to synchronize the first version on the first computer with the second version;

causing an application program interface to access said first and second versions;
comparing information associated with each of said first and second versions and generating a result;

reporting the result of the comparison to the first software application;

causing a second software application to generate a user interface in accordance with a feature of the second software application, wherein the electronic file is associated with the second software application; and

displaying the user interface on a display device.

2. The method of claim 1, wherein the first and second software applications are the same application.

3. The method of claim 1, wherein the result of said comparing step is a synchronization status of the first and second versions.

4. The method of claim 1, wherein the second version is stored on a second computer, and said first and second computers are operatively connected by a communications network.

5. The method of claim 1, further comprising the step of when the result of the comparison indicates that the first and second versions of the electronic file are not synchronized, said generating step further comprises presenting, using the second software application, synchronization options.

6. The method of claim 4, wherein said step of presenting synchronization options further comprises:

generating a synchronization option by way of the second software application;

causing the synchronization option to be presented by the user interface and displayed on the display device;

receiving an option selection; and
synchronizing the first and second versions of the electronic file according to the option selection.

7. The method of claim 6, wherein the step of synchronizing the first and second versions comprises overwriting the second versions of the electronic file with the second version of the electronic file.

8. The method of claim 6, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions with a third version of the electronic file, said third version comprising a portion of the first version of the electronic file.

9. The method of claim 6, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions with a third version of the electronic file, said third version comprising a portion of the second version of the electronic file.

10. The method of claim 6, wherein the step of synchronizing the first and second versions comprises merging the first and second versions into a third version according to input received from the user interface and overwriting the first and second versions with the third version.

11. The method of claim 1, further comprising:
opening the first and second versions using the second software application;
presenting a synchronization option using the user interface, the synchronization option generated by the second software application;
receiving user input from the user interface;
processing the user input using the second software application; and
synchronizing the first and second versions according to the user input.

12. The method of claim 11, wherein said step of presenting a synchronization option further comprises displaying the first and second copies in the user interface.

13. The method of claim 11, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions according to user input.

14. The method of claim 1, wherein the step of initiating the synchronization event occurs upon the first software application detecting information associated with the first version of the electronic file indicative of the second version of the electronic file.
15. The method of claim 1, wherein the step of initiating the synchronization event occurs upon the first software application opening the first version of the electronic file.
16. The method of claim 1, wherein the step of initiating the synchronization event occurs upon the first software application saving the first version of the electronic file to a storage device of the first computer.
17. The method of claim 1, wherein the step of initiating the synchronization event occurs upon the first software application recognizing the first version and second version.
18. A method for managing a first and second version of an electronic file, the method comprising:
receiving a request from a first software application to access and compare said first and second versions;
accessing said first and second versions;
comparing information associated with each of said first and second versions;
determining if said first and second versions are synchronized and generating a result;
and
reporting the result of the comparison to the first software application.
19. The method of claim 18, wherein said step of receiving a request further comprises receiving identifying information corresponding to said first and second versions.
20. The method of claim 18, wherein the second version is stored on a second computer, and said first and second computers are operatively connected by a communications network.
21. The method of claim 18, further comprising:
receiving a second request from the first software application, said second request to access said first and second versions and to perform a task related to said first and second versions; and

accessing said first and second versions and performing said task.

22. The method of claim 18, further comprising:

receiving a request from a second software application, the second software application associated with the electronic file, to overwrite the first and second versions according to user input received by the second software application; and
overwriting the first and second versions.

23. The method of claim 18, wherein the first and second software applications are the same application.

24. A computer-readable medium having computer-executable instructions for performing a method for managing a first and second version of an electronic file, the method comprising:

initiating, using a first software application operating on a first computer, a synchronization event to synchronize the first version on the first computer with the second version;

causing an application program interface to access said first and second versions;
comparing information associated with each of said first and second versions and
generating a result;

reporting the result of the comparison to the first software application;

causing a second software application to generate a user interface in accordance with a feature of the second software application, wherein the electronic file is associated with the second software application; and

displaying the user interface on a display device.

25. The computer-readable medium of claim 24, wherein the first and second software applications are the same application.

26. The computer-readable medium of claim 24, wherein the result of said comparing step is a synchronization status of the first and second versions.

27. The computer-readable medium of claim 24, wherein the second version is stored on a second computer, and said first and second computers are operatively connected by a communications network.

28. The computer-readable medium of claim 24, wherein the method further comprises the step of when the result of the comparison indicates that the first and second versions of the electronic file are not synchronized, said generating step further comprises presenting, using the second software application, synchronization options.

29. The computer-readable medium of claim 27, wherein said step of presenting synchronization options further comprises:

generating a synchronization option by way of the second software application;
causing the synchronization option to be presented by the user interface and displayed on the display device;
receiving an option selection; and
synchronizing the first and second versions of the electronic file according to the option selection.

30. The computer-readable medium of claim 29, wherein the step of synchronizing the first and second versions comprises overwriting the second versions of the electronic file with the second version of the electronic file.

31. The computer-readable medium of claim 29, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions with a third version of the electronic file, said third version comprising a portion of the first version of the electronic file.

32. The computer-readable medium of claim 29, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions with a third version of the electronic file, said third version comprising a portion of the second version of the electronic file.

33. The computer-readable medium of claim 29, wherein the step of synchronizing the first and second versions comprises merging the first and second versions into a third version according to input received from the user interface and overwriting the first and second versions with the third version.

34. The computer-readable medium of claim 24, wherein the method further comprises:
opening the first and second versions using the second software application;
presenting a synchronization option using the user interface, the synchronization option generated by the second software application;
receiving user input from the user interface;
processing the user input using the second software application; and
synchronizing the first and second versions according to the user input.
35. The computer-readable medium of claim 34, wherein said step of presenting a synchronization option further comprises displaying the first and second copies in the user interface.
36. The computer-readable medium of claim 34, wherein the step of synchronizing the first and second versions comprises overwriting the first and second versions according to user input.
37. The computer-readable medium of claim 24, wherein the step of initiating the synchronization event occurs upon the first software application detecting information associated with the first version of the electronic file indicative of the second version of the electronic file.
38. The computer-readable medium of claim 24, wherein the step of initiating the synchronization event occurs upon the first software application opening the first version of the electronic file.
39. The computer-readable medium of claim 24, wherein the step of initiating the synchronization event occurs upon the first software application saving the first version of the electronic file to a storage device of the first computer.
40. The computer-readable medium of claim 24, wherein the step of initiating the synchronization event occurs upon the first software application recognizing the first version and second version.
41. A computer-readable medium having computer-executable instructions for managing a first and second version of an electronic file, the method comprising:

receiving a request from a first software application to access and compare said first and second versions;

accessing said first and second versions;

comparing information associated with each of said first and second versions;

determining if said first and second versions are synchronized and generating a result;

and

reporting the result of the comparison to the first software application.

42. The computer-readable medium of claim 41, wherein said step of receiving a request further comprises receiving identifying information corresponding to said first and second versions.

43. The computer-readable medium of claim 41, wherein the second version is stored on a second computer, and said first and second computers are operatively connected by a communications network.

44. The computer-readable medium of claim 41, wherein the method further comprises:
receiving a second request from the first software application, said second request to access said first and second versions and to perform a task related to said first and second versions; and

accessing said first and second versions and performing said task.

45. The computer-readable medium of claim 41, wherein the method further comprises:
receiving a request from a second software application, the second software application associated with the electronic file, to overwrite the first and second versions according to user input received by the second software application; and
overwriting the first and second versions.

46. The computer-readable medium of claim 41, wherein the first and second software applications are the same application.

ABSTRACT

A method for managing a first and second version of an electronic file is disclosed. In the method, first and second versions of an electronic file are recognized and a synchronization event is initiated. An application program interface accesses the first and second versions, compares information associated with each, generates a result and reports the result to the first software application. A second software application generates a user interface in accordance with a feature of the second software application, where the electronic file is associated with the second software application, and the user interface is displayed on a display device.

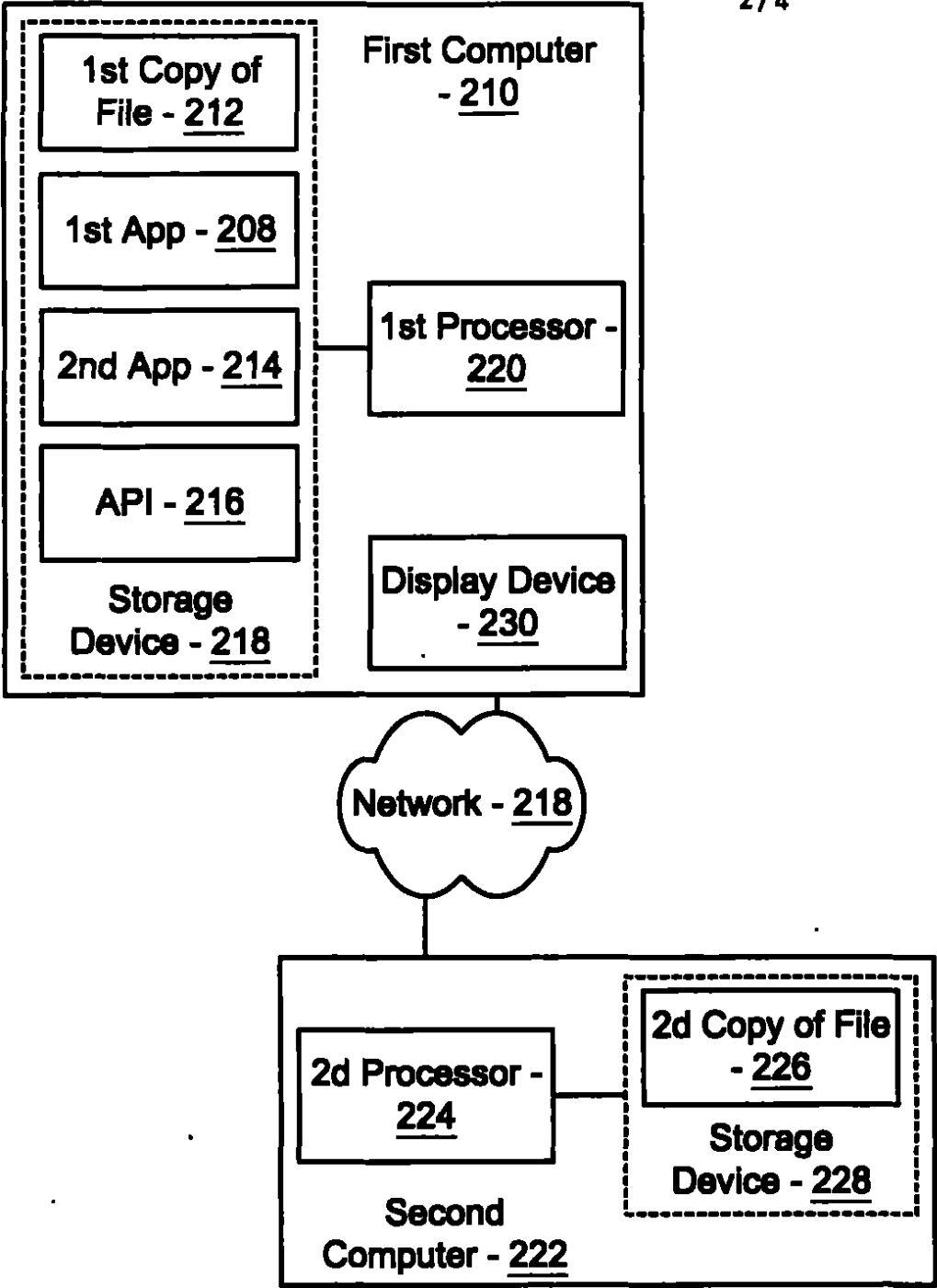


FIGURE 2

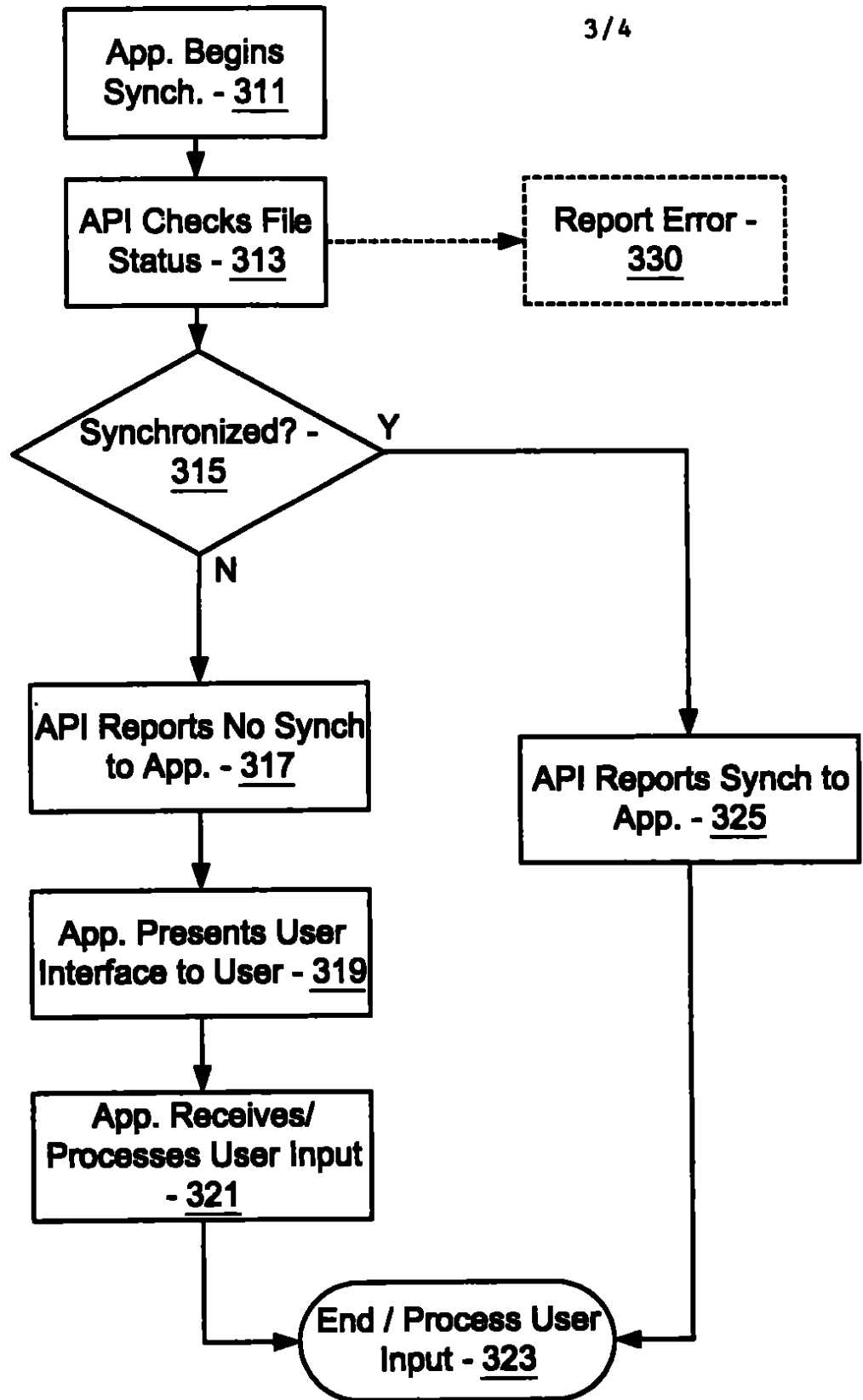


FIGURE 3

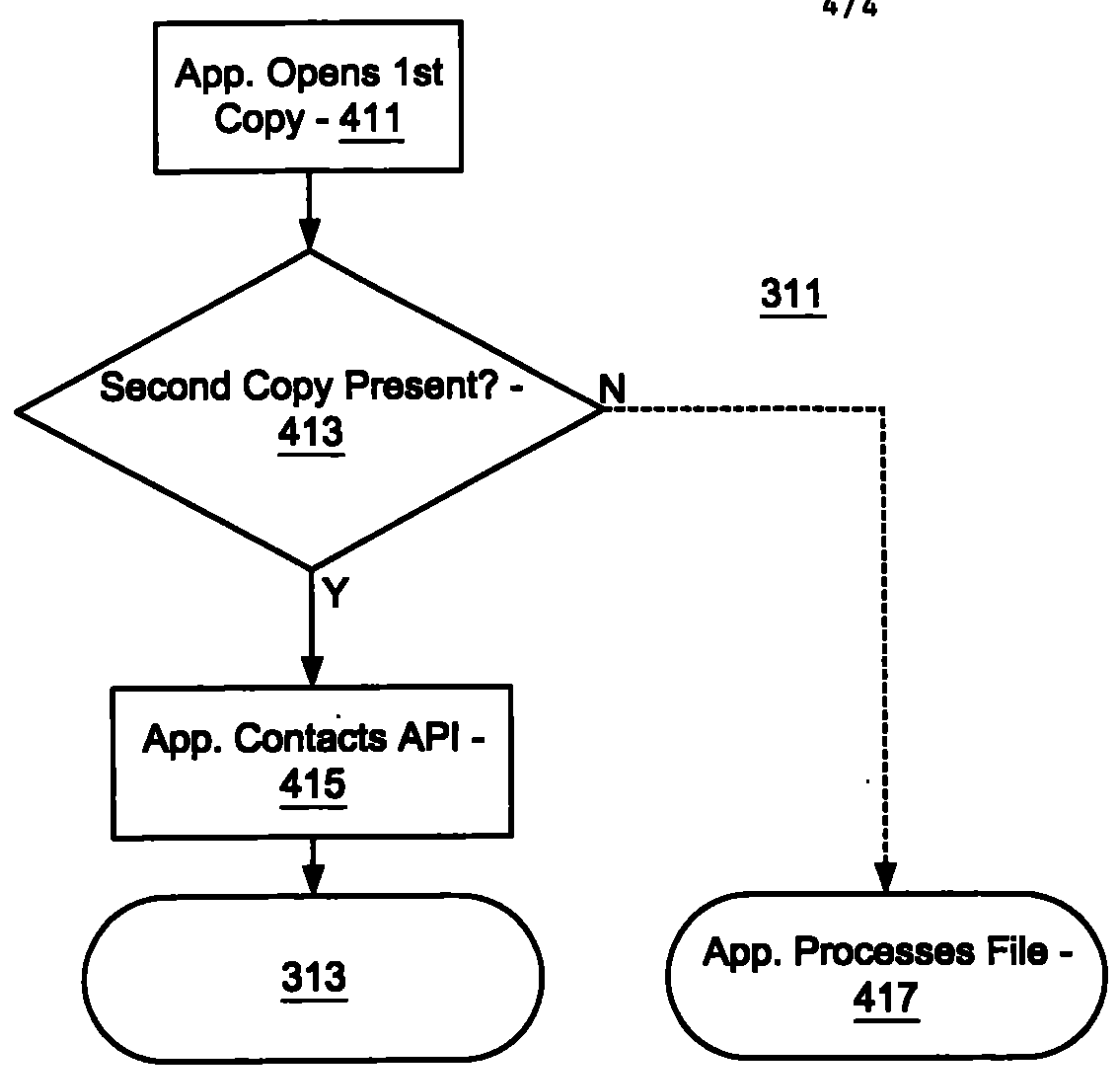


FIGURE 4

SUPERINTENDENCIA Y COMERCIO
Radicación: 04016523 000000000
Fecha (MM/AA): 2004-02-25 14:44:50
Trámite: 002 PATENTES O DISEÑO INDUSTRIAL Y MARCA
Dependencia: 2020 DIVISION DE MARCAS Y DISEÑOS

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

	USUARIO	RADICADOR
PATENTE DE INVENCION	()	()
MODELO DE UTILIDAD	()	()
Indicación de que se solicita la concesión de una patente	()	()
Datos de identificación del solicitante o de la persona que se presenta la solicitud	()	()
Descripción de la invención	()	()
Dibujos de ser estos pertinentes.	()	()
Comprobante de Pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	()
DISEÑO INDUSTRIAL ()		
Indicación de que se solicita el registro de Diseño Industrial	()	()
Datos de identificación del solicitante o de la persona que presenta la solicitud	()	()
Representación gráfica y fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño	()	()
Comprobante de pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	()
ESQUEMA DE TRAZADO ()		
Indicación de que solicita el registro de un Esquema de trazado	()	()
Datos de identificación del solicitante o de la persona que presenta La solicitud	()	()
Representación gráfica del esquema de trazado	()	()
Comprobante de pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	()

Firma Responsable *Claudia M Nueva*

Fecha

[Faint handwritten notes and stamps at the bottom of the page]



[Handwritten signature]

Industria y Comercio
SUPERINTENDENCIA

PROTOCOLO

Expediente 04-16523

SUPERINDUSTRIA Y COMERCIO SISTEMA DE REGISTRO

MANTENIMIENTO DE PROTOCOLOS

Numero : 16069

Fecha : 04/08/1995

Conferido: MICROSOFT CORPORATION

Pais : US ESTADOS UNIDOS DE AMERICA

Ciudad : 234 REDMOND, WASHINGTON

Represent: S Sustituir: S Desistir: S

No. Radicacion: 95 34920 Clase: 0 Seev: 0 Seac: 0

scncia Codigo Ctr

Nombre

1 77 A CARDENAS NAVAS DARIO

2 3853 CARDENAS CABALLERO EDUARDO

3 5048 CARDENAS MARTINEZ BERNARDO

Encontrados (1/1) registro(s)

Carrera 13 No. 27-00 Pisos 5, 7 y 10

Conmutador: 382 0840

Fax: 350 5220

E-mail: info sic.gov.co

www.sic.gov.co

Bogotá, D.C., Colombia



101

Industria y Comercio
SUPERINTENDENCIA~~Folio 100~~

2020
Bogotá, D. C.

Doctor(a)
DARIO CARDENAS NAVAS
Apoderado(a)

Oficio No. 63417

REFERENCIA:	Radicación	04 16523
	Trámite	002
	Evento	001
	Actuación	430
	Folios	001

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, se le comunica que:

- Allegar copia del documento en que conste la cesión del derecho a la patente otorgado por el(los) inventor(es) al solicitante o a su causante. (art. 26-k, Dec 486/2000).

En cumplimiento del artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos (2) meses siguientes a la fecha de notificación del presente oficio, complete los requisitos anteriormente enunciados. En caso de no dar cumplimiento al presente requerimiento, la solicitud se considerará abandonada y perderá su prelación.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 5.2, literal e), del capítulo quinto del título primero de la Circular Unica No.10 de 2001.

ALIX CARMENZA CESPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones

01 ABR 2004

El anterior requerimiento fue notificado por fijación en lista de fecha _____

Se recuerda al interesado que debe presentar, dentro de los dieciséis meses siguientes contados a partir de la fecha de presentación de la solicitud cuya prioridad se invoca, copia de la misma certificada por la autoridad que la expidió y, en los casos en que haya lugar, la traducción simple del capítulo reivindicatorio. El incumplimiento de este plazo, acarreará la pérdida de la prioridad invocada.

Carrera 13 No. 27-00 Pisos 5, 7 y 10

Conmutador: 382 0840

Fax: 350 5220

E-mail: info@sic.gov.co

www.sic.gov.co

Bogotá, D.C., Colombia

202027