



Industria y Comercio

S U P E R I N T E N D E N C I A

Delegatura de Propiedad Industrial .

División de Nuevas Creaciones

SOLICITUD

PATENTE DE INVENCION

PCT

21. EXPEDIENTE No.

04-61877

54. TÍTULO

"Shell" Del SISTEMA De Archivo

51. CLASIFICACIÓN INTERNACIONAL

G06F 17/30

71. SOLICITANTE

MICROSOFT CORPORATION

DOMICILIO

REDMOND, WASHINGTON; Estados Unidos De America

74. APODERADO

DARIO CARDENAS Nolas

22. BOGOTÁ, D. C.,

2004-VI-29

PETITORIO



Industria y Comercio
SUPERINTENDENCIA

SUPERINDUSTRIA Y COMERCIO

Radicación : 84661817 00000000

Folios: 480

Fecha (AMD): 2004-06-29 16:59:56

Trámite : 811 PCT-PATENT 378 FASE NACI 411 PRESENTAC

Dependencia: 2620 DIVISION DE NUEVAS CREACIONES

FORMULARIO UNICO DE SOLICITUD DE PATENTE
PCT
ENTRADA EN FASE NACIONAL

①

SOLICITUD DE:

☒

Patente de Invención

☐

Patente de Modelo de Utilidad

☒

Capítulo I

☐

Capítulo II

②

SOLICITANTE
(71)

Nombre: **MICROSOFT CORPORATION**

Dirección: One Microsoft Way Redmond, Washington 98052, Estados Unidos de America

Nacionalidad o Domicilio: REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA.

Lugar de Constitución: ESTADOS UNIDOS DE AMERICA

Teléfono: 3123600 Fax: 3122410

E-mail: general@cardenasycardenas.com

IDENTIFICACION

C.C. ☐NIT ☐C.E. ☐Otro ☒Cual _____
Número _____

③

REPRESENTANTE
O APODERADO

Nombre: **DARIO CARDENAS NAVAS**

Dirección: Carrera 7 No 71-52 Torre B Piso 9

Teléfono: 3123600 Fax: 3122410

E-Mail: general@cardenasycardenas.com

IDENTIFICACION

C.C. ☒NIT ☐C.E. ☐Otro ☐Cual _____
No 17.066.629 TP 1.250
BTA C.S.J.

④

INVENTOR(ES)
(72)

Nombre:

1. Jason F. Moore
2. David G. de Vorchik
3. Nathaniel H. Ballou
4. Richard M. Banks
5. Chris J. Guzak
6. Patrice L. Miner
7. David J. Sheldon
8. James B. Kurtz
9. Judson C. Hally
10. Tyler K. Beam

Dirección:

1. 431 Fifth Avenue S., Apt.2 Kirkland, Washington 98033, E.U.A.
2. 414 W Newell Street Seattle, WA 98119, E.U.A.
3. 11126 NE 104 th Way Kirkland, Washington 98033, E.U.A.
4. 9739 - 112th Avenue NE Kirkland, Washington 98033, E.U.A.
5. 8363 NE Juanita Drive Kirkland, Washington 98034, E.U.A.
6. 3719 - 101st Way NE Kirkland, Washington 98033, E.U.A.
7. 2319 - 1ST AVENUE, Suite 302 Seattle, Washington 98119, E.U.A.
8. 4338 Latona Seattle, Washington 98105, E.U.A.
9. 4226 - 205th Place NE Sammamish, Washington 98074, E.U.A.
10. 9200 Red-Wood Road NE Apt. D324 Redmond, Washington 98052, E.U.A.

Nacionalidad o Domicilio:

1. WASHINGTON, ESTADOS UNIDOS DE AMERICA
2. WASHINGTON, ESTADOS UNIDOS DE AMERICA
3. WASHINGTON, ESTADOS UNIDOS DE AMERICA
4. WASHINGTON, ESTADOS UNIDOS DE AMERICA
5. WASHINGTON, ESTADOS UNIDOS DE AMERICA
6. WASHINGTON, ESTADOS UNIDOS DE AMERICA
7. WASHINGTON, ESTADOS UNIDOS DE AMERICA
8. WASHINGTON, ESTADOS UNIDOS DE AMERICA
9. WASHINGTON, ESTADOS UNIDOS DE AMERICA
10. WASHINGTON, ESTADOS UNIDOS DE AMERICA

⑤

PCT (86)

Solicitud Internacional No. PCT/US03/15610Fecha Mayo 16, 2003

Publicación Internacional No. _____

Fecha _____

⑥

Título(54):

"SHELL" DEL SISTEMA DE ARCHIVOS

*se refiere al espacio donde se manejan las instrucciones a ejecutar, a nivel del sistema operativo, y en este caso, al navegador que permite recorrer el sistema de archivos"

⑦

Clasificación Internacional (51):

G06F 17/00

⑧

Prioridad

(33) País de Origen

(32) Fecha

(31) N° de Solicitud

SI ☐ NO ☒

Comprobante de Pago No.:

04-50.356

Fecha

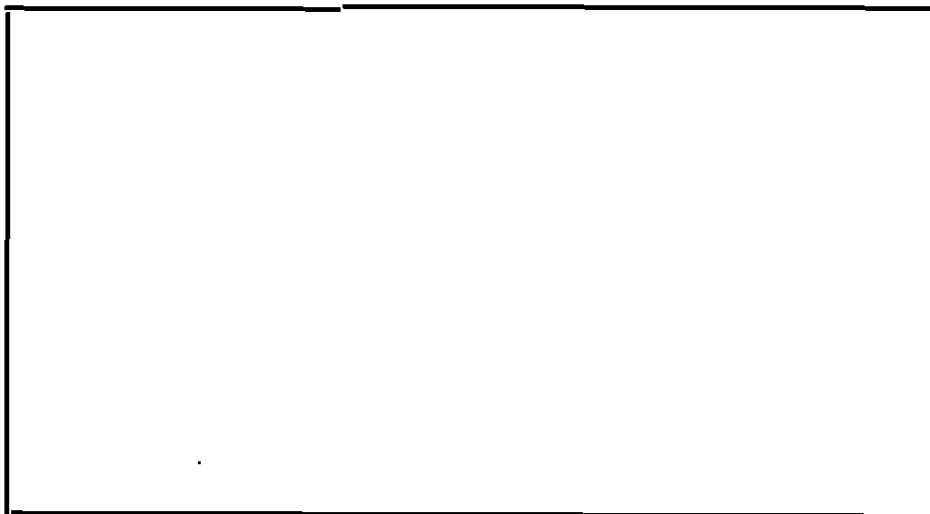
Junio 25, 2004

⑩

ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud.
- ☒ Documento que acredita la existencia y representación legal cuando el solicitante sea persona jurídica. (Protocolo 16.069)
- ☒ Poderes, si fuere el caso. (Protocolo 16.069)
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☐ Declaraciones
- ☒ Copia de la solicitud internacional. (Resumen, descripción, reivindicaciones, dibujos)
- ☒ Traducción de la solicitud internacional al castellano. (Resumen, descripción, reivindicaciones, dibujos)
- ☐ Copia del examen preliminar internacional efectuado por la Autoridad Internacional de Examen Preliminar (IPEA).
- ☐ Traducción del examen preliminar internacional efectuado por la IPEA.
- ☐ 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 las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☐ Arte final 12x12.
- ☐ De ser el caso, información sobre otras solicitudes de patente o títulos obtenidos en el extranjero por el mismo titular o su causante, relacionadas parcial o totalmente con la invención de esta solicitud.
- ☐ De ser el caso, modificaciones efectuadas a la solicitud internacional.
- ☐ Otros.

11 FIGURA CARACTERISTICA



12

Solicito la Concesión de la patente,

NOMBRE:

DARIO CARDENAS NAVAS

FIRMA:

C.C.

17.066.629BTA

TP.

1.250
C.S.J.

"SHELL" DEL SISTEMA DE ARCHIVOS

La presente invención se refiere a los archivos del sistema, y más específicamente, al shell del sistema de archivos. (N.T. el "*shell*" se refiere al espacio donde se manejan las instrucciones a ejecutar, a nivel del sistema operativo, y en este caso, al navegador que permite recorrer el sistema de archivos).

ANTECEDENTES DE LA INVENCION

Los sistemas de archivos actuales de los computadores presentan una serie de limitaciones no deseables. Una limitación consiste en que los usuarios generalmente no tienen control sobre la estructura que se les muestra. En otras palabras, cuando se organizan las carpetas, el usuario debe escoger una estructura, y luego dicha estructura es difícil de cambiar. Como ejemplo específico, para una carpeta de "música", un usuario puede escoger organizar los archivos musicales en un formato por artista o álbum, donde todas las carpetas de los álbumes de cada artista se agrupan en esa carpeta particular de ese artista, y todas las canciones de un álbum en particular se agrupan en una carpeta correspondiente a dicho álbum. El formato artista / álbum no conduce a escuchar un tipo de música (p. ej. reproducir dos canciones de jazz de dos artistas diferentes)), ni para escuchar una selección de álbumes de diferentes artistas.

Como otro asunto, un usuario puede tener un gran número de archivos, los cuales son difíciles de organizar. Algunos usuarios implementan un sentido rígido de colocación de sus archivos, creando así jerarquías rígidas para ello. La administración de dichos archivos se vuelve compleja en aumento y difícil, en la medida en que el número de documentos disponibles crece, haciendo también difícil la búsqueda y la recuperación. Este problema se agrava adicionalmente cuando se utilizan archivos adicionales desde otros lugares, tales como archivos compartidos, etcétera.

Los usuarios también deben manejar archivos que se encuentran en diferentes lugares, tales como aquellos en diferentes dispositivos, en otros PCs, o en línea. Por ejemplo, los usuarios pueden escoger escuchar su música en el computador (en la medida en que se pueda tener acceso a ésta desde un programa de música) o pueden estar en línea y escuchar música desde los sitios de Internet, sin embargo existe una división estricta entre estas dos fuentes. La música que proviene de diferentes lugares está organizada de una manera diferente, y no se guarda de la misma forma ni en el mismo lugar. Otro ejemplo es el los archivos almacenados en una red empresarial que pueden estar inherentemente separados de los archivos que un usuario tiene en su máquina actual.

Los usuarios también tienen que hacer un seguimiento no solamente de cuáles archivos de datos se almacenan, pero en dónde se almacenan. Por ejemplo, para archivos musicales, los usuarios se ven forzados a guardar

copias en varios sistemas y tratar de hacer el seguimiento de cuáles archivos están localizados dónde. Esto puede hacer difícil la localización de archivos, aún cuando éstos estén almacenados localmente.

En ocasiones también es difícil encontrar y regresar a los archivos de un usuario. Un usuario puede encontrar dificultad en recordar dónde y cómo se almacenan ciertos archivos. Dado un conjunto de carpetas, y aún un grupo de archivos similares, los usuarios encuentran dificultad en encontrar rápidamente el archivo que buscan. Para los archivos almacenados en un lugar difícil de encontrar, localizarlos es mucho más complejo. Además, una vez que los usuarios tienen suficientes archivos en una carpeta, se vuelve más difícil analizar esa carpeta con rapidez, especialmente si sus contenidos son similares.

También para los usuarios algunas veces es difícil encontrar o regresar a los archivos en una red. Compartir y publicar archivos frecuentemente es difícil, y puede ser aún más difícil recuperar dichos archivos de algún usuario que los tenga disponibles. Los usuarios típicamente tienen que memorizar y localizar geográficamente los diferentes sitios y los nombres que requieren para poder encontrar archivos en una red.

Los nombres de los sitios pueden variar, lo que puede causarle confusión a un usuario en cuanto a lo "correcto". Esto es particularmente cierto en una red que contiene diferentes convenciones de nomenclatura, limitaciones y así

sucesivamente. Por ejemplo, algunos sistemas operativos pueden requerir nombres cortos sin incluir espacios en blanco con el fin de que sean visibles.

Los programas también frecuentemente guardan los archivos en su propio directorio o en otros lugares con nombre, lo que puede dificultarle a los usuarios encontrar el camino para regresar a los archivos. Con frecuencia los programas tienen directorios preestablecidos por defecto (en forma predeterminada) al igual que lugares para guardar los documentos. Un usuario normalmente tiene que buscar en el disco duro, y hacer suposiciones sobre el lugar donde se encuentra almacenado un archivo.

Los objetos relacionados también frecuentemente se almacenan en diferentes partes del disco duro, etc. Este problema se vuelve más común con el desarrollo de los servicios de medios digitales que tienen contenidos múltiples (p. ej. imágenes, música, vídeo).

Otro tema referente a los sistemas de archivos está relacionado con la barra de direcciones. En la medida en que los usuarios navegan dentro de un sistema de archivos en un computador, un control de interface gráfica convencional, al a cual se hace referencia como barra de direcciones, le muestra a los usuarios en qué lugar están dentro de la jerarquía del sistema de archivos. La barra de direcciones convencional indica la localización actual en términos de la estructura jerárquica del sistema de archivos de carpetas, subcarpetas y archivos. La modificación de la localización del usuario desplegada en la barra de direcciones convencional se hace típicamente de una

de dos formas. La primera es editar manualmente la dirección en la barra de direcciones. Esta edición manual de la dirección le permite aun usuario relocalizar cualquier número de localizaciones dentro de la jerarquía del sistema de archivos, pero requiere que el usuario sepa cierta información específica en relación con la organización de sistema de archivos en un computador, i.e. la localización específica de un sistema de archivos. El segundo método involucra usar herramientas externas de navegación, las cuales, cuando se manipulan, actualizan la barra de direcciones para reflejar la nueva dirección o localización. Mientras se evita hacer la edición manual de la dirección en la barra de direcciones, la manipulación de las herramientas externas de navegación aún requiere que el usuario tenga cierta información específica referente a la organización del sistema de archivos para recorrer la estructura jerárquica. Sin embargo, las barras de direcciones convencionales no pueden hacer referencia a archivos ni a datos almacenados en localizaciones múltiples del sistema de archivos, tales como carpetas o unidades, debido a la relación uno a uno existente entre la dirección en la barra de direcciones y una localización específica dentro de la jerarquía del sistema de archivos.

El estado anterior de la técnica adolece de una barra de direcciones que le permita a los usuarios especificar las direcciones que despliegan los archivos almacenados entre localizaciones múltiples del sistema de archivos. El estado anterior de la técnica además adolece de una barra de direcciones que le permita a los usuarios modificar con facilidad la dirección de la barra de

direcciones sin la edición manual de dicha dirección, o requiere de conocimiento específico referente a la organización del sistema de archivos subyacente. También, hace falta en el estado de la técnica anterior, una barra de direcciones que le presente al usuario presente selecciones alternativas, a partir de las

5 cuales el usuario pueda escoger navegar hacia dichas selecciones de archivos. Tal barra de direcciones podría selectivamente presentarle al usuario una interface de barra de direcciones convencional, permitiéndole interactuar con la barra de direcciones de acuerdo con la experiencia anterior, de conformidad con las preferencias del usuario.

10 Otro tema referente al sistema de archivos se refiere a la identificación de los objetos almacenados en un computador. La necesidad de identificar rápidamente los objetos almacenados en un ambiente de computación tal como un computador personal (PC por sus siglas en inglés: "*Personal Computer*"

15 aumenta dramáticamente en la medida en que más individuos utilizan computadores en sus rutinas diarias, y en la medida en que el tipo de información almacenada varía entre Imágenes, música, documentos, etc. Los documentos y los demás medios están típicamente almacenados en computadores de una forma jerárquica y están organizados con archivos de información o medios almacenados dentro de las carpetas. Los navegadores

20 de sistemas de archivos le permiten a los usuarios navegar a través del sistema de archivos y localizar y abrir los archivos y las carpetas. Por ejemplo WINDOWS® EXPLORER™ de Microsoft Corporation es una utilidad del

sistema operacional que le permite a los usuarios navegar por el sistema de archivos.

Muchos usuarios encuentran dificultades en identificar correctamente un archivo con base en la información que esté disponible en el momento con los navegadores convencionales del sistema de archivos. Por supuesto que el contenido de un archivo se puede verificar al abrirlo con un programa de aplicación, pero este método de navegación de archivos es ineficiente al extremo. La habilidad para visualizar metadatos referentes a un sistema de archivos puede ser de gran utilidad para proporcionar asistencia para identificar un archivo particular sin tener que abrirlo. En los sistemas operacionales WINDOW mediante el acceso a S@ 9X , por ejemplo, un usuario puede visualizar metadatos mediante el acceso a la hoja de propiedades de un objeto en particular. Una hoja de propiedades le muestra al usuario la lista de atributos o parámetros de un objeto, en forma de páginas de propiedades, en forma tabulada con una selección al estilo de tarjetas de índice, cada una de las cuales representa controles estándares estilo diálogo para personalizar los parámetros. Sin embargo, el uso de la página de propiedades para localizar un objeto puede resultar demorado y problemático, y algunos usuarios encuentran dificultad en localizar los metadatos relevantes en una hoja de propiedades. De igual forma, el uso de consejos de información para localizar un objeto puede ser demorado y problemático, ya que un usuario debe mover el ratón sobre

cada uno de los archivos, con el fin de visualizar los metadatos desplegados en forma limitada en dichos consejos de información.

Los navegadores convencionales del sistema de archivos no le permiten a los usuarios agregar o editar metadatos referentes a archivos o a carpetas, lo que mejoraría significativamente la habilidad del usuario para ubicar luego un archivo. A la fecha, la habilidad de los usuarios para agregar y editar metadatos está limitada por programas de computador (*software*) para fines especiales. Por ejemplo, los reproductores de medios para archivos de música electrónica le presentan a los usuarios la posibilidad de editar los metadatos asociados con los álbumes de música y con los artistas. Otro ejemplo de dichos programas incluye los programas de aplicaciones para archivos de dibujos electrónicos. Sin embargo, la utilidad de los reproductores de medios y tales otros programas está limitada al tipo particular de archivo que admite el programa, en oposición al navegador de sistemas de archivos de propósito común que da soporte y admite múltiples tipos de archivos.

El sistema operacional WINDOWS® XP de Microsoft Corporation incluye un navegador de imágenes para uso en la carpeta llamada "Mis Imágenes" ("*My Pictures*"). La carpeta Mis Imágenes está dotada de características especiales que le permiten a los usuarios visualizar imágenes y fotos, no solamente como íconos de documentos. Las características del navegador de imágenes de Mis Imágenes incluyen la habilidad de visualizar versiones de tamaño pequeño y versiones de fotos de tamaño grande, rotar fotos que se

encuentran de lado, y crear presentaciones de transparencias. Un usuario también puede visualizar los detalles de la foto, tales como sus dimensiones, la fecha y hora en que fue tomada, y el nombre de la cámara que la capturó. El área de control de previsualizaciones en la carpeta de Mis Imágenes contiene

5 una imagen agrandada de previsualización de una imagen seleccionada por el usuario, botones para iterar, que le ayudan al usuario a iterar a través de una serie de imágenes y controles para rotar imágenes en dirección horaria o contra-horaria. Mientras las características del navegador de imágenes en WINDOWS XP® han hecho avanzar el estado de la técnica mediante el alivio

10 de la necesidad de invocar un programa de aplicación para ver y manipular imágenes, los usuarios aún no pueden agregar y editar metadatos asociados con las imágenes.

De conformidad con lo anterior, existe una necesidad de una experiencia mejorada del usuario dentro del shell o navegador del sistema de archivos que

15 le permita a los usuarios localizar rápidamente un objeto, con base en los metadatos asociados con dicho objeto. También existe la necesidad de un sistema y un método que le permitan al usuario agregar y editar metadatos asociados con los objetos de varios tipos dentro de un navegador del shell que le ofrezca a los usuarios características mejoradas para reconocimiento de

20 archivos de tal forma que los usuarios puedan localizar sus archivos con rapidez. También existe la necesidad de un navegador de shell ampliable que pudiera permitirle a los desarrolladores de programas de computador (*software*)

suministrarle información y funcionalidad adicional a los usuarios, sobre una base de tipo de archivo. También existe la necesidad de suministrar una experiencia de interface de usuario (UI por su sigla en inglés, user interface) similar a través de diferentes colecciones de objetos.

5

RESUMEN DE LA INVENCION

De conformidad con uno de los aspectos de la invención, se suministra un método y un sistema que utiliza carpetas virtuales. Las carpetas virtuales exponen carpetas y archivos comunes (también conocidos como directorios) ante los usuarios en diferentes vistas, basados en los metadatos en lugar de basarse en la estructura subyacente del sistema de archivos en el disco. Así, el sistema es capaz de tomar una propiedad almacenada en la base de datos y representarla como un contenedor que es como una carpeta. Debido a que los usuarios ya están familiarizados con el trabajo con carpetas, mediante la representación de carpetas virtuales en una forma similar, los usuarios se pueden adaptar al nuevo sistema más rápidamente.

De acuerdo con otro aspecto de la invención, las carpetas virtuales se suministran de acuerdo con un método que se utiliza en un sistemas de computador con una representación y una memoria para almacenar los objetos. De conformidad con este método, se selecciona una propiedad de los metadatos. El sistema entonces busca los objetos que tienen la propiedad seleccionada de los metadatos, y se suministra un objeto virtual de

representación de carpetas, que a su vez representa la colección de objetos que tienen la propiedad de metadatos.

De acuerdo con otro aspecto de la invención, el sistema incluye un procesador de carpetas que obtiene las consultas de un usuario, y una base de datos relacional, el procesador de carpetas le proporciona al usuario el resultado en forma de carpetas virtuales. En una de las modalidades, los resultados que se suministran de regreso al procesador de carpetas incluyen las columnas y filas de la base de datos. Las filas y columnas de la base de datos son convertidas por el procesador de carpetas en una estructura enumeradora, que luego se usa para poblar el despliegue (de información) con las carpetas virtuales resultantes.

De acuerdo con otro aspecto de la invención, los usuarios podrán trabajar con carpetas virtuales por medio de manipulación directa. En otras palabras, los mecanismos suministrados para la manipulación de carpetas virtuales son similares a aquellos que se usan actualmente para manipular carpetas físicas convencionales (p.ej. pinchar (seleccionar) y arrastrar, copiar, pegar, etc.)

De acuerdo con otro aspecto de la invención, el método para llevar a cabo la manipulación directa de las carpetas virtuales, se suministra en un sistema de computadores con una representación o despliegue y una memoria para almacenar los objetos. De conformidad con el método, los grupos de objetos se representan como carpetas virtuales. Las acciones definidas se

suministran en tal forma que pueden ser llevadas a cabo mediante la manipulación directa de las carpetas virtuales, en la cual cuando una acción definida se realiza, la carpeta virtual es manipulada de acuerdo con las instrucciones de la acción definida. Un ejemplo de una acción definida sería

5 seleccionar y arrastrar una carpeta virtual. En una de las modalidades, la acción de seleccionar y arrastrar una primera carpeta virtual a una segunda carpeta virtual cumple la función de copiar los objetos desde la primera carpeta virtual a la segunda carpeta virtual. La copia de los objetos a una carpeta virtual puede involucrar la adición o de otra forma la alteración de las propiedades

10 seleccionadas de metadatos asociadas a estos objetos.

De acuerdo con otro aspecto de la invención, se suministran filtros para la manipulación de las carpetas virtuales. Los filtros son esencialmente herramientas para disminuir el tamaño de un conjunto de objetos. En una de las modalidades, los filtros se generan dinámicamente, con base en las

15 propiedades de los objetos por separado. Por ejemplo, para un conjunto de objetos, el mecanismo de filtro puede revisar las propiedades, y si los objetos incluyen en general al "autor" dentro de las propiedades, entonces el filtro podrá suministrar una lista de autores. Luego, seleccionando un autor determinado, los objetos que no correspondan a ese autor desaparecerán. Esto le permite al

20 usuario disminuir el contenido.

De acuerdo con otro aspecto de la invención, se suministra un método para filtrar objetos en sistemas de computador que tienen una representación o

despliegue (de información) y una memoria para almacenar objetos con propiedades de metadatos. Los objetos del despliegue se suministran en el despliegue en el cual cada uno representa uno o más objetos. Las propiedades de los metadatos de los objetos que se representan en los objetos del despliegue se evalúan. Un término de filtro se suministra en el despliegue, que corresponde a una propiedad de metadatos que se comparte por una pluralidad de objetos, en los cuales la selección del término del filtro hace que los objetos que se representan en el despliegue se reduzcan a aquellos objetos que comparten la propiedad de metadatos específica.

10 De conformidad con otro aspecto de la invención, una pluralidad de objetos se representan en el despliegue y se genera dinámicamente un término de filtro con base en las propiedades de metadatos de los objetos. Cuando se selecciona un filtro, se reducen los objetos representados en el despliegue, a aquellos que tienen la propiedad de metadatos que corresponde a dicho término de filtro.

De acuerdo con otro aspecto de la invención, una pluralidad de objetos se representan en el despliegue y se suministra un área de filtro en la cual un usuario puede definir un término de filtro. Cuando se introduce un término de filtro por parte de un usuario, los objetos representados en el despliegue se reducen a aquellos que contienen el término del filtro. En la medida en que el usuario escribe el término del filtro, se pueden filtrar objetos adicionales, al irse adicionando cada carácter al término del filtro.

De acuerdo con otro aspecto de la invención, se suministra un botón de retroceso que puede usarse para retroceder durante el proceso de filtrado. Por ejemplo, después de que un usuario ha entrado el término de filtro, el usuario puede desear devolverse al conjunto de objetos que estaba representado en el despliegue antes de aplicar el filtro. El botón de retroceso le permite al usuario retroceder hasta el punto deseado durante la navegación usando el filtro.

De acuerdo con otro aspecto de la invención, se suministran enlaces rápidos. En una de las modalidades, los enlaces rápidos son un conjunto de enlaces determinados (p.ej. localizados en el lado izquierdo del despliegue) que pueden pincharse (seleccionarse) para generar vistas útiles del conjunto de objetos. Estos se pueden predefinir a través del programa, o pueden ser fijados por parte del usuario. Por ejemplo seleccionado sobre "todos los autores" podría retornar una vista apilada por autores. "Todos los documentos" podría retornar una vista plana de todos los documentos en todas las áreas de almacenamiento. Los usuarios también podrían crear sus propios enlaces rápidos. Por ejemplo, un usuario podría filtrar todos los documentos modificados en Enero de 2003, y luego podría guardar esta acción en un enlace rápido.

De acuerdo con otro aspecto de la invención, el método de suministro de enlaces rápidos se implementa en un sistema de computadores con un despliegue y con una memoria para almacenar objetos. De conformidad con el método, un usuario primero navega para visualizar una colección deseada de objetos. El enlace rápido que responde a la colección deseada de objetos se

guarda y se le da un nombre. El nombre del enlace rápido se presenta en el despliegue en forma tal que al seleccionar el enlace rápido, un usuario puede retomar a la vista de la colección deseada de objetos.

De acuerdo con otro aspecto de la invención, se suministran librerías.

- 5 Las librerías consisten en grandes grupos de tipos de archivos utilizables que se pueden asociar entre sí. Por ejemplo, las fotos pueden ser una librería, la música otra, y los documentos otra. Las librerías ofrecen herramientas y actividades que están relacionadas con los tipos particulares de objetos. Por ejemplo, en la librería de fotos existen herramientas y filtros que se relacionan
- 10 con la manipulación de las fotos, así como para la creación de presentaciones de transparencias o para compartir imágenes.

- De acuerdo con otro aspecto de la invención, se suministra un método para crear librerías en un sistema de computador con despliegue y con una memoria para almacenar objetos. El método inicia mediante la creación de una
- 15 librería que contenga objetos con una o más de las propiedades especificadas de los metadatos. Los objetos con una o más de las propiedades de los metadatos se agrupan automáticamente en la librería. También se suministran herramientas para manipular los objetos de la librería.

- De acuerdo con otro aspecto de la invención, una amplia gama de
- 20 archivos u objetos se encuentran disponibles. En otras palabras, el sistema es capaz de representar archivos / objetos que se encuentran en localizaciones físicas múltiples (p. ej. diferentes unidades de disco duro, diferentes

computadores, diferentes localizaciones en redes, etc.) de tal forma que los objetos son vistos por el usuario apareciendo en una misma localización. Por ejemplo, a un usuario se le pueden presentar todos los archivos musicales en una sola pantalla y todos los archivos se pueden manipular desde una vista, aún cuando los archivos puedan estar físicamente almacenados en diferentes unidades de disco duro, diferentes computadores o diferentes localizaciones en redes.

De acuerdo con otro aspecto de la invención, se utiliza un alcance en el método para desplegar objetos en un sistema de computador con un despliegue (de información). El método involucra la definición de un alcance de las localizaciones físicas de memoria a partir de las cuales se deben recuperar los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una de las demás localizaciones físicas que se definieron en el alcance, y los objetos que se recuperan con la consulta luego se representan en una vista sobre el despliegue. En una de las modalidades, la otra localización física puede ser otro computador, una localización en una red, o un dispositivo de almacenamiento externo. En una de las modalidades, la vista en el despliegue puede cambiarse por una lista de las carpetas físicas, la cual indica las localizaciones físicas en las cuales los objetos se almacenan físicamente.

De acuerdo con otro aspecto de la invención, objetos diferentes a archivos se pueden representar en las carpetas virtuales. En otras palabras, los archivos almacenados en memoria se localizan en un almacenamiento físico.

Las carpetas virtuales se pueden hacer de tal forma que incluyan objetos que actualmente no están representados en el almacenamiento físico. Ejemplos de estos objetos que no son archivos son los correos electrónicos y los contactos.

De acuerdo con otro aspecto de la invención, se implementa un método
5 para presentar objetos diferentes a archivos en un sistema de computador con un despliegue (de información) y una memoria para almacenar objetos. El método incluye el suministro de una base d e datos que permite seleccionar objetos de archivos y objetos diferentes a archivos mediante una búsqueda de una consulta. Una vez se recibe una consulta, tanto los objetos de archivo
10 como los objetos diferentes a los archivos que cumplen con el criterio de la consulta se recuperan, y los objetos que cumplen el criterio de la consulta se presentan en el despliegue. En una de las modalidades, se suministra una base de datos relacional que incluye información seleccionada referente a los objetos de archivo, y que pueden contener algunos objetos diferentes a archivos en su
15 totalidad.

De acuerdo con otro aspecto de la invención, se suministra una barra de direcciones virtuales para seleccionar contenidos almacenados en un sistema de archivos en un computador. Una barra de direcciones virtuales comprende una pluralidad de segmentos. Cada segmento corresponde a un filtro para
20 seleccionar el contenido almacenado en el sistema de archivos en un computador. Colectivamente, los filtros correspondientes de cada segmento en

la barra de direcciones virtuales representa una dirección virtual para seleccionar el contenido almacenado en un sistema de archivos de computador.

Cada segmento es un segmento interactivo que puede responder a las interacciones del usuario para modificar las direcciones virtuales de la barra de direcciones virtuales. La selección de un segmento en la barra de direcciones virtuales hace que aquellos segmentos subsiguientes al segmento seleccionado sean eliminados de la barra de direcciones virtuales. Alternativamente, la selección de un segmento en la barra de direcciones virtuales genera una lista de filtros pares seleccionables que son desplegados para el usuario. Los filtros pares son pares del filtro correspondiente de los segmentos seleccionados alternativamente. La selección de uno de los filtros pares hace que el segmento seleccionado alternativamente reemplace su filtro correspondiente con el filtro par seleccionado. Adicionalmente, aquellos segmentos subsiguientes a los segmentos de filtro seleccionados alternativamente sean eliminados de la barra de direcciones virtuales.

Los segmentos se pueden agregar a la barra de direcciones virtuales de acuerdo con las acciones externas del usuario. Los segmentos son agregados al final de los segmentos en la barra de direcciones virtuales. Cualesquiera segmentos que puedan estar en conflicto con un segmento agregado son removidos. Un segmento existente en la barra de direcciones virtuales también entra en conflicto con el segmento agregado cuando el segmento existente excluye mutuamente al segmento agregado. Un segmento existente en la barra

de direcciones virtuales también entra en conflicto con el segmento agregado cuando el segmento agregado es más amplio o menor en alcance que el segmento agregado.

De acuerdo con otro aspecto de la invención, se suministra un navegador de *shell* que incluye una ventana y un control de edición. La ventana despliega un grupo de objetos y también despliega valores de metadatos asociados con uno o más de los objetos desplegados. El control de edición permite la modificación por parte del usuario de por lo menos una porción de los valores de metadatos desplegados en la ventana.

De acuerdo con otro aspecto de la invención, una interface gráfica de usuario (GUI, por sus siglas en inglés: "*Graphical User Interface*") está incorporada a un medio legible por computador y ejecutable en el mismo. La interface gráfica de usuario incluye una primera área de pantalla que despliega en conjunto de objetos en un navegador de la *shell* y una segunda área de pantalla que despliega los metadatos asociados con uno o más de los objetos desplegados. La interface gráfica de usuario también le muestra al usuario los medios, dentro del navegador de *shell*, para modificar los metadatos desplegados.

De acuerdo con otro aspecto de la invención, se suministran métodos implementados por computador para permitirle al usuario modificar los metadatos dentro del navegador de la *shell*. Uno de tales métodos incluye el despliegue de una pluralidad de objetos, recibiendo una primera entrada del

usuario que represente la selección de por lo menos un objeto desplegado, desplegando los metadatos asociados con el(los) objeto(s) seleccionado(s) y suministrando un control para las modificaciones por parte del usuario, de los metadatos desplegados. Otro de esos métodos incluye el despliegue de un

5 aviso de bienvenida y los metadatos asociados con tal aviso de bienvenida, suministrando un control de edición para las modificaciones de los metadatos desplegados.

De acuerdo con otro aspecto de la invención, en el navegador de la *shell* se despliega una estructura de datos que contiene los metadatos asociados con

10 uno o más objetos. La estructura de datos, que se guarda en uno o más de los medios legibles por computador, incluye un campo que contiene los metadatos modificables por el usuario asociados con uno o más de los objetos desplegados, y los metadatos modificables por el usuario contenidos en la estructura de datos también se despliega en el navegador de la *shell*.

15 De acuerdo con otro aspecto de la invención, se suministra un navegador de la *shell* que incluye un visor previo predeterminado y un mecanismo de extensión. El visor previo predeterminado suministra un nivel estándar de funcionalidad para múltiples tipos de objetos. El mecanismo de extensión permite una funcionalidad superior a la del nivel estándar suministrado por el

20 visor previo predeterminado para uno o más de los tipos de objetos.

De acuerdo con otro aspecto de la invención, se suministra un navegador de la *shell* que incluye un primer y un segundo visores previos. El primer visor

previo suministra un nivel estándar de funcionalidad para múltiples tipos de objetos, y el segundo visor previo suministra un nivel alternativo extendido de funcionalidad para uno o más de los múltiples tipos de objetos. El navegador de la *shell* está configurado para desplegar ya sea el primero o el segundo visor previo para uno o más de los tipos de objetos.

De acuerdo con otro aspecto de la invención, se suministra una interface gráfica de usuario para el navegador de la *shell* que soporta múltiples tipos de objetos. La interface gráfica de usuario incluye una primera área de pantalla para desplegar un conjunto de objetos en el navegador de la *shell* y los medios para seleccionar un visor previo para los objetos desplegados, a partir de una pluralidad de visores previos disponibles.

De acuerdo con otro aspecto de la invención, se suministra un método implementado por computador para permitir el uso de visores previos de terceros en un navegador de *shell* que soporta múltiples tipos de objetos. El método incluye el suministro de un navegador de *shell* que tiene un visor previo para los múltiples tipos de objetos y que suministra un mecanismo de extensión que le permite a terceros desarrollar un visor previo alternativo para por lo menos uno de los múltiples tipos de objetos.

De acuerdo con otro aspecto de la invención, se suministra una estructura de datos que contiene información que indica la pluralidad de visores previos en un navegador de la *shell*. La estructura de datos, que se guarda en uno o más medios legibles por computador, incluye un primer campo que

contiene información que indica un visor previo predeterminado que soporta múltiples tipos de objetos. Un segundo campo que contiene información que indica un visor previo alternativo para un primer tipo de objeto, y un tercer campo que contiene información que indica si se debe invocar el visor previo
5 predeterminado o un visor previo alternativo cuando los objetos del primer tipo de objetos se despliegan en el navegador de la *shell*.

De acuerdo con otro aspecto de la invención, diferentes tipos de objetos se agrupan en librerías para las cuales se suministra un conjunto similar de características básicas de interface del usuario (UI, por sus siglas en inglés:
10 "User Interface"). En otras palabras, se suministra un conjunto similar de características básicas de interface del usuario para diferentes tipos de librerías, tales como librerías de documentos, librerías de fotografías, y librerías de música. Este conjunto de características de la UI pueden incluir características tales como filtros, creación de nuevas categorías, edición de los metadatos de
15 los objetos, modificación de los pivotes, etc. EL conjunto similar de características de la UI para las librerías le permite a un usuario procesar y organizar diferentes tipos de objetos usando atributos y características con las cuales estén familiarizados.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

20 Los aspectos anteriores y muchas de las ventajas concurrentes de esta invención serán apreciadas más fácilmente en la medida en que ellas sean

entendidas mejor mediante la referencia a la siguiente descripción detallada, al verse en conjunto con los dibujos adjuntos en donde:

La FIGURA 1 es un diagrama de bloques de un sistema de computador de propósito general adecuado para implementar la presente invención;

5 La FIGURA 2 es un diagrama un diagrama de bloques de un sistema de carpetas virtuales de acuerdo con la presente invención;

La FIGURA 3 es un diagrama de flujo que ilustra una rutina por medio de la cual un usuario suministra una consulta que recupera carpetas y archivos seleccionados;

10 La FIGURA 4 es un diagrama de flujo que ilustra una rutina por medio de la cual las carpetas virtuales se construyen y se despliegan en la pantalla, de acuerdo ya sea con una consulta predeterminada o una consulta realizada por el usuario;

La FIGURA 5 es un diagrama de árbol de una estructura de carpetas de acuerdo con un arreglo físico de carpetas en una unidad de disco duro;

15

La FIGURA 6 es un diagrama de árbol de una estructura de carpetas virtuales;

La FIGURA 7 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 6, donde el conjunto de clientes además se filtra por contratos y por año;

20

La FIGURA 8 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 7, donde el conjunto de clientes además se filtra por año;

La FIGURA 9 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 6 donde el grupo de contratos además se filtra por clientes y por año, de la cual los clientes aún están filtrados por año;

La FIGURA 10 es un diagrama ilustrativo de un despliegue de pantalla que muestra los grupos de una librería de documentos;

La FIGURA 11 es un diagrama ilustrativo de un despliegue de pantalla que muestra los documentos en el grupo de ABC Corp., de la FIGURA 10;

La FIGURA 12 es un diagrama ilustrativo de un despliegue de pantalla en el cual se selecciona una función de agrupamiento para los documentos de la FIGURA 11;

La FIGURA 13 es un diagrama ilustrativo de un despliegue de pantalla en el cual se selecciona un parámetro de agrupamiento para "agrupar por autor", para la función de agrupamiento de la FIGURA 12;

La FIGURA 14 es un diagrama ilustrativo de un despliegue de pantalla en el cual los archivos de la FIGURA 13 se ha agrupado por autor;

La FIGURA 15 es un diagrama ilustrativo de un despliegue de pantalla en el cual la función de agrupamiento se selecciona, y además se selecciona una opción de "agrupar por categoría" para reagrupar los archivos de la FIGURA 14;

La FIGURA 16 es un diagrama ilustrativo de un despliegue de pantalla en el cual los archivos de la FIGURA 14 se han reagrupado por categoría;

La FIGURA 17 es un diagrama ilustrativo de un despliegue de pantalla en el cual se selecciona un enlace rápido para mostrar carpetas físicas;

5 La FIGURA 18 es un diagrama ilustrativo de un despliegue de pantalla en el cual se muestran las carpetas físicas que contienen los archivos de los agrupamientos de las carpetas virtuales de la FIGURA 17;

La FIGURA 19 es un diagrama de flujo que ilustra una rutina por medio de la cual un usuario puede directamente manipular las carpetas virtuales;

10 La FIGURA 20 es un diagrama ilustrativo de un despliegue de pantalla en el cual se ha agregado un nuevo grupo "Costa Oeste" al los grupos de la FIGURA 10;

La FIGURA 21 es un diagrama ilustrativo de un despliegue de pantalla en el cual se usa la manipulación directa para copiar los archivos del grupo "ABC Corp." al grupo "Costa Oeste" de la FIGURA 20;

La FIGURA 22 es un diagrama de flujo que ilustra una rutina para el sistema que genera dinámicamente nuevos términos de filtro;

La FIGURA 23 es un diagrama de flujo que ilustra una rutina para el sistema de filtro de objetos con base en la selección de un término de filtro;

20 La FIGURA 24 es un diagrama ilustrativo de un despliegue de pantalla en el cual los grupos de la FIGURA 10 se han filtrado con el término "AB";

La FIGURA 25 es un diagrama ilustrativo de un despliegue de pantalla en el cual los grupos de la FIGURA 10 se han filtrado con el término "ABC";

La FIGURA 26 es un diagrama ilustrativo de un despliegue de pantalla en el cual se selecciona el término de filtro "año 2002" para los grupos de la
5 FIGURA 10;

La FIGURA 27 es un diagrama ilustrativo de un despliegue de pantalla en el cual los grupos de la FIGURA 10 se han filtrado por "año 2002" y por una selección adicional del término de filtro "mes";

La FIGURA 28 es un diagrama ilustrativo de un despliegue de pantalla en
10 el cual se presenta una lista para seleccionar un mes para el filtro;

La FIGURA 29 es un diagrama ilustrativo de un despliegue de pantalla en el cual los grupos de la FIGURA 10 han sido además filtrados con el mes de Enero y además muestra un filtro por "día";

La FIGURA 30 es un diagrama de flujo que ilustra una rutina para crear
15 un enlace rápido nuevo;

La FIGURA 31 es un diagrama de flujo que ilustra una rutina para crear un enlace rápido nuevo llamado "Trabajo de Enero"

La FIGURA 32 es un diagrama ilustrativo de un despliegue de pantalla en el cual se selecciona un enlace rápido de "Todos los Autores";

20 La FIGURA 33 es un diagrama ilustrativo de un despliegue de pantalla en el cual se presenta la lista de los autores de la FIGURA 32;

La FIGURA 34 es un diagrama ilustrativo de un despliegue de pantalla en el cual el "Autor 1" se ha seleccionado de la lista de la FIGURA 33 y donde se muestran todos los documentos del "Autor 1";

La FIGURA 35 es un diagrama de flujo que ilustra una rutina para crear
5 una nueva librería;

La FIGURA 36 un diagrama ilustrativo de un despliegue de pantalla en el cual se presenta la colección de librerías disponibles;

La FIGURA 37 es un diagrama de flujo que ilustra una rutina para definir el alcance de una colección de carpetas virtuales;

10 La FIGURA 38 es un diagrama de bloques que ilustra varias fuentes a partir de las cuales se puede formar el alcance de una colección de carpetas virtuales;

La FIGURA 39 es un diagrama de flujo que ilustra una rutina para incluir objetos diferentes a archivos en una colección de carpetas virtuales;

15 La FIGURA 40 es un diagrama ilustrativo de un despliegue de pantalla en el cual se muestran varios objetos diferentes a archivos incluidos en una carpeta virtual;

La FIGURA 41 es un diagrama pictórico de un ambiente de red de computadores ejemplar, adecuado para implementar la presente invención;

20 La FIGURA 42 es un diagrama pictórico que ilustra un visor de archivos ejemplar con una barra de direcciones convencionales asociada al despliegue

de archivos en un sistema de archivos de computador, tal como se encuentra en el estadio anterior de la técnica;

La FIGURA 43 es un diagrama pictórico que ilustra un visor de archivos ejemplar para desplegar archivos en un sistema de archivos de computador de acuerdo con una dirección virtual en una barra de direcciones virtuales de conformidad con la presente invención;

La FIGURA 44A es diagrama pictórico del visor de archivos ejemplar de la FIGURA 5 que ilustra la selección de un segmento de la dirección virtual en la barra de direcciones virtuales para navegar en el sistema de archivos;

La FIGURA 44B es diagrama pictórico del visor de archivos ejemplar de la FIGURA 45A que ilustra los resultados de seleccionar un segmento de la dirección virtual en la barra de direcciones virtuales;

Las FIGURAS 45A-45D son diagramas pictóricos que ilustran la selección de un filtro par asociado a un segmento de una dirección virtual en la barra de direcciones virtuales;

Las FIGURAS 46A-46D son diagramas pictóricos que ilustran filtros adicionales a una dirección virtual en la barra de direcciones virtuales;

Las FIGURAS 47A-47B son diagramas pictóricos que ilustran una barra de direcciones virtuales ejemplar desplegando una dirección virtual donde la dirección virtual excede la capacidad de despliegue de la barra de direcciones virtuales;

La FIGURA 48A es un diagrama pictórico que ilustra una barra de direcciones virtual ejemplar con una dirección virtual con filtros que referencian localizaciones tanto virtuales como actuales en un sistema de archivos;

La FIGURA 48B es un diagrama pictórico que ilustra la barra de direcciones virtual ejemplar de la FIGURA 48A configurada para desplegar una barra de direcciones convencional;

La FIGURA 49 es un diagrama pictórico que ilustra la barra de una selección alternativa de rutina de filtro para seleccionar filtros alternativos para una dirección virtual en una barra de direcciones virtuales;

La FIGURA 50 es un diagrama de flujo que ilustra una rutina ejemplar para agregar filtros a una dirección virtual en una barra de direcciones virtuales;

La FIGURA 51A es un diagrama de una interface gráfica de usuario para un navegador de la *shell* que tiene un control de edición de acuerdo con una modalidad de la presente invención;

La FIGURA 51B es un diagrama de bloques de una interface gráfica de usuario para un navegador de la *shell* que tiene uno o más controles de edición de acuerdo con una modalidad de la presente invención:

La FIGURA 52 es un diagrama esquemático de un aviso de bienvenida en un navegador de la *shell*;

La FIGURA 53 es un diagrama esquemático de un aviso seleccionado en un navegador de la *shell*;

La FIGURA 54 es un diagrama esquemático de un aviso seleccionado de la FIGURA 53 que incluye un menú de contexto que le permite al usuario modificar los metadatos en un navegador de la *shell* de acuerdo con una modalidad de la presente invención;

- 5 La FIGURA 55 es un diagrama de flujo que ilustra un método para permitirle al usuario modificar los metadatos desplegados en un aviso de bienvenida dentro de un navegador de la *shell* de acuerdo con una modalidad de la presente invención;

- 10 La FIGURA 56 es un diagrama de flujo que ilustra un método para permitirle al usuario modificar los metadatos desplegados en un aviso seleccionado dentro de un navegador de la *shell* de acuerdo con una modalidad de la presente invención;

- 15 La FIGURA 57 es un diagrama de bloques de una estructura de datos que contiene metadatos modificables por el usuario asociados a un objeto desplegado en el navegador de la *shell* ;

La FIGURA 58 es un diagrama esquemático de una interface gráfica de usuario anterior para navegar a través de imágenes guardadas en una carpeta dentro de un ambiente de navegador de la *shell* que se usa para visualizar otros archivos y carpetas no gráficos;

- 20 La FIGURA 59 es un diagrama de bloques de una interface gráfica de usuario para un navegador de la *shell*;

La FIGURA 60 es un diagrama esquemático de un aviso de bienvenida en un navegador de la *shell*;

La FIGURA 61 es un diagrama esquemático de un aviso seleccionado en un navegador de la *shell*;

5 La FIGURA 62 es un diagrama esquemático de un aviso seleccionado en un navegador de la *shell*, con controles extendidos de acuerdo con una modalidad de la presente invención;

La FIGURA 63 es un diagrama esquemático de un aviso seleccionado similar a la FIGURA 61 pero incluyendo un menú de contexto que le permite al
10 usuario seleccionar un visor previo en un navegador de la *shell*, de acuerdo con una modalidad de la presente invención;

La FIGURA 64A es un diagrama de flujo que ilustra un método que le permite al usuario seleccionar un visor previo en un navegador de la *shell*, de acuerdo con una modalidad de la presente invención;

15 La FIGURA 64B es un diagrama de flujo que ilustra un método que le permite al sistema seleccionar un visor previo en un navegador de la *shell*, de acuerdo con una modalidad de la presente invención;

La FIGURA 65 es un diagrama de flujo que ilustra un método que permite el uso de visores previos de terceros en un navegador de la *shell*, de acuerdo
20 con una modalidad de la presente invención;

La FIGURA 66 es un diagrama de bloques de una estructura de datos que contiene información que indica múltiples visores en un navegador de la *shell*;

DESCRIPCIÓN DETALLADA DE LA MODALIDAD PREFERIDA

5 La presente invención está dirigida a un *shell* de sistema de archivos que incluye un número de características deseables. En esencia, la *shell* le suministra a los usuarios la habilidad para visualizar y manipular archivos y otros objetos que están almacenados en un computador. La siguiente descripción primero suministra un resumen de las características que se
10 muestran en las FIGURAS 1-66, y luego presenta una descripción detallada.

 , En resumen, las FIGURAS 1-9 están dirigidas en general a un sistema general para carpetas virtuales. Las carpetas virtuales suministran un método para permitir que una interface gráfica de usuario convencional muestre carpetas y archivos comunes (también conocidos como directorios) a los
15 usuarios en diferentes vistas basadas en sus metadatos en lugar de basarse en la estructura del sistema de archivos física subyacente en el disco. Las FIGURAS 10-18 están dirigidas en general a los grupos, que están relacionados con la habilidad de las carpetas virtuales para tomar cualquier propiedad que está almacenada en la base de datos y representarla como un contenedor que
20 es como una carpeta. Las FIGURAS 19-21 están dirigidas en general a la manipulación directa de las carpetas virtuales, lo que se refiere al suministro de mecanismos par a manipular carpetas virtuales que son similares a los

mecanismos usados actualmente para manipular carpetas estándar (p. ej. copiar, pegar, seleccionar y arrastrar, etc.). las FIGURAS 22-29 están dirigidas en general a los filtros, que suministran un conjunto de herramientas para reducir un conjunto de archivos / objetos. Las FIGURAS 30-34 generalmente

5 están dirigidas a los enlaces rápidos, que son un conjunto de enlaces predefinidos que pueden seleccionarse para generar vistas útiles de conjuntos de archivos / objetos. Las FIGURAS 35-36 están dirigidas en general a las librerías, que están relacionadas con el concepto de que grupos de tipos usables de archivos se pueden asociar entre sí, y que se pueden suministrar

10 herramientas y actividades relacionadas con los tipos particulares de objetos a ser suministrados. Las FIGURAS 37-38 están dirigidas en general al alcance, que está relacionado con el concepto de tener la habilidad de adquirir archivos / objetos desde múltiples localizaciones físicas (p. ej. diferentes unidades de disco duro, diferentes computadores, diferentes localizaciones en una red de

15 computadores, etc.) de tal forma que ante el usuario, todos los archivos / objetos están presentados con la misma conveniencia como si fuesen suministrados desde una localización. Las FIGURAS 39-40 están dirigidas generalmente a objetos diferentes a archivos, que pueden ser incluidos en la base de datos junto con los archivos, y que pueden incluir objetos tales como

20 correos electrónicos y contactos. Las FIGURAS 41-50 están generalmente dirigidas a una barra de direcciones virtuales que comprende una pluralidad de segmentos , cada uno de los cuales corresponde a un filtro para seleccionar contenidos. Las FIGURAS 51-57 están dirigidas en general a un navegador de

la *shell* , con el cual los usuarios pueden rápidamente identificar un objeto con base en los metadatos asociados a dicho objeto. Las FIGURAS 58-66 están dirigidas generalmente a la ampliación de la funcionalidad de un visor previo de objetos en un navegador de la *shell* configurado para desplegar una pluralidad de objetos que representan múltiples tipos de objetos. La siguiente descripción suministra una descripción detallada de cada uno de estos aspectos de la invención.

Como se anotó antes, las FIGURAS 1-9 están dirigidas en general a un sistema para implementar carpetas virtuales. Las carpetas virtuales utilizan las mismas interfaces o similares a las que se usan actualmente para sistemas de archivos. Las carpetas virtuales exponen archivos y carpetas comunes (conocidos también como directorios) ante los usuarios en diferentes vistas basados en sus metadatos en lugar de basarse en la estructura subyacente de archivos del sistema en el disco. Vistas independientes de la localización se crean para permitirle a los usuarios manipular sus archivos y carpetas utilizando controles similares a aquellos que actualmente se usan para administrar sistemas de archivos. En general esto significa que los usuarios pueden organizar y reacomodar sus archivos con base en las propiedades inherentes de los archivos en sí, en lugar de llevar a cabo la organización y la administración como si esto fuera una parte separada del sistema. Las carpetas virtuales pueden representar archivos u objetos desde diferentes localizaciones físicas, como desde múltiples unidades de disco dentro del

mismo computador, entre múltiples computadores, o entre diferentes localizaciones en la red, de tal forma que una vista de archivos u objetos puede exponer los archivos u objetos residentes en diferentes localizaciones físicas. En una de las modalidades, los diferentes objetos o archivos requieren únicamente estar conectados a través de una red IP para poder ser incluidos.

El modelo de las carpetas virtuales también se puede usar para entidades tradicionalmente diferentes a archivos. Una aplicación de esto es tener un conjunto de interfaces de usuario similares a los archivos y las carpetas (esto es objetos y contenedores) para mostrar las entidades tradicionalmente diferentes a los archivos. Uno de los ejemplos de tales entidades diferentes a los archivos podría ser el correo electrónico, mientras que otro ejemplo es la información de los contactos proveniente de una base de datos de los contactos. De esta manera, las carpetas virtuales proporcionan un sistema de visualización o vistas independiente de la localización, basado en los metadatos, que funciona independientemente de si los datos mostrados provienen de archivos o de entidades diferentes a archivos. En general, estos aspectos permiten una mayor flexibilidad en términos de permitirle a los usuarios manipular sus archivos y datos usando tanto las técnicas comunes de interface de usuario (arrastrar, soltar y hacer doble "clic" o doble pinchada, etc.), así como el apalancamiento de la integración rica de varios tipos de datos.

La FIGURA 1 y la siguiente discusión pretenden proporcionar una descripción general y breve de un ambiente de computación adecuado en el

cual las carpetas virtuales y cualquiera de los demás aspectos de la presente invención que aquí se describen se pueden implementar. Aunque no se requiere, la invención se describirá en el contexto general de instrucciones ejecutables de computador, tales como módulos de programas que son
5 ejecutados por un computador personal. En general, los módulos de programas incluyen rutinas, programas, caracteres, componentes, estructuras de datos, etc., que realizan tareas particulares o que implementan tipos de datos abstractos en particular. Como lo pueden apreciar aquellos versados en la materia, la invención podrá ser puesta en práctica con otras configuraciones
10 de sistemas de computadores, incluyendo dispositivos portátiles, sistemas de multiprocesadores, electrónica programable de consumo basada en microprocesadores, redes de computadores personales (PCs), minicomputadores, "mainframes" y similares. La invención también puede ponerse en práctica en ambientes de computación distribuida donde las tareas
15 se llevan a cabo en dispositivos de procesamiento remoto que se encuentran enlazados a través de una red de comunicaciones. En un ambiente de computación distribuida, los módulos de los programas se pueden localizar tanto en dispositivos de almacenamiento de memoria tanto locales como remotos.

20 En relación con la FIGURA 1, un sistema ejemplar para implementar la invención, incluye un dispositivo de computación de propósitos generales en forma de un computador personal 20, incluyendo una unidad de procesamiento

21, memoria del sistema 22, y un "bus" del sistema 23 que acopla varios componentes del sistema incluyendo la memoria del sistema 22 a la unidad de procesamiento 21. El bus del sistema 23 puede ser de varios tipos de estructura de buses incluyendo un bus de memoria o controlador de memoria, un bus periférico, y un bus local utilizando cualquier variedad de arquitecturas de bus. La memoria del sistema incluye memoria solamente de lectura (ROM por sus siglas en inglés: "*Read Only Memory*") 24 y memoria de acceso aleatorio (lectura (RAM por sus siglas en inglés: "*Random Access Memory*") 25. Un sistema básico de entrada / salida (BIOS) 26, que contiene las rutinas básicas que ayudan a la transferencia de información entre elementos dentro del computador personal 20, tal como durante el arranque, almacenadas en el ROM 24. El computador personal 20 además incluye una unidad de disco duro 27 para escribir sobre o leer desde un disco duro 39, una unidad de disco magnético 28 para escribir sobre o leer desde un disco magnético removible 29, y una unidad de disco óptico 30 para escribir sobre o leer desde un disco óptico removible 31, tal como un Disco Compacto ROM (CD-ROM) u otro medio óptico. La unidad de disco duro 27, la unidad de disco magnético 28, y la unidad de disco óptico 30 están conectados al bus del sistema 23 mediante una interface de unidad de disco duro 32, una interface de unidad de disco magnético 33, una interface de unidad de disco óptico 34 respectivamente. Las unidades y sus respectivos medios asociados legibles por computador proporcionan almacenamiento no volátil para las instrucciones legible por computador, para estructuras de datos, módulos de programas, y otros datos

de los computadores personales 20. Aunque el ambiente ejemplar aquí descrito emplea un disco duro 39, un disco magnético removible 29, y un disco óptico removible 31, los conocedores de la materia deberían apreciar el hecho de que otro tipo de medios legibles por computador que pueden almacenar
5 datos a los cuales se puede tener acceso mediante un computador, tales como cintas magnéticas, tarjetas de memoria "flash", discos de video digital, cartuchos Bernoulli, memorias de acceso aleatorio (RAMs), memorias únicamente de lectura (ROMs), y similares, también podrán se usados en el ambiente operativo ejemplar.

10 Un número de módulos de programa se puede almacenar en el disco duro 39, en el disco magnético 29, en el disco óptico 31, en la ROM 24 o en la RAM 25, incluyendo el sistema operacional 35, uno o más programas de aplicación 36, otros módulos de programa 37 y datos de programa 38. Un usuario puede introducir comandos e información al computador personal 20 a
15 través de dispositivos de entrada tales como los teclados 40 y dispositivo señalador 42. Otros dispositivos de entrada (que no se muestran) pueden incluir micrófono, "joystick" tableta de juegos, plato satelital, escáner, o similares. Estos y otros dispositivos de entrada frecuentemente se conectan a la unidad de procesamiento 21 a través de una interface de puerto serial 46,
20 que está acoplado al bus del sistema 23, pero que también puede estar conectado a otras interfaces, tales como un puerto paralelo, un puerto de juegos o un puerto serial universal (USB, por sus siglas en inglés: "*Universal*

Serial Bus"). Un despliegue en forma de pantalla 47 también se acopla al bus del sistema 23 a través de una interface, tal como una tarjeta de video o un adaptador 48. Uno más parlantes 57 también se pueden conectar al bus del sistema 23 a través de una interface, tal como un adaptador de audio 56.

5 Adicionalmente al despliegue y a los parlantes, los computadores personales típicamente incluyen otros dispositivos periféricos de salida (que no se muestran) tales como impresoras.

El computador personal 20 puede operar en un ambiente de red utilizando conexiones lógicas a uno o más computadores personales, tales

10 como un computador remoto 49. El computador remoto 49 puede ser otro computador personal, un servidor, un enrutador, un PC de red, un dispositivo par u otro nodo común de red, y típicamente incluye muchos o todos los elementos descritos arriba referentes al computador personal 20. Las conexiones lógicas ilustradas en la FIGURA 1 incluyen una red de área local

15 (LAN por sus siglas en inglés: "*Local Area Network*") 51 y una red de área amplia (WAN por sus siglas en inglés: "*Wide Area Network*") 52, tal como la Internet. El módem 54, que puede ser externo o interno, se conecta al bus del sistema 23 a través de la interface del puerto serial 46. En un ambiente en red, los módulos de los programas ilustrados referentes al computador personal 20 o

20 a porciones del mismo se pueden almacenar en el dispositivo remoto de almacenamiento. Se apreciará que las conexiones de red mostradas son

ejemplares, y se pueden usar otros medios para establecer enlaces de comunicaciones entre computadores.

Así como se implementó en un sistema del tipo ilustrado en la FIGURA 1, la presente invención utiliza carpetas virtuales que le facilitan a los usuarios llevar a cabo tareas básicas alrededor de la manipulación de archivos y de la navegación entre carpetas ("*browsing*") y para proporcionar un mayor nivel de capacidades de almacenamiento que se puede apalancar con nuevas características. Las carpetas virtuales le exponen los archivos y los objetos a los usuarios en diferentes vistas con base en sus metadatos en lugar de basarse en la estructura física subyacente de archivos del sistema en el disco.

La FIGURA 2 es un diagrama de bloques de una sistema de carpetas virtuales 200 de acuerdo con la presente invención. Como se describe en mayor detalle a continuación, las carpetas virtuales le permiten a un usuario cambiar el "pivote" que controla la forma en que se visualizan los datos. Como ejemplo,, un usuario podría visualizar su música como una lista plana de todas las canciones, que se podrían agrupar por álbumes. Alternativamente, el usuario podría cambiar la vista para visualizar solamente los géneros de los artistas o sus años, etc. El usuario puede personalizar la vista para ver solamente los objetos que se ajustan a la tarea que tiene a la mano. Esto permite tener una experiencia mejorada de navegación que niega la necesidad de navegación adicional a través de carpetas (tanto hacia arriba como hacia abajo). Las misma lecciones y capacidades aplican al modelo de otros tipos de

datos no almacenados como archivos. Los contactos, por ejemplo, se pueden exponer de esta forma para el usuario, proporcionando capacidades de interface familiares así como una infraestructura más tica para manipularlos, comparado a lo que proporciona una libreta de direcciones plana.

5 Como lo ilustra la FIGURA 2, el sistema de carpetas virtuales 200 incluye un procesador de carpetas 210, una base de datos relacional 230, una base de datos de descripciones de carpetas virtuales 232, un componente de otras carpetas de la *shell* 234, un componente manejador de carpetas 236, un navegador de la *shell* y componente de visualización 240. El procesador de
10 carpetas 210 incluye un componente de manejador de código nativo 212, un componente de fábrica de manejadores 214, un componente de escritor de propiedades 216, un componente analizador de conjuntos de filas 218, un componente constructor de consultas 220, un componente enumerador 222, y un componente de fabricación de propiedades 224.

15 La base de datos relacional 230 almacena las propiedades acerca de todos los archivos del sistema. También almacena algunos objetos, como los contactos (i.e, objetos diferentes a archivos), en su totalidad. En general, almacena metadatos acerca de los tipos de archivos y objetos que contiene. La base de datos relacional 230 recibe consultas SQL desde el constructor de
20 consultas 220. La base de datos relacional 230 también envía conjuntos de filas SQL al componente analizador de conjuntos de filas 218, con una fila por columna de objeto, siendo las columnas las propiedades d los objetos.

La base de datos de descripciones de las carpetas virtuales 232 incluye las descripciones de las carpetas virtuales. La base de datos de descripciones de las carpetas virtuales 232 envía datos al componente constructor de consultas 220, incluyendo una lista de tipos para desplegar en la carpeta, el
5 filtro inicial, y las localizaciones físicas para mostrar los resultados de (los alcances).

En relación con el componente de otras carpetas del *shell* 234, el procesador de carpetas 210 le delega a las carpetas existentes de la *shell* de muchas clases de objetos, incluyendo todos los archivos, por manejadores o
10 propiedades. El componente de otras carpetas de la *shell* 234 le envía propiedades de otras carpetas al componente de fabricación de propiedades 224. El componente de otras carpetas de la *shell* también envía manejadores a la fábrica de manejadores 214.

Para el componente de manejador de código nativo 212, el procesador
15 de carpetas 210 implementa directamente ciertos manejadores con base en las propiedades de los objetos. El componente de manejador de código nativo 212 también envía manejadores a la fábrica de manejadores 214. Para el componente de manejador de código nativo 212 y el componente manejador de carpetas 236, así como todos los espacios de nombres, las carpetas
20 virtuales deben proporcionar un conjunto de manejadores (menú de contexto, íconos, consejos de información, señaladores,...) para sus objetos. Para la mayoría de éstos, (consejos de información, objetos de datos, manejadores

para arrastre y liberación, menú de contexto de antecedentes...) la carpeta virtual proporciona un manejador común (nativo) para todos los tipos que contenga. Sin embargo existen otros que deben ser suministrados por el autor del tipo (menú de contextos sobre el objeto en sí, almacén de propiedades que se pueden escribir,...). El Manejador predeterminado también puede sobrepasarse. Las carpetas virtuales vuelven a usar esto para los archivos y le permiten a los objetos diferentes a los archivos hacer lo mismo.

La fábrica de manejadores 214 toma listas ID y produce comportamientos de códigos que proporcionan menús de contexto, íconos, etc.

10 En general, el procesador de carpetas 210 puede usar manejadores nativos, manejadores externos, o delegar la consecución de manejadores en el componente de otras carpetas de la *shell*, como se describe arriba con respecto del componente de manejo de código nativo 212, el componente de otras carpetas de la *shell* 234, y el componente de manejadores de carpetas 236. El

15 componente de fábrica de manejadores 214 envía manejadores al navegador de la *shell* en la vista 240, en la forma en que lo requiere la vista. El componente de fábrica de manejadores 214 envía un manejador de propiedades al escritor de propiedades 216.

El escritor de propiedades 216 convierte las intenciones del usuario tales

20 como cortar, copiar, y pegar en derechos de propiedades del archivo o del objeto. Un navegador de la *shell* y un componente de visualización 240 le envía datos al escritor de propiedades 216, incluyendo la manipulación directa

(cortar/copiar/pegar) o la edición de los metadatos. En general, debido a que las carpetas virtuales presentan una organización basada en las propiedades de in objeto, operaciones tales como mover y copiar (arrastrar y dejar caer) se convierten en una edición de dichas propiedades. Por ejemplo, mover un
5 documento, en una vista agrupada por Autor, desde el Autor 1 al Autor 2, significa un cambio de autor. El componente escritor de propiedades 216 implementa esta función.

El analizador de conjuntos de filas 218 toma los conjuntos de filas y almacena todas las propiedades de los objetos en una estructura de lista ID
10 *shell*. Un conjunto de filas toma definiciones parciales de una carpeta virtual y construye una cadena SQL que entonces puede ser enviada a la base de datos. El componente analizador de filas 218 envía listas ID al componente enumerador 222. Como se describe arriba, el componente analizador de filas 218 también recibe datos de la base de datos relacional 230, incluyendo
15 conjuntos de filas SQL, con una fila por objeto siendo las columnas las propiedades de los objetos.

El componente constructor de consultas 220 recibe datos de l componente enumerador 222, incluyendo nuevos filtros de la navegación. El componente constructor de consultas 220 también recibe datos de la base de
20 datos de descripciones de las carpetas virtuales 232, incluyendo una lista de tipos a desplegar en la carpeta, el filtro inicial, y la localización física a partir de

la cual se muestran los resultados (los alcances). El componente constructor de consultas 220 envía consultas SQL a la base de datos relacional 230.

En general, el componente constructor de consultas 220 incluye un conjunto de filas (en otras palabras una tabla). Esta es la razón por la cual el
5 correr una consulta produce resultados. El componente analizador de conjuntos de filas toma cada fila y transforma la fila en una lista ID usando los nombres de las columnas. Una lista ID es una estructura de *shell* ampliamente conocida que se usa para referenciar objetos en un espacio de nombres. El hacer esto permite que las carpetas virtuales sea iguales a cualquier otro
10 espacio de nombres, a los ojos del resto de la *shell*. Además, el guardar en memoria ("*caching*") estos datos ayuda al que el acceso a la base de datos, que puede ser costoso, sea mínimo.

El componente enumerador 222 opera en respuesta a la navegación hacia una carpeta virtual. Como se describe arriba, el componente enumerador
15 222 recibe listas ID del componente analizador de conjuntos de columnas 218, y envía nuevos filtros desde la navegación hacia el componente constructor de consultas 220. El enumerador 222 también envía datos al navegador de la *shell* y al componente de visualización 240, incluyendo listas ID que son devueltas para ser insertadas en la vista después de la navegación.

20 El componente de fabricación de propiedades 224 toma las listas ID y los identificadores de propiedades y devuelve los valores de estas propiedades. El componente de fabricación de propiedades 224 recibe datos del componente de

fabricación de manejadores 214, incluyendo el manejador de propiedades. Como se describe arriba, el componente de fabricación de propiedades 224 también recibe datos del componente de otras carpetas de la *shell* 234, incluyendo propiedades de otras carpetas. El componente de fabricación de
5 propiedades 224 también envía datos al navegador de la *shell* y al componente de visualización 240, incluyendo propiedades de los objetos, de acuerdo con el requerimiento de la vista.

El navegador de la *shell* y el componente de visualización 240 despliega el contenido de una carpeta en una ventana, y maneja toda la interacción del
10 usuario con las carpetas y objetos desplegados, tal como la selección ("clicking"), arrastre y navegación. Así, el navegador de la *shell* y el componente de visualización 240 reciben las acciones del usuario. El navegador de la *shell* y el componente de navegación 240 también obtienen datos en relación con el comportamiento de los códigos que requiere de la
15 carpeta, en este caso del procesador de carpetas 210.

Como se describe arriba, las carpetas virtuales exponen carpetas y archivos comunes (también conocidos como directorios) ante los usuarios, en diferentes vistas con base en sus metadatos en lugar de basarse en su estructura física de sistema de archivos subyacente en el disco. Así, el sistema
20 puede tomar una propiedad que está almacenada en la base de datos y representarla como un contenedor que es como una carpeta. Debido a que los usuarios ya están familiarizados con el trabajo con carpetas, al representar las

carpetas virtuales de una forma similar, los usuarios pueden adaptarse al nuevo sistema más rápidamente.

La FIGURA 3 es un diagrama de flujo que ilustra una rutina 300 por medio de la cual un usuario proporciona una consulta que le retorna los objetos seleccionados. En el bloque 302, el procesador de carpetas obtiene una consulta del usuario. En el bloque 304 el procesador de carpetas le pasa la consulta a la base de datos relacional. En el bloque 306 la base de datos relacional proporciona los resultados de vuelta al procesador de carpetas. En el bloque 308 el procesador de carpetas suministra los resultados al usuario en forma de carpetas y objetos virtuales.

La FIGURA 4 es un diagrama de flujo que ilustra una rutina 320 por medio de la cual las carpetas virtuales se construyen y se despliegan en la pantalla de acuerdo con ya sea una consulta predeterminada o con una consulta del usuario. En el bloque 322, cuando un usuario abre la carpeta virtual por primera vez, se utiliza una consulta predeterminada. Esta consulta predeterminada se toma del registro. Por ejemplo, la consulta predeterminada para una librería musical podría ser mostrar todas las canciones agrupadas por álbum. En el bloque 324, el procesador de carpetas construye un objeto de consulta para esta consulta, y luego le pasa esta consulta a la base de datos relacional. En el bloque 326 la base de datos relacional genera los resultados de la consulta y los pasa de nuevo al procesador de carpetas en forma de filas y columnas de base de datos.

En el bloque 328, el procesador de carpetas toma estos resultados y los convierte de filas y columnas de datos en una estructura enumeradora, que es usada por la vista de carpetas para poblar la pantalla con las carpetas y objetos virtuales resultantes con el fin de que el usuario pueda interactuar sobre ellos.

- 5 En el bloque de decisión 330, un usuario decide sobre el cambio de vista (con la expedición de una consulta diferente o "pivote"). Por ejemplo, un usuario podría expedir un pivote de "muestre todos los artistas". Si el usuario no desea cambiar la vista, entonces la rutina retorna el bloque 324 en la cual el procesador de carpetas pasa esta nueva consulta a la base de datos relacional,
- 10 y recibe de nuevo filas y columnas de resultados, y construye una nueva estructura numeradora. El proceso continúa así como se describe arriba, a la vez que la vista de carpetas se borra y se actualiza, usando el enumerador para dibujar los objetos del "artista" sobre la pantalla.

- En uno de los ejemplos, se proporcionan objetos de álbumes que
- 15 representan contenedores dentro de los cuales el usuario puede navegar. Por ejemplo, haciendo una doble selección (doble-"clic") sobre los álbumes de los "Beatles" proporcionará una navegación en la vista para ver todas las canciones de los Beatles. El procesador de carpetas envía una consulta de "mostrar todas las canciones de los Beatles" a la base de datos relacional, la cual devuelve las
- 20 filas y las columnas con los datos de las canciones. El proceso de cambiar la vista de objetos a otra representación se llama agrupamiento ("*stacking*"). Esto ocurre porque los objetos están conceptualmente organizados en "grupos"

(pilas) con base en esa representación. En este caso, las canciones se reorganizan en grupos (pilas) para cada uno de los álbumes. Los usuarios pueden navegar entre uno de estos grupos, viendo únicamente las canciones de un álbum en particular. De nuevo, el usuario puede reorganizar la vista de las canciones restantes en grupos, con base en una propiedad (p.ej. una calificación). Si se seleccionara la propiedad de calificación, las canciones del álbum de los Beatles se mostraría agrupada (en pilas) de acuerdo a una calificación de una, dos o tres estrellas.

Los resultados de cada consulta dependen de cuáles localizaciones físicas se incluyen en el alcance. Por ejemplo, el alcance puede definirse para incluir solamente las carpetas en la carpeta de "Mis Documentos" del usuario. Alternativamente, el alcance podría incluir todas las carpetas en el computador, o aún todas las carpetas en computadores múltiples conectados en red. El usuario puede visualizar y cambiar el alcance a través de una hoja de propiedades del alcance. En uno de los ejemplos, la hoja de propiedades de alcance podría ser expuesta mediante la selección con el botón derecho del ratón sobre una carpeta virtual y de esta forma seleccionar las "propiedades". El usuario podría agregar nuevas carpetas al alcance, o remover carpetas que habían sido agregadas.

Un grupo de usuarios para el cual las carpetas virtuales pueden proporcionar una utilidad particular es el grupo de trabajadores del conocimiento. Las carpetas virtuales le permiten a los trabajadores del

conocimiento cambiar fácilmente la visualización de documentos según tipo de archivo, proyecto, número de caso, autor, etc. Debido a que los trabajadores del conocimiento tienden a tener cada uno un método diferente para organizar los documentos, las carpetas virtuales se pueden usar para acomodar este tipo de preferencias diferentes.

La FIGURA 5 es un diagrama de árbol de una estructura de carpetas de acuerdo con un arreglo físico de las carpetas en la unidad de disco duro. Este arreglo físico de carpetas se basa en la implementación tradicional de carpetas, que puede basarse en NFTS o en otros sistemas de archivos existentes. A estas carpetas se les refiere como carpetas físicas puesto que su estructura se basa en el sistema de archivos físico subyacente actual en el disco. Como se describirá con mayor detalle a continuación, esto contrasta con las carpetas virtuales, que crean vistas independientes de las localizaciones, que le permiten a los usuarios manipular los archivos y las carpetas en formas similares a las que actualmente se usan para manipular las carpetas físicas.

Como lo muestra la FIGURA 5, una carpeta 400 se encuentra en la carpeta "Mis documentos". En el primer nivel, la carpeta 400 incluye las carpetas 410, 420 y 430, correspondientes a los Clientes 1, 2 y 3 respectivamente. En el segundo nivel, cada una de las carpetas 410, 420 y 430 contienen las carpetas 411, 421 y 431 respectivamente cada una de las cuales corresponde al año 2001. En el tercer nivel, cada una de las carpetas 411, 421

y 431 también contienen las carpetas 413, 423 y 433 respectivamente, cada una de las cuales corresponde al año 2002.

Se verá que surgen una serie de obstáculos ante el usuario que desea navegar hacia una estructura física de carpetas tal como la que se ilustra en la FIGURA 5. Por ejemplo, si el usuario desea trabajar con todos los contratos que el usuario ha producido, el usuario tendría primero que navegar hacia la carpeta 411 para trabajar con los contratos del Cliente 1 y luego tendría que volver a navegar a la carpeta 421 para obtener los Contratos del Cliente 2 y luego tendría nuevamente que navegar hacia la carpeta 431 para ver los contratos del Cliente 3. Esta organización le dificulta al usuario el acceso a todos los contratos, y en general le impide visualizar y manipular simultáneamente todos los contratos. De igual manera, si el usuario desea visualizar todos los contratos correspondientes al año 2001, el usuario tendrán que volver a navegar por las carpetas 412, 422 y 432 respectivamente. Como se describirá en mayor detalle más adelante, las carpetas virtuales de la presente invención proporcionan una estructura de sistema de archivos mejorada.

La FIGURA 6 es un diagrama de árbol de una estructura de carpetas virtuales. Como se describirá en mayor detalle más adelante, las carpetas virtuales crean vistas independientes de la localización que le permiten a los usuarios manipular sus archivos y carpetas en formas convenientes. Como se muestra en la FIGURA 6, las carpetas virtuales se representan como grupos (pilas). Una carpeta virtual 500 es una carpeta de "todos los objetos". En el

primer nivel, la carpeta virtual 500 contiene las carpetas virtuales 510, 520 y 530, correspondientes a los clientes, contratos y año respectivamente. Como se describirá en mayor detalle más adelante, esta estructura le permite al usuario tener acceso a los archivos de acuerdo con el parámetro deseado.

5 La FIGURA 7 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 6, donde en el segundo nivel la carpeta virtual 510 además incluye las carpetas virtuales 511 y 512, que corresponden a contratos y al año, respectivamente. En otras palabras el grupo de clientes de la carpeta virtual 510 se filtra además por contratos y por año. El proceso para determinar
10 cuáles archivos y objetos están contenidos en cada una de las carpetas virtuales se describirá en más detalle a más adelante.

 La FIGURA 8 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 7, donde en un tercer nivel la carpeta virtual 511 además incluye la carpeta virtual 513 que corresponde al año. En otras
15 palabras el grupo de contratos de la carpeta virtual 511 además esta filtrado por año. Mientras que la estructura de carpetas virtuales 510, 511 y 513 se han estructurado de acuerdo a los clientes, contratos y años, se apreciará que las carpetas virtuales permiten la ocurrencia de otras secuencias de estructura, como se describirá mas adelante en referencia de la FIGURA 9.

20 La FIGURA 9 es un diagrama de árbol de la estructura de carpetas virtuales de la FIGURA 6, donde en un segundo nivel la carpeta virtual 520 además ha sido filtrada en las carpetas virtuales 521 y 522, correspondientes a

clientes y a año. En un tercer nivel la carpeta virtual 521 además ha sido filtrada en la carpeta virtual 523, correspondiente al año. El contraste entre las estructuras organizacionales de las FIGURAS 8 Y 9 ayuda a ilustrar la flexibilidad del sistema de carpetas virtuales. En otras palabras, en el sistema de carpetas virtuales un usuario puede navegar por las carpetas virtuales de acuerdo con los parámetros deseados, y no depende de las vistas dependientes de la localización de una estructura física de archivos tal como se ilustra en la FIGURA 5.

La FIGURA 10 es un diagrama ilustrativo de un despliegue de pantalla 600 que muestra los grupos (pilas) de una librería de documentos. Como se anotó arriba, los grupos se pueden usar para representar un tipo de carpeta virtual. Como se describirá en más detalle a continuación, el un despliegue de pantalla 600 incluye elementos de enlaces rápidos 610-613, elementos de filtro 620-626, elementos de actividad 630-633, elementos de información y control 640-645 y grupos de carpetas virtuales 651-655.

Los elementos de enlaces rápidos incluyen un enlace 610 a "todas las categorías", otro enlace rápido 611 a "todos los autores", un enlace rápido 612 a "trabajos de Enero" y una selección para desplegar enlaces rápidos adicionales 613. Como se describirá en más detalle abajo, los enlaces rápidos se pueden seleccionar por parte de un usuario para llevar a cabo navegaciones deseadas por las carpetas virtuales. Los enlaces rápidos pueden ser suministrados por el

sistema, y algunos de los enlaces rápidos pueden ser creados y guardados por el usuario.

Los elementos de filtro incluyen un indicador 620 "por filtro" una entrada en blanco 621, un indicador 622 "por fecha", un selector 623 de "año", un selector 624 para "escoger autor", un selector 625 para "escoger categoría", y ", un selector 626 para "más filtros". El indicador 620 "por filtro" dirige al usuario al hecho de que los objetos a continuación s pueden usar para filtrar las carpetas u objetos virtuales. La una entrada en blanco 621 proporciona un área en la cual un usuario puede escribir un nuevo término de filtro deseado. El indicador 622 "por fecha" dirige al usuario al hecho de que mediante la selección de una fecha del selector 623 de "año", las carpetas u objetos virtuales pueden filtrarse por el año seleccionado. El selector 624 para "escoger autor" le permite al usuario filtrar de acuerdo con un autor específico. El selector 625 para "escoger categoría" le permite al usuario filtrar de acuerdo con una categoría seleccionada. El selector 626 para "más filtros" le permite al usuario recuperar más filtros sobre el despliegue.

Los selectores de actividad incluyen un selector 630 de "crear unja nueva categoría", los selectores 631 y 632 de "actividad" y el selector 633 de "más actividades". Como se describirá en más detalle a continuación, las actividades que se presentan pueden ser para funciones generalmente deseables, o pueden dirigirse específicamente al tipo de carpetas u objetos que se están desplegando. Por ejemplo, el despliegue actual es de una librería de

documentos, para la cual los selectores de "actividad" 631 y 632 pueden direccionarse a actividades específicamente personalizadas para documentos, tales como editar o crear anexos. Si la librería actual hubiese sido un librería de fotografías, los selectores de "actividad" 631 y 632 pueden direccionarse
5 hacia actividades específicamente dirigidas a las fotografías, como conformar álbumes de fotografías o compartir fotografías con otros usuarios.

La información y los elementos de control incluyen las líneas de información 640 y 641, una línea de control 642, un control de retroceso de espacio 643 y las líneas de información 644 y 645. Las líneas de información
10 640 y 641 proporciona información referente a la navegación actual de las carpetas u objetos virtuales. En este ejemplo, la línea de información 640 indica que la navegación actual ocurre en una librería de documentos, mientras que la línea de información 641 indica una navegación más completa, mostrando que la librería de documentos se encuentra dentro del área de almacenamiento. La
15 línea de control 642 suministra un número de controles estándar y el botón de retroceso de espacio 643 le permite al usuario retroceder durante la navegación. La línea de información 644 suministra información numérica acerca del contenido de la navegación actual. En este ejemplo, la línea de información 644 indica que había 41 objetos que ocupaban 100MB en los grupos de la
20 librería de documentos. La línea de información 645 está disponible para suministrar información adicional, tal como la información adicional acerca de un archivo seleccionado.

Los grupos de la librería de documentos incluyen un grupo 651 "ABC Corp.", un grupo 652 "grupo de respaldos", un grupo 653 de "planes de negocios", un grupo 654 de "XYZ Corp." Y un grupo 655 de "informes de mercadeo". Los números en la parte superior de cada uno de los grupos indican
5 cuántos objetos se encuentran en cada grupo. Por ejemplo, el grupo 651 "ABC Corp." muestra que incluye 8 objetos. El número total de objetos de los grupos se suma al número de objetos indicados en la línea de información 644, la cual como se describe arriba es 41 en este ejemplo. Se proporciona una casilla de selección SB que un usuario puede utilizar para seleccionar un objeto deseado.
10 La selección de "ABC Corp." resulta en una vista de los objetos del grupo, como se describe a continuación respecto de la Figura 11.

La FIGURA 11 es un diagrama ilustrativo de una representación de pantalla que muestra los objetos en el grupo "ABC Corp." 651 de la Figura 10. Se debe anotar que las líneas de información 640 y 641 indican ahora que la
15 navegación actual muestra el grupo "ABC Corp.". El grupo "ABC Corp." 651 se muestra incluyendo 8 documentos 751-758, que corresponden respectivamente a los documentos 1-8. La línea de información 644 indica en forma correspondiente que existen 8 objetos que ocupan 20MB de memoria. Además, los documentos de la FUFURA 11 se pueden organizar en grupos dentro del
20 grupo "ABC Corp.". En otras palabras, dentro de la carpeta virtual representada por el grupo ABC Corp 651, adicionalmente se pueden organizar diferentes

carpetas virtuales para contener los elementos, como se describe a continuación respecto de las FIGURAS 12-16.

La FIGURA 12 es un diagrama ilustrativo de una representación de pantalla en la cual la función de agrupamiento se selecciona para los documentos de la FIGURA 11. Como se muestra en la FIGURA 12, el usuario es capaz de seleccionar la casilla de funciones 760. La casilla de funciones 760 incluye un selección de "vista" 761, una selección de "organizar íconos según" 762, una selección de "grupos" 763, un selección de "actualización" 764, una selección de "abrir las carpetas contenidas" 765, una selección de "corte" 766, una selección de "copiar" 767, y una selección de "deshacer" 768, una selección de "nueva" 769, y una selección de "propiedades" 770. La casilla de selección SB se muestra alrededor de la selección de grupos 763.

La FIGURA 13 es un diagrama ilustrativo de una representación de pantalla en la cual el parámetro "agrupamiento por autor" se selecciona de la función de agrupamiento de la FIGURA 12. Como se muestra en la FIGURA 13, se muestra una casilla 780, la cual representa varias funciones de agrupamiento. La opción de agrupamiento incluye una opción de "desagrupar" 781, una opción de "agrupar por categoría" 782, un opción de "agrupar por autor" 783, y una opción de "agrupar por autor" 783.

La FIGURA 14 es un diagrama ilustrativo de una representación de pantalla en la cual los archivos de la FIGURA 13 han sido agrupados por autor. Como se muestra en la FIGURA 14, los grupos 791 y 792 corresponden a los

autores Bob y Lisa respectivamente. Como lo indican los números en la parte superior de los grupos, el grupo Bob 791 incluye dos objetos, mientras que el grupo Lisa 792 incluye cinco objetos. El objeto 785 (que corresponde al documento 8) no tenía autor, y por esa razón no está incluido en el grupo de "autores". Los grupos 791 y 792 ilustran el hecho de que los grupos se pueden organizar en niveles múltiples, tal como lo muestra el grupo 651 "ABC Corp." Así, las carpetas virtuales se pueden formar en niveles múltiples, tal como en el grupo "Lisa" 792, que está dentro del grupo "ABC Corp." 651, el cual están dentro de la librería de documentos.

La FIGURA 15 es un diagrama ilustrativo de una representación de pantalla en la cual además se selecciona una opción de "agrupar por categoría", para el reagrupamiento de los archivos de la FIGURA 14. Como se muestra en la FIGURA 15, la casilla de selección SB gira en torno a la opción "agrupar por categoría" 782. Dado que algunos objetos ya se encuentran agrupados en los grupos 791 y 792, la selección de la opción "agrupar por categoría" 782 reagrupará los objetos, como se describe en mayor detalle a continuación en relación con la FIGURA 16.

La FIGURA 16 es un diagrama ilustrativo de una representación de pantalla en la cual los archivos de la FIGURA 14 se reagrupan por categoría. Como se muestra en la Figura 16, los grupos 793 y 794 corresponden a las categorías de "XYZ Corp." e "informes d mercadeo" respectivamente. Los objetos 751 y 752 corresponden a los documentos 1 y 2, no fueron designados

para categorías adicionales, en consecuencia no corresponden a ninguna categoría entre los demás grupos de categorías.

La FIGURA 17 es un diagrama ilustrativo de una representación de pantalla en la cual se selecciona un enlace rápido para carpetas físicas. La casilla de selección SB se muestra girando en torno al enlace rápido de "todas las carpetas" 616. Como se describe en mayor detalle a continuación en relación con la FIGURA 18, el enlace rápido de "todas las carpetas" 616 proporciona la forma de cambiar a una vista que muestra las carpetas físicas.

La FIGURA 18 es un diagrama ilustrativo de una representación de pantalla que muestra las carpetas físicas. Las carpetas físicas que se muestran contienen los archivos de los grupos virtuales de carpetas de la FIGURA 17. En otras palabras, los objetos contenidos dentro de los grupos 651-655 de la FIGURA 17 también están contenidos en algunas carpetas físicas en el sistema. Estos se muestran en la FIGURA 18 como la carpeta "Mis Documentos" 851 que está localizada en este computador, una carpeta de "Mi Escritorio" 852 localizada en este computador, una carpeta "Foo" 853 localizada en la unidad de disco duro C; una carpeta de "Mis archivos" 854 localizada en un servidor, una carpeta de "Unidad Externa" 855 localizada en una unidad externa, una carpeta "Mis Documentos" 856 localizada en otro computador, una carpeta "Mi Escritorio" 857 localizada en otro computador.

Como se muestra en la FIGURA 18, un usuario puede cambiar la representación de archivos virtuales de la FIGURA 17 a la representación física

de la FIGURA 18. Esto le permite al usuario cambiar entre representaciones virtuales de archivos a representaciones físicas de archivos, dependiendo en cuál desea para la tarea actual. Las diferentes localizaciones de las carpetas físicas 851-857 también ilustran que el alcance del sistema virtual de archivos puede ser relativamente amplio, como se describirá en más detalle a continuación.

La FIGURA 19 es un diagrama de flujo que ilustra una rutina 880 por medio de la cual un usuario puede manipular directamente las carpetas virtuales. Como se describe en mayor detalle a continuación, los mecanismo que se suministran para manipular las carpetas virtuales son similares a aquellos actualmente usados para manipular carpetas comunes (p. ej, seleccionar y arrastrar, copiar, pegar etc.). Como se muestra en la FIGURA 19, en el bloque 882, el sistema suministra acciones definidas que el usuario puede activar para manipular directamente las carpetas virtuales que están representadas por objetos desplegados. En el bloque 884 el usuario desarrolla una acción definida. Como se anotó arriba, un ejemplo de esto puede ser un usuario seleccionando y arrastrando una carpeta virtual para copiar su contenido en otra carpeta virtual. En el bloque 886, la carpeta virtual y/o su contenido se manipula en la forma indicada por la acción activada por el usuario.

La FIGURA 20 es un diagrama ilustrativo de una representación de pantalla en la cual un nuevo grupo de la Costa Oeste 656 ha sido adicionado a

los grupos de la FIGURA 10. El grupo Costa Oeste 656 fue formado por un usuario que creó la nueva categoría "Costa Oeste". En su creación inicial, el nuevo grupo Costa Oeste 656 estaría vacío y no contendría elementos. En la modalidad de la FIGURA 20, se han adicionado dos nuevos elementos al grupo

5 Costa Oeste 656. Uno de los métodos para adicionar objetos al grupo es seleccionar un objeto en particular, y ya sea modificar o adicionar nuevas categorías a la categoría de metadatos del objeto, tal como adicionar la categoría "Costa Oeste" a los dos objetos, como se hizo en la modalidad de la FIGURA 20. Este proceso ilustra que los datos de la categoría son una

10 propiedad de los metadatos de un objeto, que es un tipo de propiedad *ad hoc*. En otras palabras, una propiedad de este tipo no tiene un significado implícito, y cualquier usuario puede asignarle un valor arbitrario. Por ejemplo, la categoría "propiedad" puede tener cualquier valor, mientras que la propiedad "autor" debería ser el nombre de una persona. Como se describirá en mayor detalle a

15 continuación en relación con la FIGURA 21, los objetos también se pueden seleccionar y arrastrar para ser copiados desde otros grupos hacia el grupo West Coast 656 (en cuyo caso las categorías de los objetos se actualizarán automáticamente para incluir la propiedad "Costa Oeste"). A este respecto, la FIGURA 20 muestra que la casilla de selección SB gira en torno al grupo ABC

20 Corp 651, preparando su contenido para que sea copiado.

La FIGURA 21 es un diagrama ilustrativo de una representación de pantalla en la cual se usa la manipulación directa para copiar archivos del grupo

651 ABC Corp al grupo Costa Oeste 656. En otras palabras, como se muestra en la FIGURA 20, el usuario seleccionó el grupo ABC Corp 651, y luego, como se muestra en la FIGURA 21, el usuario ha seleccionado y arrastrado el grupo para que sea copiado en el grupo Costa Oeste 656. Así, el grupo Costa Oeste 5 656 que tenía dos objetos en la FIGURA 20, se muestra ahora incluyendo un total de diez objetos, incluyendo los ocho objetos adicionales del grupo ABC Corp 651. Cuando los objetos del grupo ABC Corp 651 se copiaron al grupo Costa Oeste 656, esto se logró mediante la modificación de las descripciones de las categorías de los ocho objetos, para que incluyeran la categoría "Costa 10 Oeste" adicionalmente a incluir la categoría original "ABC Corp". Esto ilustra un tipo de manipulación directa que se puede desarrollar.

Otro ejemplo de manipulación directa es la selección de un objeto con el botón derecho (del ratón) y seleccionando la opción "borrar". En una de la modalidades, cuando un usuario selecciona la función "borrar" se le pregunta al 15 usuario si el objeto se debe eliminar totalmente, o si solamente debe ser removido de la carpeta virtual actual. Si el objeto solamente debe ser removido del grupo de categorías de carpetas virtuales actuales, como se anotó arriba, esto se puede lograr mediante la remoción de la categoría deseada de los metadatos del objeto. En otras palabras, si uno de los objetos que fue copiado 20 desde el grupo ABC Corp. 651 al grupo Costa Oeste 656 se quitara del grupo Costa Oeste 656, esto se lograría con la modificación de los datos de la

categoría del archivo particular para que no incluyera la categoría "Costa Oeste".

La FIGURA 22 es un diagrama de flujo que ilustra una rutina 900 para que el sistema genere dinámicamente nuevos términos de filtro. Los términos
5 de filtro se utilizan para manipular carpetas virtuales. Los términos de filtro se utilizan esencialmente como un conjunto de herramientas para estrechar el contenido de un conjunto de objetos. En una de las modalidades, los filtros consisten en categorías de metadatos y sus valores (presentados al usuario en la interface de usuario enlaces seleccionables o como menús verticales que se
10 desenrollan). El usuario selecciona (hace "clic") en uno de los términos de filtro para filtrar los resultados actuales de un conjunto de objetos en la representación actual.

La FIGURA 22 ilustra cuántos filtros se pueden generar dinámicamente. Como se muestra en la FIGURA 22, en el bloque 902, se revisan las
15 propiedades (de los metadatos) de los objetos de la colección de la representación actual. En el bloque 904, se generan dinámicamente términos propuestos para filtros, con base en las propiedades comunes de los objetos. En el bloque 906, los términos propuestos de los filtros se le presentan al usuario para una posible selección para el filtrado de objetos. Como ejemplo de
20 este proceso, el sistema puede revisar las propiedades de un conjunto de objetos, y si los objetos generalmente tienen "Autores" como una de sus propiedades, el filtro puede suministrar una lista de los autores con la cual se

pueda aplicar el filtro. Después, al seleccionar un autor específico, los objetos que no contienen a dicho autor se quitan del conjunto que se encuentra desplegado. El proceso de filtro le proporciona al usuario un mecanismo para reducir el conjunto de objetos representados.

5 La FIGURA 23 es un diagrama de flujo que ilustra la rutina 902 para un sistema de filtrado de objetos basado en una selección de un término de filtro. En el bloque 922, el usuario entra un nuevo término de filtro o selecciona uno de los términos de filtro que presenta el sistema. Como se anotó arriba, los términos de filtro se pueden generar dinámicamente por el sistema, o se
10 pueden predefinir. En el bloque 924, se evalúan los objetos de la colección desplegada, en relación con el hecho de si sus propiedades seleccionadas coinciden el término del filtro. Por ejemplo, si el término de filtro para objetos con autores fuese "Bob", entonces los objetos serán evaluados de conformidad con el hecho de si la propiedad de "autor" incluye a "Bob". En el bloque 926, los
15 objetos para los cuales las propiedades seleccionadas no coinciden con el término de filtro, son retirados de la colección en el despliegue.

Las FIGURAS 24-29 generalmente ilustran cómo el proceso de filtro aparece en el despliegue de la pantalla. Como se describe a continuación en relación con las FIGURAS 24-29, en una de las modalidades, el filtrado puede
20 operar generalmente de acuerdo al siguiente proceso. Después de que el usuario selecciona un valor de filtro, los objetos que no estén dentro del rango del filtro desaparecen de la pantalla en forma de animación. La animación

generalmente se diseña para hacer obvio el hecho de que los objetos están siendo removidos y que no se están agregando nuevos elementos. El botón de atrás 643 se puede seleccionar por parte de un usuario para deshacer las operaciones de filtrado. En una de las modalidades, se crea un grupo de navegación, que contiene las acciones secuenciales del filtro, lo que se utiliza para deshacer cada una de las acciones de filtrado cuando se selecciona el botón de retroceso 643. Cada vez que se selecciona un valor de filtro, las áreas de información 640 y 641 se actualizan para indicar el valor actual del filtro. En una de las modalidades, después de seleccionar un valor de filtro, se le proporciona al usuario la opción de guardar un nuevo enlace rápido a la navegación actual con filtros, como se describirá en más detalle a continuación con respecto de la FIGURA 30. Mientras se seleccionan los valores de filtro, los controles de filtro pueden actualizarse en tal forma que permanezcan adecuados para los objetos que quedan en la vista.

La FIGURA 24 es un diagrama ilustrativo de una representación de pantalla en la cual los grupos de la FIGURA 10 se han filtrado con el término "AB". Como se muestra, en el área de filtro 621, el término "AB" ha sido digitado por parte de un usuario. Las líneas de información 640 y 641 indican que los objetos en el despliegue son ahora aquellos que han sido filtrados por el término "AB". Como se muestra, el grupo ABC Corp. 651 aún contiene ocho objetos, mientras que el grupo de RespalDOS 652 ahora contiene tres objetos, y el grupo XYZ Corp 654 también contiene tres objetos. La línea de información

644 indica entonces que existe un total de 14 objetos que ocupan un total de 35 MB de memoria.

La FIGURA 25 es un diagrama ilustrativo de una representación de pantalla en la cual los grupos de la FIGURA 10 han sido filtrados con el término "ABC". En relación con el término de "AB" de la FIGURA 24, el usuario simplemente ha digitado adicionalmente la letra "C" para construir el término total de filtro "ABC". Como lo muestra la FIGURA 25, las líneas de información 640 y 641 ahora indican que los elementos en el despliegue son aquellos que contienen el término "ABC". El grupo ABC Corp. 651 aún se muestra conteniendo ocho objetos, mientras que el grupo Respaldos 652 ahora contiene solamente dos objetos. El grupo XYZ Corp ha desaparecido porque ninguno de sus contenidos coincidió con el filtro "ABC". La línea de información 644 ahora indica que hay un total de 10 elementos en los grupos en el despliegue, que ocupan 25 MB de memoria. Las FIGURAS 24 y 25 suministran en esta forma los ejemplos de cómo un usuario puede definir nuevos términos de filtro, y cómo dichos términos de filtro pueden usarse para filtrar los elementos que se muestran en el despliegue.

El botón de retroceso 643 puede utilizarse por parte de un usuario para retroceder a través del proceso de filtrado. Como se describe arriba respecto de la FIGURA 10, el botón de retroceso 643 le permite al usuario de retroceder durante la navegación. En relación con los ejemplos de las FIGURAS 24 y 25, después de filtrar con el término "ABC" en la FIGURA 25, un usuario podrá

seleccionar el botón de retroceso 643 para retroceder un paso el proceso de filtrado, lo que retornaría al estado de la FIGURA 24. Alternativamente, en otra modalidad, el botón de retroceso 643 podrá borrar por completo la totalidad del término de filtro, y en esta forma retornar al estado anterior al cual ocurrió dicho filtrado. En este caso, oprimiendo el botón de retroceso 643 del a FIGURA 25, un usuario podrá retornar al estado de la FIGURA 10.

En una de las modalidades, adicionalmente al botón de retroceso, se suministra un medio adicional para que un usuario retroceda durante o de otra forma modifique la navegación de filtrado. Esto adicionalmente incluye el permitirle a un usuario tener acceso directo y modificar la línea de información 641, que cambia en forma correspondiente la navegación de filtrado. En otras palabras, mediante el acceso directo y la modificación de la línea de información 641, el usuario puede remover uno o más de los filtros aplicados. Esta característica se describe en más detalle en la solicitud de patente de EE.UU. No. 10/420,040 presentada el 17 de Abril, 2003, la cual se cede comúnmente y se incorpora aquí mediante referencia en su totalidad.

También se puede utilizar un temporizador en conjunto con un usuario digitando términos de filtro tales como aquellos que se muestran en las FIGURAS 24 y 25. El temporizador se usa para supervisar las pausas hechas por el usuario al digitar. Luego de un intervalo donde no hay acción de digitación, se aplica el filtro. Por ejemplo, en el estado de la FIGURA 24, un usuario ha digitado el término de filtro "AB" sin un retraso de tiempo entre la "A"

y la "B". Luego de digitar el término "AB" el usuario hace una pausa, produciendo así el estado de la FIGURA 24, donde el término "AB" se aplica. Un tiempo después, el usuario agrega la letra "C" para completar el término de filtro "ABC", y luego hace una pausa de nuevo, y en este punto se aplica el término de filtro "ABC" como se ilustra en la FIGURA 25.

En una de las modalidades, después de que un usuario ha digitado el término de filtro en el área de filtro 621, y luego escoge otro filtro u otra navegación, el estado de navegación se actualiza, y el término de filtro en el área de filtro 621 se desocupa nuevamente. Adicionalmente, como se describirá en mayor detalle a continuación en relación con las FIGURAS 26-29, otros controles de filtro se pueden actualizar con base en la selección de ciertos términos de filtro.

La FIGURA 26 es un diagrama ilustrativo de una representación de pantalla en la cual el sistema suministró la selección del término de filtro "año 2002". Como se anotó arriba, bajo el indicador de fecha 622, las selecciones de año 623 incluyen los años 200, 2001 o 2002. La casilla de selección SB se muestra en torno al año 2002, indicando que el usuario está seleccionando aquel como el término de filtro deseado.

La FIGURA 27 es un diagrama ilustrativo de una representación de pantalla en la cual se aplicó el término de filtro "2002". También se muestra la selección adicional del selector 623A "escoja un mes". Como se muestra en la FIGURA 27, luego de aplicar el término de filtro 2002, el número de elementos

en los grupos se ha reducido. Más específicamente, el grupo ABC Corp. 651 ahora contiene seis elementos, el grupo de Respaldos 652 ahora contiene ocho elementos, el grupo de Planes de Negocios 653 contiene ahora tres elementos, y el grupo XYZ Corp 654 contiene ahora cinco elementos. La línea de información 644 ahora indica un total de 22 elementos que ocupan un total de 50 MB de memoria. Las líneas de información 640 y 641 ahora indican que los elementos mostrados en el despliegue son aquellos que han sido filtrados para contener el término "2002".

La FIGURA 28 es un diagrama ilustrativo de una representación de pantalla en la cual se presenta una lista para seleccionar el mes de filtro. Se suministra una casilla 950 que incluye las listas de los meses. La casilla 950 se suministra en el despliegue por que un usuario seleccionó el selector de "escoja un mes" 623A. La casilla de selección SB se muestra alrededor del mes de Enero.

La FIGURA 29 es un diagrama ilustrativo de una representación de pantalla en la cual los grupos de la FIGURA 28 se ha filtrado además con el mes de Enero, y además muestran el término de filtro "día". Como se muestra en la FIGURA 29, las líneas de información 640 y 641 ahora indican que los objetos en los despliegues son aquellos filtrados por el término "Enero". El grupo de Respaldos 652 se muestra incluyendo dos objetos, mientras que el grupo de Planes de Negocios 653 también se muestra incluyendo dos objetos. La línea de información 644 indica que hay un total de cuatro elementos en el

despliegue, que ocupan un total de 10 MB de memoria. Se suministra el selector de "escoja por día" 623B, en caso de que el usuario quisiera filtrar adicionalmente los resultados de acuerdo a un día específico.

Como se describe arriba respecto de las FIGURAS 24-29, los términos
5 de filtro se pueden presentar por parte del sistema o el usuario los puede digitar. Una vez se selecciona un filtro, los demás términos de filtro que se presentan pueden ser actualizados (e.g. después de seleccionar el término "2002" en la FIGURA 26, las opciones para seleccionar un año no se presentan en la FIGURA 27, y en su lugar se presenta la opción "escoja un mes"). Como
10 se anota arriba, el botón de retroceso 643 puede ser seleccionado por parte de un usuario para retroceder el proceso de filtrado. Por ejemplo, después de seleccionar el mes "Enero" en la FIGURA 29, el usuario puede seleccionar el botón de retroceso 643 para retroceder el proceso de filtrado al año "2002", como se ilustra en la FIGURA 27. El menú de filtros puede también incluir una
15 función de "agrupar por", que operaría en forma similar a la función "agrupar por" descrita anteriormente en relación con las FIGURAS 15 y 16. Por ejemplo, un filtro por "tipo de archivo" podría tener las opciones de "Excel" Power Point", "Word", y además "Agrupar por tipo de archivo". Escoger la función "agrupar por" cambia la vista para mostrar grupos de diferentes tipos de archivos.

20 En general, los filtros se pueden configurar para aplicarse sobre las diferentes propiedades de los archivos u objetos. En una de las modalidades, los filtros se pueden clasificar de acuerdo con diferentes tipos, tales como:

índice alfabético; valores discretos; fechas; y rangos numéricos. Las propiedades de ejemplo del índice alfabético pueden incluir el nombre del archivo, autor, artista, nombre amigable del contacto, propietario, autor del documento, título del documento, tema del documento, y descripción. Las

5 propiedades de valores discretos pueden incluir localización, tipo de archivo, (nombre de la aplicación), género, pista, década (para música), calificación (para música), tasa de bits, protección, categoría del documento, contador de páginas del documento, comentarios al documento, modelo de la cámara, dimensiones, nombre del producto, versión del producto, imagen X, imagen Y, y

10 momento de creación del documento. Ejemplos de propiedades para fechas pueden incluir último acceso, última modificación, momento de creación, fecha de toma (para fotografías). Un ejemplo de propiedades para un rango numérico puede ser el tamaño del archivo.

Se apreciará que los filtros descritos respecto de las FIGURAS 24-29 le

15 permiten a los usuarios reducir la lista de objetos para encontrar un objeto en particular que es de interés. Como ejemplo específico, de acuerdo con los procesos descritos anteriormente, un usuario podría reducir una lista actual de documentos, para que únicamente muestre los archivos de Microsoft Word, creados por una persona en particular y editados durante la semana pasada.

20 Esta funcionalidad le permite a un usuario encontrar un objeto en particular dentro de una lista de muchos, y le ayuda al usuario evitar tener que hacer búsquedas manuales de cada objeto en la lista.

La FIGURA 30 es un diagrama de flujo que ilustra una rutina 940 para la creación de un enlace rápido. Como se describirá con mayor detalle a continuación, los enlaces rápidos son enlaces predefinidos que pueden ser seleccionados por parte de un usuario para crear vistas seleccionadas por el usuario de conjuntos de objetos. En una de las modalidades, un enlace rápido se puede pensar como un tipo de pivote. Los enlaces rápidos proporcionan un mecanismo para recuperar una carpeta virtual. La selección de un enlace rápido puede llevar al usuario a la carpeta deseada (en la misma forma en que la selección de "favoritos" puede llevar a un usuario a un sitio en Internet.) Los enlaces rápidos pueden estar predefinidos en el sistema, o pueden ser establecidos por el usuario. Por ejemplo, la selección de "todos los autores" podría retornar una vista agrupada de autores. La selección de "todos los documentos" podría retornar una vista plana de todos los documentos en todas las áreas de almacenamiento. Los usuarios pueden crear sus propios enlaces rápidos.

Como se muestra en la FIGURA 30, en el bloque 942, un usuario hace una selección en el despliegue para indicar que se debe conformar un nuevo enlace rápido a partir del término de filtro o de la navegación actual. En el bloque 944, el usuario proporciona el nuevo nombre para el enlace rápido. En el bloque 946 el enlace rápido se guarda y se suministra el nuevo nombre del enlace rápido en la sección de enlaces rápidos en el despliegue.

La FIGURA 31 es un diagrama ilustrativo de una representación de pantalla para crear un nuevo enlace rápido llamado "Trabajo de Enero" con base en el filtrado de la FIGURA 29. Como se describió arriba, en la FIGURA 29, los grupos se han filtrado con el mes de Enero. En la FIGURA 31, el usuario ha indicado que el filtrado de la FIGURA 29 se debe guardar como un nuevo enlace rápido, y ha nombrado el enlace rápido como "Trabajo en Enero". Así, el nuevo enlace rápido de Trabajo en Enero 612 se muestra en la sección de enlaces rápidos en despliegue. En relación con la formación de nuevos enlaces rápidos, generalmente se le suministra al usuario una opción tal como "guarde esta colección como enlace rápido".

La FIGURA 32 es un diagrama ilustrativo de una representación de pantalla en la cual se selecciona un enlace rápido de "Todos los Autores". Como se muestra en la FIGURA 32, la casilla de selección SB se muestra alrededor de la selección Todos los Autores 611. Otros ejemplos de colecciones a las cuales se puede tener acceso mediante enlaces rápidos incluye "todos los autores", "documentos recientes", todos los documentos que he compartido", todos los documentos de mi autoría", todos los documentos que no son de mi autoría", "escritorio", y "todos los tipos".

La FIGURA 33 es un diagrama ilustrativo de una representación de pantalla en la cual se presenta una lista de todos los autores de los elementos de la FIGURA 32. Como se muestra en la FIGURA 33, se suministra una línea de información 950, la cual indica columnas que muestran el nombre del objeto,

el autor, la fecha de modificación, el tipo, el tamaño, y la localización de un objeto. La lista de Autores 951-954 se muestra, correspondiente a los Autores 1-4 respectivamente.

La FIGURA 34 es un diagrama ilustrativo de una representación de
5 pantalla en la cual el "Autor 1" se ha seleccionado de la lista de la FIGURA 33. Los documentos del Autor 1 incluyen los documentos 951A y 951B correspondientes a los documentos 1 y 2 respectivamente. El documento 951A se muestra con la autoría del Autor 1, modificado en Julio 11, 2001, es un archivo de Microsoft Excel, ocupa 282 Kb de Memoria, y se obtuvo de la
10 localización \\servidor1\carpeta2. El documento 951B se muestra con la autoría del Autor 1, modificado en Diciembre 22, 2002, es un archivo de Microsoft Word, ocupa 206 Kb de Memoria, y físicamente se guarda en la localización Mis Documentos\carpeta1. Las localizaciones de los documentos 951A y 951B también ilustran que las carpetas virtuales de la presente invención pueden
15 contener objetos desde diferentes localizaciones físicas, como se describe en más detalle a continuación.

La FIGURA 35 es un diagrama de flujo que ilustra la rutina 960 para crear una nueva librería. Un ejemplo de librería es la librería de documentos descrita anteriormente en referencia a la FIGURA 10. En general, las librerías
20 consisten en grandes grupos de tipos de archivos usables que se pueden asociar en conjunto. Por ejemplo, las fotografías podrían ser una librería, la música otra, y los documentos otra. Las librerías pueden suministrar

herramientas y actividades que estén relacionadas con el tipo particular de los objetos. Por ejemplo, en la librería de fotos puede haber herramientas y filtros para manipular fotos, tales como la creación de presentaciones de diapositivas o compartir fotos. Como se muestra en la FIGURA 35, en el bloque 962 se crea
5 una nueva librería para incluir objetos con unas características seleccionadas. En el bloque 964, los objetos seleccionados se agrupan en la librería. En el bloque 966 se suministran las herramientas y/o actividades relacionadas con las características seleccionadas de los objetos o de otras funciones deseadas.

La FIGURA 36 es un diagrama ilustrativo de una representación de
10 pantalla en la cual se muestra una colección de librerías disponibles. Como se muestra en la FIGURA 36, las librerías incluyen una librería de documentos 971, una librería de fotos y video 972, una librería de música 973, una librería de mensajes 974, una librería de contactos 975, una librería de Tv y películas 976, así como una librería de todos los objetos 977. La librería de todos los
15 objetos 977 se muestra incluyendo 275 objetos, que es el número total de objetos de todas las librerías en conjunto. La línea de información 644 indica un total de 275 objetos que ocupan un total de 700 MB de memoria. Se debe notar que la librería de documentos 971 es la librería que se describió anteriormente respecto de la FIGURA 10.

20 La FIGURA 37 es un diagrama de flujo que ilustra la rutina 990 para definir el alcance de una colección de carpetas virtuales. Como se describirá en mayor detalle a continuación, un sistema de carpetas virtuales es capaz de

representar objetos desde diferentes localizaciones físicas (e.g. diferentes unidades de disco duro, computadores diferentes, diferentes localizaciones en la red etc.) en tal forma que un usuario pueda tener acceso fácil a todos los objetos. Por ejemplo, a un usuario se le pueden presentar los archivos de
5 música de múltiples localizaciones físicas en un solo despliegue, para manipular los archivos en conjunto.

Como se muestra en la FIGURA 37, en el bloque 992 se define un alcance para las localizaciones físicas a partir de las cuales se obtienen los objetos. En el bloque 994, en respuesta a una consulta, los objetos se obtienen
10 de las localizaciones físicas definidas en el alcance. En el bloque 996 todos los objetos obtenidos a través de la consulta se presentan en una sola representación.

La FIGURA 38 es un diagrama de bloques que ilustra varias de las fuentes que pueden formar el alcance de una colección de carpetas virtuales.
15 Como se muestra en la FIGURA 38, el sistema 1000 puede incluir el computador actual 1010, un computador adicional 1020, un almacenamiento externo y removible 1030, y localizaciones sobre una red 1040. El alcance general 1001 se describe incluyendo todas las localizaciones físicas a partir de las cuales se obtienen los objetos de un usuario para crear colecciones. El
20 alcance puede ser establecido y modificado por el usuario. Como se anotó arriba, las otras figuras ilustran el hecho de que los objetos pueden provenir de diferentes localizaciones físicas, tal como en la FIGURA 34 que muestra

diferentes documentos que provienen de un servidor y de la carpeta Mis Documentos en el computador actual, y en la FIGURA 18 que muestra las carpetas físicas que están físicamente guardadas en localizaciones múltiples.

La FIGURA 39 es un diagrama de flujo que ilustra la rutina 1080 que
5 incluye objetos diferentes a archivos en una colección de carpetas virtuales. Los objetos diferentes a archivos se contrastan con objetos de archivos que están típicamente localizados en un almacenamiento físico de archivos. Ejemplos de objetos diferentes a archivos serían cosas como correos electrónicos, o contactos. Como se muestra en la figura 39, en el bloque 1082
10 se utiliza una base de datos para incluir objetos diferentes a archivos conjuntamente con objetos de archivos que pueden ser buscados mediante una consulta. En el bloque 1084, como respuesta a una consulta tanto los objetos de archivo como los objetos diferentes a los archivos se obtienen para coincidir con la búsqueda. En el bloque 1086 tanto los tanto los objetos de archivo como
15 los objetos diferentes a los archivos se presentan el despliegue.

La FIGURA 40 es un diagrama ilustrativo de una representación de pantalla en la cual se muestran diferentes objetos diferentes a archivos. Como se muestra en la FIGURA 40, los objetos han sido filtrados para incluir aquellos que incluyen la expresión "John". Los objetos se muestran para incluir el objeto
20 de contacto 1101, el objeto de correo electrónico 1102 y los objetos de documentos 1103 y 1104. El objeto de contactos 1101 y el objeto de correo electrónico 1102 son objetos diferentes a archivos. Este sistema permite que

se incluyan objetos diferentes a archivos junto con objetos comunes de archivos, en tal forma que puedan ser organizados y manipulados en la forma deseada por el usuario. Como se describió anteriormente con respecto ala FIGURA 2, tales objetos diferentes a los archivos pueden estar totalmente
5 contenidos dentro de la base de datos relacional 230, la cual además incluye información referente a las propiedades de los archivos.

Las FIGURAS 41-50 son diagramas relacionados con una barra de direcciones virtuales que corresponde al a información de la línea 641 de la FIGURA 10, y la cual se forma de acuerdo con la presente invención. Como se
10 describirá en mayor detalle a continuación, la barra de direcciones virtuales comprende una pluralidad de segmentos, y cada segmento corresponde a un filtro para seleccionar contenido. Colectivamente, los filtros correspondientes de cada segmento representan una dirección virtual para seleccionar contenido.

La FIGURA 41 es un diagrama de bloques de ambiente de computación
15 ejemplar en red 1200 apropiado para operar la barra de direcciones virtuales, u otro aspecto cualquiera de la presente invención aquí descrito. El ambiente de computación ejemplar en red 1200 incluye un dispositivo computacional 1202, tal como el computador personal descrito en relación con la FIGURA 1, para interactuar con el usuario, y sobre el cual el usuario puede visualizar archivos
20 almacenados ya sea localmente o remotamente en el dispositivo computacional. Mientras que la siguiente discusión describe la presente invención en relación con un computador personal, se debe entender que el

dispositivo computacional 1202 incluye muchos tipos de dispositivos físicos, incluyendo pero sin limitarse a mini computadores o computadores "mainframes", PDAs, computadores de tarjeta, y otros dispositivos capaces de interactuar con un usuario y de desplegar archivos y contenidos almacenados en el dispositivo computacional y en otros lugares..

El ambiente de computación ejemplar en red 1200 puede también incluir uno o más servidores remotos, tales como el servidor 1204, que guarda archivos a los cuales el dispositivo 1202 puede tener acceso, y que están conectados con el dispositivo computacional por medio de una red de comunicaciones, tales como la Internet 1206, como se muestra en la FIGURA 41. Adicionalmente, el dispositivo computacional 1202 también puede estar conectado a otras fuentes de información que guardan archivos u otros contenidos, tales como la base de datos remota 1208. Aquellos versados en la técnica reconocerán que tanto los archivos como la información almacenada en el servidor remoto 1204 y en la base de datos remota 1208, así como en dispositivos locales de almacenamiento tales como unidades de disco duro (FIGURA 1), pueden ser accedidos desde y desplegados en, el dispositivo computacional 1202, como parte de un sistema integrado de archivos en el dispositivo computación al. Adicionalmente, mientras una configuración particular de un servidor remoto 1204 y una base de datos remota 1208 se presenta en la FIGURA 41, aquellos versados en la técnica reconocerán fácilmente que esta configuración particular se presenta con fines ilustrativos

únicamente, y no se debe interpretar como una limitación sobre la presente invención.

La FIGURA 42 ilustra un visor de archivos ejemplar 1300 con una barra de direcciones convencional 1302 asociada al despliegue de archivos en un sistema de archivos de computador, como se encuentra en el estado anterior de la técnica. Para fines de la presente discusión, un visor de archivos puede ser una vista dentro de una caja de diálogo abierta o cerrada en un programa ejecutable que debe guardar o recuperar datos en o desde un dispositivo de almacenamiento conectado localmente o remotamente al sistema computacional. Se debe notar que los ejemplos anteriores de un visor de archivos son ilustrativos, y no se deben interpretar como limitaciones a la presente invención.

Una dirección sobre la barra de direcciones convencional 1302 corresponde a una localización específica dentro de un sistema de archivos. Como se describió anteriormente, con el fin de editar las direcciones desplegadas en la barra de direcciones convencional 1302, un usuario debe modificar la dirección de acuerdo con el conocimiento específico del sistema de archivos. Alternativamente, un usuario debe seleccionar una entrada en una vista de árbol 1304 para navegar hacia una localización alternativa. Aquellos versados en el estado de la técnica reconocerán que otros controles externos a la barra de direcciones 1302 también pueden estar disponibles, y que no se muestran en el visor de archivos ejemplar 1300. Mientras que la dirección

desplegada en la barra convencional de direcciones 1302 corresponde a una localización específica en un sistema de archivos, los archivos relacionados distribuidos entre carpetas múltiples en el sistema de archivos no se podrán desplegar en conjunto con la barra de direcciones convencionales 1302.

5 La FIGURA 43 ilustra in visor ejemplar de archivos 1400 con una barra de direcciones virtuales 1402 asociada con el despliegue de archivos en un sistema de archivos de computador. La barra de direcciones virtuales 1402 corresponde a la línea de información 641 dela FIGURA 10. La barra de direcciones virtuales 1402, con una dirección virtual 1404, está configurada
10 para desplegar información similar a aquella desplegada por la barra convencional de direcciones 1304 del estado de la técnica anterior en el visor de archivos 1300 de la FIGURA 42. Una dirección virtual, a la cual también se le llama un trayecto virtual ("*virtual path*") hace referencia a archivos almacenados en un sistema computacional de archivos de acuerdo con el
15 criterio de selección.

En forma similar a una dirección convencional, tal como la dirección 1304 de la FIGURA 42, el criterio de selección de la dirección virtual puede referenciar archivos almacenados en una localización específica dentro de la jerarquía del sistema. Sin embargo, en contraste con una dirección
20 convencional, el criterio de selección de la dirección virtual también puede referenciar archivos independientemente de su localización en un sistema de archivos. Así, una dirección virtual puede referenciar archivos almacenados en

localizaciones múltiples de un sistema computacional de archivos. Como se muestra en la FIGURA 43, el visor de archivos 1400, de acuerdo con la dirección virtual 1404 en la barra de direcciones virtuales 1402 es capaz de desplegar archivos adicionales, tales como los archivos 1406 y 1408, que no se encuentran en el visor de archivos 1300 del FIGURA 42. Adicionalmente, la barra de direcciones virtuales 1402 también puede utilizarse para desplegar otros contenidos diferentes a los archivos en un sistemas de archivos de computador. Por ejemplo, la barra de direcciones virtuales 1402 podrá usarse para referenciar contenidos incluyendo dispositivos del sistema, servicios del sistema, o localizaciones en Internet.

La FIGURA 44A ilustra la manipulación de un segmento de la dirección virtual 1404 en la barra de direcciones virtuales 1402 con el fin de navegar a través de un sistema de archivos de computador. Cada barra de direcciones virtuales, tal como la barra de direcciones virtuales 1402, está comprendida por uno o más segmentos interactivos, tales como los segmentos 1502, 1504, 1506 y 1508. Cada segmento dentro de una barra de direcciones virtuales corresponde a un filtro predeterminado, o a un criterio de selección, en todos los contenidos disponibles o archivos a los cuales se pueden tener acceso desde un sistema de archivos de computador. Colectivamente, los filtros de todos los segmentos en una barra de direcciones virtuales 1402 representa la dirección virtual de la barra de direcciones virtuales.

El primer segmento en una barra de direcciones virtuales, tales como el segmento 1502, se llama segmento raíz o filtro raíz. El segmento raíz representa la categoría más amplia de contenido disponible para la selección de una barra de direcciones virtuales 1402. Por ejemplo, el segmento 1502 de

5 "Archivos" podría representar muy probablemente un filtro que referencia todos los archivos a los cuales se puede tener acceso por parte del sistema de archivos de computador. Alternativamente, un segmento raíz puede representar un filtro que hace referencia a todos los servicios del sistema disponibles en el sistema de computador, o a un filtro que hace referencia a

10 todos los dispositivos físicos (*hardware*) instalados en el sistema computacional. Aquellos versados en la materia reconocerán que en la presente invención se pueden utilizar numerosos filtros raíces diferentes. Así los ejemplos arriba descritos se dan solamente con fines ilustrativos, y no se deben interpretar como una limitación a la presente invención. Adicionalmente, las etiquetas

15 desplegadas para cada segmento, tales como "Archivos" en el segmento raíz 1502 son ilustrativas y no se deben interpretar como una limitación a la presente invención. De acuerdo con la modalidad, una etiqueta desplegada en un segmento es configurable por el usuario.

Cada segmento adicional en una barra de direcciones virtual 1402, tal

20 como los segmentos 1504, 1506 y 1508 representan filtros adicionales para ser aplicados durante la selección y el despliegue de archivos o de contenidos en un visor de archivos 1400. Por ejemplo, el segmento raíz 1502 "Archivos"

referencia todos los archivos disponibles para el sistema computacional. El segmento 1504 "Librería de Documentos" filtra los archivos seleccionados por el segmento raíz 1502, seleccionando aquellos archivos que se generaron como documentos por parte del usuario, tales como con procesador de palabra, hoja
5 de cálculo o por otra aplicación generadora de documentos. El segmento 1506 "Documentos de Word" filtra los archivos seleccionados por el segmento 1504 de acuerdo con aquellos documentos que fueron generados por un procesador de palabra, tal como la aplicación Word de Microsoft Corporation. Finalmente, el segmento 1508 "Autor A" filtra los documentos de procesamiento de palabra
10 seleccionados por el segmento 1506 de acuerdo al hecho si fueron creados por el "Autor A". Así. El contenido seleccionado de acuerdo con la dirección virtual representada en la barra de direcciones virtuales 1402 debe satisfacer los filtros correspondientes a todos los segmentos que se encuentran en la barra de direcciones virtuales.

15 Los segmentos en la barra de direcciones virtuales 1402 generalmente están ordenados desde aquellos filtros que son más inclusivos, hasta por aquellos filtros que son menos inclusivos. Por ejemplo, como se discutió anteriormente, el segmento 1502 "Archivos" es el más amplio y más inclusivo. Los segmentos 1506 "Documentos de Word" y el segmento 1508 "Autor A" son
20 menos inclusivos. La barra de direcciones virtuales 1402 ilustra el ordenamiento de los segmentos de izquierda a derecha y, para fines de la presente discusión, los segmentos 1504, 1506, y 1508 son subsiguientes al

segmento raíz 1502. Sin embargo, se debe entender que hay otras orientaciones posibles, tale como el arreglo de arriba hacia abajo, sin alejarse del alcance de la invención. Así, la orientación de izquierda a derecha se debe ver como ilustrativa, y no se debe interpretar como una limitante a la presente
5 invención.

Como se mencionó anteriormente, los segmentos en una barra de direcciones virtuales 1402, tal como los segmentos 1502, 1504, 1506 y 1508 no corresponden necesariamente a localizaciones específicas dentro de un sistema de archivos de computador, tal como carpetas, unidades y directorios.
10 Así, el segmento 1504 "Librería de Documentos" puede referenciar archivos o contenidos distribuidos en múltiples servidores, unidades, o carpetas / directorios. Sin embargo, algunos segmentos en una barra de direcciones virtuales 1402 pueden referenciar localizaciones específicas dentro de la jerarquía del sistema de archivos de computador. A continuación se presenta
15 una discusión adicional de los segmentos de las direcciones virtuales referenciando localizaciones específicas en u sistema de archivos, en relación con las FIGURAS 48A y 48B.

En contraste con la barra de direcciones convencional, cada segmento en la barra de direcciones virtual 1402 representa un elemento accionable,
20 interactivo de la interface de usuario. Por ejemplo, un segmento en la barra de direcciones virtuales 1402 responde a una selección por parte del usuario, supervisa si el cursor está localizado sobre el segmento durante un periodo

específico de tiempo, y puede ser retirado de la barra de direcciones virtuales mediante una interacción de arrastre por parte del usuario. Así, como se muestra en la FIGURA 44A, un usuario puede colocar un cursor 1510 sobre un segmento en la barra de direcciones virtuales 1402, tal como el segmento 1504

5 "Librería de Documentos" para seleccionar dicho segmento, con el fin de navegar hacia ese nivel, i.e. truncar la dirección virtual en ese segmento, como se describe en relación con la FIGURA 44B.

La FIGURA 44B ilustra los resultados de la selección del segmento 1504 en la barra de direcciones virtuales 1402. Mediante la selección del segmento

10 1504 en la barra de direcciones virtuales 1402, el usuario indica su deseo de navegar hacia ese nivel en las direcciones virtuales. En efecto, el usuario está recortando aquellos filtros subsiguientes al segmento seleccionado. Por ejemplo, mediante la selección sobre el segmento 1504 "Librería de Documentos" (FIGURA 44A), la dirección virtual resultante 1404 ya no contiene

15 los segmentos 1506 "Documentos de Word" ni 1508 "Autor A" (FIGURA 44A). Adicionalmente, por el hecho de que el usuario navegó hacia un conjunto menos restrictivo de filtros, la dirección virtual resultante 1404 en la barra de direcciones virtuales 1402 es más incluyente. Esto se indica mediante la adición de documentos en el visor de archivos 1400 de la FIGURA 44B que no

20 se encontraban anteriormente en el visor de documentos 1400 de la FIGURA 44A, incluyendo el documento 1512, el documento 1514, y el documento 1516, y por la presencia de un botón de recorrido (*scroll button*) 1518 que indica que

archivos adicionales que pueden visualizarse no pueden ser desplegados en el visor de archivos 1400 (FIGURA 44B) debido a limitaciones de espacio.

Adicionalmente a la selección de segmentos en una barra de direcciones virtuales para navegar hacia un segmentos menos restrictivo, un usuario puede también desear navegar hacia o seleccionar filtros compañeros de los segmentos actuales en la dirección virtual. Un filtro compañero es un filtro alternativo que se puede seleccionar aplicado a un segmento dado en la barra de direcciones virtuales. Por ejemplo, referente ala FIGURA 44A, los filtros compañeros para el segmento 1506 "Documento de Word" pueden incluir filtros tales como "Documentos de Excel", "Diarios" y similares. Otros tipos de filtros, incluyendo localizaciones específicas de archivos, dispositivos físicos (*hardware*), o servicios computacionales, también pueden aplicarse a un segmento dado en la barra de direcciones virtuales. Los filtros compañeros pueden estar o no relacionados lógicamente con un filtro particular de un segmento. Cada segmento en una barra de direcciones virtuales puede tener filtros compañeros. La selección de un filtro compañero en una barra de direcciones virtuales algunas veces se llama navegación lateral. A continuación se describe la selección de filtros compañeros en una barra de direcciones virtuales en relación con las FIGURAS 45A-45D, y también en relación con la FIGURA 49.

Las FIGURAS 45A-45D son diagramas pictóricos que ilustran la selección de filtros compañeros asociados con un segmento d direcciones

virtuales en una barra de direcciones virtuales 1600. Como se muestra en la FIGURA 45A, la barra de direcciones virtuales 1600 tiene una dirección virtual que comprende múltiples segmentos, los segmentos 1602-1608. Con el fin de seleccionar un filtro compañero para un segmento interactivo dado en una barra

5 de direcciones virtuales 1600, un usuario debe hacer una selección alternativa, o una manipulación alternativa, de ese segmento interactivo. Una forma de hacer una selección interactiva es hacer la selección con el botón derecho sobre un segmento dado. En la técnica la selección con el botón derecho se conoce y se refiere al botón secundario en el ratón, o en otro dispositivo d

10 entrada, en el cual el botón secundario está típicamente al lado derecho del ratón. Alternativamente, dado que un segmento interactivo puede supervisar cuándo el cursor está localizado sobre él, la selección alternativa se puede hacer colocando el cursor sobre el segmento interactivo y dejando el cursor en ese sitio durante un lapso predeterminado de tiempo, algunas veces llamado

15 rondar ("*hovering*"). Sin embargo, mientras que esta discusión describe alternativas para causar el despliegue de filtros compañeros, éstas son solamente ilustrativas, y no se deben interpretar como limitaciones a la presente invención. Aquellos versados en la técnica reconocerán que existe muchas alternativas para generar la selección alterna.

20 Para ilustrar la selección alterna de un segmento, con referencia a la FIGURA 45A, el usuario inicialmente coloca el cursor 1610 sobre el segmento 1604 "Librería de Documentos" durante un periodo determinado de tiempo, i.e

"ronda" sobre el segmento, para seleccionar dicho segmento. La FIGURA 45B demuestra los resultados de alternativamente seleccionar el segmento 1604 "Librería de Documentos", una vista de filtros compañeros 1612 se despliega incluyendo los filtros compañeros que corresponden a segmento seleccionado.

- 5 Se debe entender que los filtros compañeros presentados en la vista de filtros compañeros 1612 son solamente ilustrativos, y no deben interpretarse como limitantes a la presente invención.

- Con el fin de seleccionar in filtro compañero alternativo o alerno, como se muestra en la FIGURA 45C, el usuario coloca el cursor 1610 sobre uno de
- 10 los filtros presentados en la vista de filtros compañeros 1612, tal como el filtro compañero 1614, y selecciona dicho filtro compañero. Como se muestra en la FIGURA 45D, después de seleccionar el filtro compañero alerno 1614, segmento 1604 previamente seleccionado (FIGURA 45A) se reemplaza con un nuevo segmento 1616 que representa el filtro compañero seleccionado 1614.
- 15 Adicionalmente, aquellos segmentos siguientes al segmento alerno seleccionado 1604 en la barra de direcciones virtuales 1600 de la FIGURA 45A, específicamente los segmentos 1606 "Diarios" y 1608 "Todos los Documentos en el 2002", son retirados de la barra de direcciones virtuales 1600 en la FIGURA 45D. Aunque no se muestran, sigue que cualesquiera archivos o
- 20 contenidos seleccionados previamente de acuerdo con los segmentos 1604 "Librería de Documentos", 1606 "Diarios" y 1608 "Todos los Documentos en el 2002" no se mostrarían en el correspondiente visor de archivos, y solamente se

mostrarían aquellos archivos o contenidos seleccionados de acuerdo con los segmentos 1602 "Archivos" y 1616 "Librería de Fotos".

Los segmentos se pueden agregar a una dirección virtual en una barra de direcciones virtuales a través de varias interacciones del usuario al final de los segmentos existentes. Para agregar un filtro a una dirección virtual en una barra de direcciones virtuales, el usuario puede manipular un control accionable 1412 "2003" para adicionar un filtro correspondiente a la dirección virtual 1404 en la barra de direcciones virtuales 1402. Alternativamente (no se muestra), un usuario puede manualmente entrar un filtro conocido al final de la dirección virtual digitando el nombre del filtro. Existen muchas otras formas de agregar un filtro a una dirección virtual, todas las cuales se consideran abarcadas dentro del alcance de la presente invención. Así, debe entenderse que los ejemplos anteriores son solamente ilustrativos y no se deben entender como limitaciones a la presente invención.

Cuando se agrega un filtro a una dirección virtual en una barra de direcciones virtuales, se inicia un proceso para asegurar que el filtro recién agregado no esté en conflicto con ninguno de los filtros existentes como parte de la dirección virtual. El filtro recién agregado está en conflicto con alguno de los filtros existentes en una de las direcciones virtuales o si el filtro recién agregado presenta variaciones de alcance en relación con el filtro existente, ya sea que presente mayor o menor amplitud que el filtro existente. Adicionalmente, un filtro recién agregado entra en conflicto con un filtro

existente si el filtro recién agregado es mutuamente excluyente respecto del filtro existente. Sin embargo, un filtro recién agregado que es equivalente a un filtro existente no será agregado por que no tendrá efecto alguno. Debe entenderse que la discusión anterior se presenta para fines ilustrativos, y no debe interpretarse como limitación a la presente invención. Aquellos versados en el estado de la técnica reconocerán que pueden existir otros conflictos entre filtros, y que se consideran abarcados dentro del alcance de la presente invención.

Las FIGURAS 46A-46D son diagramas pictóricos que ilustran la adición de filtros a una dirección virtual 1702 en una barra de direcciones virtuales 1700, retirando filtros existentes con conflictos. La FIGURA 46A ilustra un ejemplo de una dirección virtual 1702 desplegada en una barra de direcciones virtuales 1700. Como se muestra en la FIGURA 46B, un nuevo filtro representado por el segmento 1706 "2002", se agrega a la dirección virtual 1702. Como se describió anteriormente, los filtros nuevos se adicionan al final de las direcciones virtuales, como se indica colocando el segmento 1706 "2002" al final de los segmentos en la barra de direcciones virtuales 1700 de la FIGURA 46B. En adelante, el proceso que se inicia para agregar este nuevo segmento 1706 "2002" determina que el filtro agregado no está en conflicto con ninguno de los filtros presentes en la dirección virtual 1702. De esta manera, no se retira ninguno de los filtros existentes de la dirección virtual 1702.

Como se muestra en la FIGURA 46C, se agrega otro filtro a la dirección virtual 1702, representado por el segmento "Autor A" 1708. El proceso que se inicia para agregar este nuevo segmento determina que el nuevo filtro "Autor A" estaría en conflicto con el filtro representado por el segmento 1704 "Autor A-F" dado que el filtro "Autor A" es más restrictivo que el filtro existente. De conformidad el segmento 1704 "Autor A-F" se retira de la barra de direcciones virtuales 1700 y el segmento 1708 "Autor A" se agrega al final de los segmentos de la barra de direcciones virtuales.

La FIGURA 46D ilustra los resultados de agregar un segmento 1710 "2003" a la barra de direcciones virtuales 1700 de la FIGURA 46C. Los filtros en una dirección virtual 1702 son restrictivos, no acumulativos. Cada filtro restringe adicionalmente el contenido seleccionado. Así filtros mutuamente excluyentes evitarían que la dirección virtual 1702 seleccionara archivos o contenido alguno, y por ende generaría un conflicto. Como lo ilustra la FIGURA 46D, el segmento 1706 "2002" (FIGURA 46C) se retira de la barra de direcciones virtuales 1700 debido a un conflicto, tal que es mutuamente excluyente con el segmento 1710 "203" recién agregado.

Cuando una barra de direcciones virtuales, tal como una barra de direcciones virtuales 1800 (FIGURA 47A) no puede desplegar completamente la dirección virtual debido a limitaciones de tamaño de la barra de direcciones virtuales, se despliega una porción de la dirección virtual de acuerdo con el tamaño de la barra de direcciones virtuales. Sin embargo, el usuario aún puede

tener acceso a las porciones no mostradas de la dirección virtual. Más específicamente, la barra de direcciones virtuales despliega indicadores visuales accionables para recorrer el trayecto virtual dentro de la barra de direcciones virtuales. Las FIGURAS 47A y 47B ilustran un ejemplo de barra de direcciones virtuales 1800 desplegando una dirección virtual en la cual la dirección virtual excede la capacidad de despliegue de la barra de direcciones virtuales. Como se muestra en las FIGURAS 47A y 47B, los íconos de seguimiento 1802 y 1804 indican la dirección de la barra de direcciones virtuales 1800 en la cual se puede hacer el seguimiento con el fin de mostrar la porción de las direcciones virtuales no mostrada. Sin embargo, mientras los diagramas ilustrativos demuestran el uso de los íconos de seguimiento, solo se hace con fines ilustrativos, y no se debe interpretar como una limitante de la presente invención. Aquellos versados en el estado de la técnica reconocerán que existen muchas otras formas de hacerle seguimiento a las direcciones virtuales en la barra de direcciones virtuales, todas las cuales se consideran abarcadas dentro del alcance de la presente invención.

La FIGURA 48A es un diagrama de bloques que ilustra una barra de direcciones virtuales 1900 con segmentos que hacen referencia tanto a las localizaciones virtuales como a las localizaciones reales dentro del sistema de archivos. Como se discutió anteriormente, una dirección virtual en una barra de direcciones virtuales 1900 puede contener segmentos que hacen referencia a localizaciones específicas dentro del jerarquía de un sistema de archivos de un

computador, y también contienen segmentos que hacen referencia a localizaciones virtuales, o lógicas, dentro del sistema de archivos de un computador. Los archivos o contenidos referenciados por un segmento virtual pueden distribuirse entre varias localizaciones físicas. Un barra de direcciones virtuales 1900 puede contener segmentos que hacen referencia a localizaciones físicas y segmentos que hacen referencia a localizaciones virtuales. Por ejemplo, la barra de direcciones virtuales 1900 incluye el segmento 1902 "Disco Local (C:)" refiriéndose a archivos o a contenidos en un área específica de un sistema de archivos de computador, en particular la unidad "C".

Alternativamente, el segmento 1904 "Archivos de Caso" de sí misma se refiere a archivos o a contenidos almacenados en carpetas múltiples en la jerarquía del sistema de archivos de computador asociados con dichos archivos de caso. Sin embargo, en combinación con el segmento 1902 "Disco Local (C:)", el segmento 1904 "Archivos de Caso" hace referencia solamente a aquellos archivos que se encuentran en la unidad local "C". Adicionalmente, el segmento 1906 "Contiene Fax" además filtra los archivos en el disco local C: y asociados con los archivos de caso, dependiendo si contienen la palabra "Fax".

Como se muestra en la FIGURA 48B, una barra de direcciones virtuales 1900 se puede configurar para que funcione como un barra de direcciones convencional. Por ejemplo, en relación con la FIGURA 48A, mediante la colocación del cursor 1908 en el espacio vacío de la barra de direcciones virtuales 1900 y seleccionándola, la barra de direcciones virtuales 1900 cambia

de mostrar segmentos que representan una dirección virtual, y funciona como una barra de direcciones convencional que muestra una dirección convencional 1910 como se muestra en la FIGURA 48B. La dirección convencional 1910 en la barra de direcciones convencional 1900 de la FIGURA 48B se aproxima la
5 dirección virtual mostrada en la barra de direcciones virtuales 1900 de la FIGURA 48A. Sin embargo, aquellos filtros en la barra de direcciones virtuales 1900 de la FIGURA 48A que no corresponden a localizaciones físicas en un sistema de archivos de computador no pueden ser mostradas, y son retiradas de la barra de direcciones convencionales 1910. Específicamente el segmento
10 1904 "Archivos de Caso" y el segmento 1906 "Contiene Fax" no forman parte de la dirección convencional 1910 (FIGURA 48B).

Con el fin de reconfigurar la barra de direcciones virtuales 1900, que funcione como una barra de direcciones convencionales, para que funcione como una barra de direcciones virtuales, el usuario debe indicarlo en una forma
15 diferente a seleccionar el espacio vacío en la barra. Cuando se configura para funcionar como una barra de direcciones convencionales, una barra de direcciones virtuales debe permitirle al usuario seleccionar el espacio vacío para fines de editar direcciones. La selección sobre el espacio vacío de una barra de direcciones convencionales coloca un cursos de edición al final de la dirección /
20 trayectoria para fines de edición. De conformidad, para reconfigurar que la barra de direcciones funcione nuevamente en la forma normal arriba descrita, el usuario debe accionar una tecla determinada o una secuencia de teclas, tal

como la tecla "Esc" o "Tab", o colocando el foco en otra área o ventana mediante la selección de otra área en la ventana o vista. Aquellos versados en la técnica reconocerán que existen muchas otras acciones que pueden también utilizarse para reconfigurar la barra de direcciones virtuales 1900 para que
5 retorne a funcionar en su modo normal como se describe arriba, todas las cuales se consideran abarcadas dentro del alcance de la presente invención.

La FIGURA 49 es un diagrama de flujo que ilustra la rutina de selección de filtros compañeros 2000 para seleccionar un filtro compañero para un segmento identificado en una barra de direcciones virtuales. Iniciando con el
10 bloque 2002, la rutina 2000 detecta la activación de la selección de un filtro compañero. El proceso de activación de la selección de un filtro compañero se describe arriba en relación con las figuras 45A-45D. En el bloque 2004, se identifica el segmento para el cual se ha solicitado la selección de un filtro compañero. En el bloque 2006 se determinan los filtros compañeros a partir de
15 una lista predeterminada de filtros compañeros. En el bloque 2008 los filtros compañeros se muestran al usuario. En el Bloque 2010 se obtiene la selección de filtros compañeros por parte del usuario de la lista de filtros compañeros mostrada. En el bloque 2012, se trunca la dirección virtual mediante el retiro del segmento identificado de la barra de direcciones virtuales, y cualesquiera
20 segmentos adicionales a continuación de dicho segmento identificado. En el bloque 2014, se adiciona un segmento que representa el filtro compañero

seleccionado, a los demás segmentos que quedan en la barra de direcciones virtuales. De ahí en adelante, la rutina 2000 finaliza.

La FIGURA 50 es un diagrama de flujo que ilustra el ejemplo de una rutina 2100 de adición de filtros para agregar un filtro a una dirección virtual en una barra de direcciones virtuales. Iniciando en el bloque 2102, la rutina 5 ejemplar 2100 obtiene el filtro a ser agregado a la dirección virtual. Por ejemplo, como se discutió anteriormente en relación con la FIGURA 43, los filtros pueden ser agregados a la dirección virtual de acuerdo con acciones externas del usuario sobre la barra de direcciones virtuales, o alternativamente, se pueden 10 agregar directamente a la barra de direcciones virtuales digitando el nombre de un filtro predeterminado.

En el bloque 2014, se determina si el nuevo filtro está en conflicto con un filtro existente en las direcciones virtuales. Como se discutió anteriormente en relación con las FIGURAS 46A-46D, un nuevo filtro puede estar en conflicto con 15 un filtro existente mediante una ampliación o reducción sustancial del alcance de dicho filtro existente. Alternativamente, un nuevo filtro puede estar en conflicto con un filtro existente si un filtro es mutuamente exclusivo de un filtro existente. Si, en el bloque de decisión 2014 el nuevo filtro está en conflicto con un filtro existente en el bloque 2016 el filtro existente se retira de la dirección 20 virtual. Alternativamente, si en 2014 el nuevo filtro no está en conflicto con un filtro existente o, si después de retirar el filtro en conflicto en el bloque 2016, en

el bloque 2018 el nuevo filtro se agrega al final de la dirección virtual. De ahí en adelante la rutina 2100 termina.

Las FIGURAS 51-57 son diagramas relacionados con un sistema y un método de acuerdo con otro aspecto de la invención, que suministra un experiencia de usuario mejorada dentro de un navegador de la *shell*. Más específicamente, se proporcionan el sistema y método por el cual los usuarios pueden más fácilmente identificar un objeto con base en los metadatos asociados a dicho objeto.

En relación con la FIGURA 51A, una ventana 2200 representa un despliegue del tamaño de la pantalla para la interface gráfica de un navegador de la *shell*. La ventana 2200 contiene un área de 2202 y un área de visualización 2204. El área 2202 puede incluir in control de previsualización 2206, una interface de usuario (UI) o un control de edición 2208 y un control de actividades 2210. Típicamente, el control de previsualización 2206 suministrará al usuario controles tales como botones de iteración que le permiten al usuario cambiar el foco de un objeto al siguiente mediante la acción del botón del ratón. Los metadatos correspondientes a uno o mas objetos y /o metadatos correspondientes al objeto contenedor pueden ser desplegados en una variedad de localizaciones dentro de la ventana 2200. Por ejemplo, el control de edición y los metadatos pueden ser relocalizados dentro del área de control de edición 2208 de tal forma que el área de control de edición no solo incluya un despliegue de las propiedades clave del objeto previsualizado, sino que también

le muestre al usuario la opción de realizar ediciones a los metadatos. El control de actividades 2210 contiene actividades relevantes al nombre del espacio y /o al selección.

Para fines de la presente invención, los términos "metadatos" y
5 "metadatos modificables por el usuario" excluyen el nombre del objeto de la *shell*. El término "nombre del objeto de la *shell*" se refiere a la propiedad que se usa para fines de escoger y mostrar el objeto dentro del navegador de la *shell* ,
Como se mencionó arriba, un único aspecto de la presente invención consiste en la habilidad del usuario de poder editar metadatos dentro del navegador de
10 la *shell* .

Aquellos versados en la técnica apreciarán que la presente invención contempla la presencia de características opcionales dentro de la ventana 2200. Por ejemplo, el control de previsualización 2206 y el control de actividades 2210 no son características esenciales para los fines de la presente invención. Más
15 aún, otras características no esenciales que no se muestran en la FIGURA 51A, tales como la barra de herramientas que incluye botones de iteración o un botón de mostrar / esconder para que el usuario pueda cerrar / abrir la ventana de previsualización, se encuentran dentro del alcance de la presente invención. No obstante, estas y otras características opcionales pueden ayudarle al
20 usuario a localizar más fácilmente un objeto particular dentro del navegador de la *shell* .

La vista 2204 suministra un vista de la lista de uno o más objetos 2212, tales como archivos o carpetas del sistema. El término "vista de lista" se refiere a la enumeración o lista dentro de un contenedor. Los términos "objeto" y "objeto de la shell", se usan indistintamente aquí, para hacer referencia

5 archivos, carpetas y otros contenedores, y otros objetos diferentes a archivos que se pueden representar en una vista de la lista. Ejemplos de objetos diferentes a archivos pueden incluir sin limitarse a, contactos, favoritos y mensajes de correo electrónico. Los términos "navegador de la shell " y "navegador de archivos del sistema" se usan indistintamente aquí para referirse

10 a un navegador que permite la navegación a través de varios nombres de espacios incluyendo archivos y otros objetos diferentes a archivos.

Aquellos versados en la técnica apreciarán que la presente invención contempla muchos diseños posibles y representaciones para la ventana 2200. Por ejemplo, la ventana de previsualización 2202 se muestra encima del área

15 de visualización 2204 en la FIGURA 51A. No obstante, otras representaciones, tales como colocar la ventana de previsualización 2202 y el área de visualización 2204, lado a lado, se encuentran claramente dentro del alcance de la presente invención. La localización del control de edición 2208 también es independiente de la localización de otros controles. También existen muchas

20 otros tipos de posibles vistas para los objetos mostrados en el áreas de lista 2204, tales como detalles, presentación de diapositivas, sección de película, imagen tamaño pulgar, capas, íconos, etc.

La FIGURA 51B es similar a la FIGURA 51A, excepto en el hecho de que el área de visualización 2204 se reemplaza por un área de visualización 2214 que despliega los objetos 2212 en un modo detallado. Como es típico para los objetos de la *shell* mostrados en el modo detallado, los objetos 2212 están

5 alineados en una columna al lado izquierdo del área de la vista 2214, y uno o más encabezados de columna 2216 forman la fila superior de un conjunto de columnas que contienen metadatos 2218 referentes al objeto correspondiente localizado en la misma fila. Es de importancia, que la presente invención contempla proporcionar al usuario la capacidad de cambiar los valores de los

10 metadatos a valores diferentes en forma instantánea mediante uno o más controles de edición 2208 en localizados en cualquier lugar de la ventana 2200. Por ejemplo, un control de edición puede ser suministrado en la ventana de previsualización 2202 y/o dentro del área de la vista 2214. Por ejemplo, se puede suministrar un control de edición que no sea inicialmente visible para el

15 usuario dentro del área de la vista 2214. Dicho control puede ser activado instantáneamente, por ejemplo, cuando el usuario "ronda" sobre los metadatos 2218, y luego selecciona sobre él para entrar al modo de edición.

A continuación haciendo referencia a la FIGURA 52, se suministra una ilustración esquemática de una vista de bienvenida 2300 dentro del navegador

20 de la *shell*. Una vista de bienvenida a veces se llama una ventana de "selección nula" porque representan un nombre de espacio o un contenedor, a diferencia de proporcionar la posibilidad de una selección. Si el usuario aún no

ha realizado una selección, una ventana de previsualización 2302 despliega los metadatos 2304 y las tareas clave referentes a la carpeta o a la librería de la *shell*. Si se desea, las tareas pueden separarse en tareas principales 2306 o otras tareas relevantes 2308. La ventana de bienvenida 2300 también incluye

5 un área de visualización 2310, en la cual archivos múltiples y otros objetos 2312 se pueden revisar. Los metadatos en la ventana de bienvenida 2304 pueden incluir información tal como muestreo de metadatos de cada uno de los objetos dentro del contenedor, en cuyo caso el despliegue de metadatos puede cambiar con frecuencia. Por ejemplo, el despliegue de metadatos puede limitarse a las

10 propiedades de un objeto a la vez, mediante la alternación de un objeto al siguiente cíclicamente cada 30 segundos.

La FIGURA 53 es una ilustración esquemática de una ventana seleccionada 2400 dentro del navegador de la *shell*. A diferencia de la ventana de bienvenida, la ventana seleccionada representa una selección por parte del

15 usuario. Si el usuario selecciona un contenedor o una carpeta, la ventana seleccionada no necesita ser idéntica a la ventana de bienvenida para ese contenedor o carpeta. En la FIGURA 53, la ventana seleccionada 2400 incluye una ventana de previsualización 2402 que contiene un control de previsualización 2404, un despliegue de metadatos 2406 y un despliegue de

20 tareas 2408. Similar a la ventana de bienvenida, 2300 (en la FIGURA 52), la ventana de selección 2400 también incluye un área de visualización 2410, en la cual múltiples archivos u otros objetos 2412 pueden ser visualizados. En la

FIGURA 53, sin embargo, el usuario ha seleccionado uno de los archivos. En consecuencia, el control de previsualización 2404 despliega una imagen de previsualización del archivo seleccionados, el despliegue de los metadatos 2406 muestra las propiedades del archivo seleccionado, y el despliegue de tareas 2408 proporciona un menú de tareas relevantes para la operación del archivo seleccionado. La FIGURA 54 es una representación esquemática de la ventana seleccionada de la FIGURA 53, pero que además incluye un menú de contexto 2500 para permitirle a un usuario modificar los metadatos en un navegador de la *shell* de acuerdo con la modalidad de la presente invención. El menú de contexto 2500 en la FIGURA 54 le presenta al usuario varias opciones para cambiar los metadatos seleccionados. El texto genérico que se muestra en el menú 2500 es por supuesto solamente un ejemplo del tipo de opciones que se le pueden ofrecer a un usuario para editar los metadatos desplegados. Se puede suministrar un menú de contexto en cualquier ventana, incluyendo una ventana de bienvenida, para mejorar la experiencia del usuario. Como lo podrán apreciar aquellos versados en la técnica, se puede dar soporte a cualquier número y variedad de menús de contexto dentro de esta invención. Para fines de la presente invención, uno de los medios para permitirle al usuario modificar los metadatos desplegados dentro de un navegador de la *shell*, es suministrar un menú de contexto, por ejemplo, mediante la selección del texto correspondiente o del objeto en la ventana de previsualización.

Aquellos versados en la técnica apreciarán que la presente invención contempla medios diferentes a menús de contexto para permitirle al usuario modificar los metadatos desplegados dentro de un navegador de la *shell*. Otro medio similar es que el usuario seleccione los metadatos para entrar en un modo de edición. En contraste un usuario podría entrar el modo de edición "rondando" sobre el texto u objeto relevante en la ventana de previsualización. Existen numerosos medios alternos disponibles y que se encuentran dentro del alcance de la presente invención.

La FIGURA 55 es un diagrama de flujo que ilustra un método 2600 para permitirle al usuario modificar los metadatos desplegados en una ventana de bienvenida dentro de un navegador de la *shell* de conformidad con la modalidad de la presente invención. El método 2600 incluye el despliegue de una ventana de bienvenida y los metadatos asociados con la ventana de bienvenida en 2602. Luego, en 2604, el método proporciona un control para modificaciones de los metadatos desplegados, por parte del usuario. Cuando el usuario manipula el control para modificar los metadatos desplegados en 2606, el método asocia los metadatos modificados con la ventana de bienvenida en 2608, de tal forma que los datos modificados se desplieguen la próxima vez que se despliegue la ventana de bienvenida.

La FIGURA 56 es un diagrama de flujo que ilustra un método 2700 para permitirle al usuario modificar los metadatos desplegados en una ventana seleccionada dentro de un navegador de la *shell* de acuerdo con la presente

invención. En 2702, el método 2700 primero despliega un número de objetos, tales como los objetos en una ventana de bienvenida u objetos en un contenedor seleccionado. Cuando el usuario selecciona uno o más objetos en 2704, el método despliega los metadatos asociados con el (los) objeto(s) 5 seleccionado(s) en 2706. En 2708 el método suministra un control para que el usuario modifique los metadatos desplegados. Cuando un usuario manipula el control para modificar los metadatos desplegados en 2710, el método entonces asocia los datos modificados con los objetos seleccionados en 2712, en tal forma que los metadatos sean desplegados la próxima vez que el (los) objeto(s) 10 sean mostrados.

En el evento de que un usuario seleccione objetos múltiples en 2704, los metadatos desplegados pueden incluir propiedades comunes de los objetos seleccionados (intersecciones), uniones de propiedades, o quizás una nueva propiedad relevante de los objetos seleccionados. Alternativamente, los 15 metadatos desplegados pueden incluir una muestra de metadatos que rota de cada uno de los elementos seleccionados (e.g. alternando cíclicamente de uno de los metadatos de un objeto seleccionado al siguiente metadatos del objeto seleccionado cada 30 segundos). Es posible desplegar metadatos que resultarían de una selección de todos los objetos que sean idénticos en el 20 despliegue de metadatos, que resultaría de una selección nula.

La FIGURA 57 es un diagrama de bloques de una estructura de datos 2800 que contiene metadatos modificables por el usuario asociados con un

objeto desplegado dentro de un navegador de la *shell* . La estructura de datos 2800 incluye un campo de título 2802 que indica el nombre del objeto. En el caso de objetos diferentes a archivos, el campo de título 2802 puede contener el nombre de cualquiera de las propiedades que se usen para alfabeticar aquel

5 objeto en una vista de listas. La estructura de datos 2800 un campo de las propiedades editables por parte del usuario 2804 que contiene una o más propiedades asociadas con el objeto desplegado y que vale la pena desplegar dentro del *navegador de la shell* . Dadas las limitaciones de tamaño del despliegue de metadatos en el navegador de la *shell* , el número de

10 propiedades en los campos 2804 y 2806 puede ser limitado. En consecuencia, la estructura de datos 2800 puede opcionalmente incluir un campo de todas las propiedades 2808, que contiene un enlace o un apuntador hacia una localización (e.g. una página de propiedades) que contenga todas las propiedades de los metadatos asociados con el objeto desplegado. Por

15 supuesto, el campo de todas las propiedades 2808 no sería necesario en el evento de que los campos 2804 y 2806 contengan todas las propiedades asociadas con el objeto desplegado. La estructura de datos 2800 se almacena en uno o más medios legibles por computador, tal como en un sistema de archivos en una *shell*, para suministrar vistas ricas en almacenamientos, y así

20 lograr una experiencia mejorada del usuario, dentro del navegador de la *shell* .

La presente invención permite un número de escenarios que no eran posibles con navegadores de la *shell* convencionales. Como primer ejemplo,

una estudiante puede manejar sus proyectos usando una vista de previsualización. Cuando obtenga nuevos documentos como parte de un proyecto en el cual esté trabajando, ella podrá seleccionar los documentos en su librería de documentos y digitar el nombre de un autor y el nombre del

5 proyecto dentro de los campos de palabras clave usando el control de edición. Ahora los nuevos documentos se mostrarán en si vista favorita: "Documentos agrupados por Palabras Claves y Listados por Autor". Un segundo ejemplo de un nuevo escenario habilitado por la presente invención involucra a un empleado buscando el material para una campaña publicitaria que se avecina.

10 En la medida en que navega a través de la colección de fotos de su empleador, usando el navegador de la *shell* , selecciona algunas fotografías y desde la ventana de previsualización, adiciona una nueva palabra clave "Campaña para el Verano 2003". Habiendo actualizado los metadatos para la selección múltiple, el empleado busca con la palabra clave y así puede visualizar todos

15 los archivos de la clave "Campaña para el Verano 2003" en conjunto. Muchos otros escenarios que se aprovechan de la presente invención serán aparentes para aquellos versados en la técnica.

Las FIGURAS 58-66 son diagramas referentes a un sistema y un método para ampliar la funcionalidad de un previsualizador de objetos en un navegador

20 de la *shell* configurado para desplegar una pluralidad de objetos que representan múltiples tipos de objetos. Como se describirá en mayor detalle a continuación, se suministra un navegador de la *shell* que incluye un

previsualizador predeterminado y un mecanismo de ampliación. El previsualizador predeterminado suministra un nivel de funcionalidad para múltiples tipos de objetos. El mecanismo de ampliación permite la funcionalidad más allá del nivel estándar suministrado por el previsualizador predeterminado para uno o más de los tipos de objetos.

La FIGURA 58 es un diagrama esquemático del estado anterior de la técnica de una interface gráfica de usuario para navegar en torno a fotos almacenadas en una carpeta dentro de un ambiente de navegador de la *shell* que se usa para visualizar otros archivos y carpetas no gráficos. Como se afirmó anteriormente, la necesidad de identificar con facilidad los objetos almacenados en un ambiente de computación tal como un PC está aumentando dramáticamente. Respecto de las imágenes digitales, los usuarios tradicionalmente debían invocar un programa de computador (*software*) de un tercero para poder ver un archivo específico en el PC. La FIGURA 58 ilustra una solución anterior, una vista de una porción de una película, que le permite a los usuarios ver con mayor facilidad e identificar la imagen asociada con un archivo dado dentro del ambiente operacional gráfico. La meta de la porción de la película era aliviar la necesidad de usar otros programas de computador al navegar dentro de una carpeta de fotos, mediante el suministro de un proceso rápido iterativo que le permita a un usuario previsualizar una imagen de un tamaño manejable de uno o más archivos gráficos dentro de la carpeta.

La FIGURA 58 se refiere a un sistema de navegación a través de imágenes almacenadas en una carpeta, donde una serie de imágenes de carpetas se presenta como una sola fila de imágenes tamaño pulgar dentro de un ambiente utilizado para visualizar otros archivos o carpetas no pictóricos (i. e., un navegador de la *shell*). Además le permite a un usuario selectivamente colocar el cursos encima de las imágenes tamaño pulgar, al tiempo que despliega una previsualización de la imagen agrandada de la selección de una de las imágenes tamaño pulgar. La FIGURA 58 es un diagrama de una ventana representativa sobre la pantalla de un usuario. Como se muestra, la ventana 3200 se divide en varias áreas incluyendo una región de encabezado, un área de opciones de tareas 3206, un área de control de previsualizaciones 3202, un área para comentarios y un área para porciones de películas 3204. El área de opciones de tareas 3206 contiene una lista de tareas que se pueden seleccionar por parte de un usuario para desempeñar una variedad de operaciones referentes a la administración de archivos y carpetas, así como otras opciones del sistema. Algunas de estas operaciones son propias de las imágenes del área de la porción de película 3204 y del área de control de previsualización 3202. El área de control de previsualización 3202 es un espacio en el cual se va a desplegar una imagen seleccionada por el usuario, en una vista de previsualización agrandada. Este espacio también puede contener íconos de navegación para proporcionarle asistencia a un usuario durante la iteración a través de una serie de imágenes. Inmediatamente debajo del área de control de previsualización se encuentra un área de comentarios

que puede ser utilizada para desplegar una variedad de información de texto. El área de la porción de la película 3204 suministra un espacio para desplegar una sola fila de imágenes tamaño pulgar P1, P2, P3 y P4 de los archivos gráficos contenidos dentro de una carpeta dada. Adicionalmente, el área de la

5 porción de la película 3204 también contiene cursores que le permiten al usuario recorrer una carpeta en relación con los archivos de imágenes. Se debe anotar que el área de la porción de la película 3204 puede contener y desplegar imágenes con una orientación mixta. Por ejemplo, como se muestra en la FIGURA 58, P1, P2 y P4 se encuentran en modo paisaje mientras que P3

10 se encuentra como un retrato.

Un usuario puede hacer una selección de cualquiera de las imágenes tamaño pulgar, lo que causará el despliegue de la imagen tamaño pulgar seleccionada en una previsualización más grande dentro del área de control. Adicionalmente, la selección de una imagen tamaño pulgar también le permitirá

15 al usuario seleccionar y desarrollar cualquiera de las tareas listadas en el área de opción de tareas 3206, respecto de la imagen seleccionada. Un primer botón de control le permite al usuario rápida y sucesivamente previsualizar una imagen agrandada de cada una de las imágenes tamaño pulgar dentro de una carpeta dada, mediante la iteración en una dirección. En otras palabras, un

20 usuario no tendría que específicamente seleccionar todas y cada una de las imágenes sucesivas tamaño pulgar con el fin de poder previsualizar el cuadro. En su lugar, el usuario solamente seleccionaría el primer botón de control

repetidamente para moverse a través de una carpeta. Un segundo botón de control realiza una función de iteración similar, pero solamente en la dirección opuesta.

Observando la FIGURA 59, una ventana 3300 representa un despliegue del tamaño de la pantalla para una interface gráfica de usuario de un navegador de la *shell* de propósito general. La ventana 3300 contiene un área de visualización 3302 y un área de visualización 3304. La ventana de previsualización 3302 puede incluir un control de previsualización 3306, un control de edición o de metadatos 3308, y un control de tareas 3310. Típicamente, el control de previsualización 3306 le proporcionará al usuario una imagen u otro despliegue visual del objeto que está siendo previsualizado (e.g. in archivo seleccionado). El control de previsualización 3306 también puede presentarle al usuario controles tales como el botón de iteración que le permiten al usuario cambiar el foco de un objeto a otro mediante el accionamiento de un botón del ratón. El control de edición 3308 no solo incluye una representación de las propiedades clave del objeto previsualizado, sino también le presentan al usuario un control para editar los metadatos. El control de tareas 3310 contiene tareas relevantes a los nombres de espacio y/o a la selección.

Aquellos versados en el la técnica podrán apreciar que la presente invención contempla la presencia de características opcionales dentro de la ventana 3300. Por ejemplo, el control de metadatos 3208 y el control de tareas 3210 no son características esenciales para los fines de la presente invención.

Más aún otras características no esenciales que no se muestran en la FIGURA 59, tales como una barra de herramientas que incluye botones de iteración o un botón de mostrar / ocultar para que el usuario pueda abrir / cerrar una ventana de previsualización, también se encuentran dentro del alcance de la presente invención. No obstante, estas y otras características opcionales pueden prestarle asistencia al usuario para localizar un objeto en particular con más facilidad dentro del navegador de la *shell*.

El área de visualización 3304 le proporciona al usuario una vista de listas de uno o más objetos 3312, tales como archivos del sistema o carpetas. El término "vista de lista" se refiere a la enumeración o lista dentro de un contenedor. Los términos "objeto" y "objeto de la shell", se usan indistintamente aquí, para hacer referencia a archivos, carpetas y otros contenedores, y otros objetos diferentes a archivos que se pueden representar en una vista de la lista. En forma similar, el "objeto *shell*" se refiere a un objeto en la librería *shell*. Ejemplos de objetos diferentes a archivos pueden incluir sin limitarse a, contactos, favoritos y mensajes de correo electrónico. Los términos "navegador de la *shell*" y "navegador de archivos del sistema" se usan indistintamente aquí para referirse a un navegador que permite la navegación a través de varios nombres de espacios incluyendo archivos y otros objetos diferentes a archivos.

Aquellos versados en la técnica apreciarán que la presente invención contempla muchos diseños posibles y representaciones para la ventana 3300. Por ejemplo, la ventana de previsualización 3302 se muestra encima del área

de visualización 3304 en la FIGURA 59. No obstante, otras representaciones, tales como colocar la ventana de previsualización 3302 y el área de visualización 3304, lado a lado, se encuentran claramente dentro del alcance de la presente invención. También existen muchas vistas posibles para los
5 objetos mostrados en el área de visualización 3304, tales como detalles, presentación de diapositivas, sección de película, imagen tamaño pulgar, capas, íconos, etc.

A continuación, haciendo referencia a la FIGURA 60, se proporciona una ilustración esquemática de una ventana de bienvenida 3400 en el navegador de
10 la *shell*. Una vista de bienvenida a veces se llama una ventana de "selección nula" porque representan un nombre de espacio o un contenedor, a diferencia de proporcionar la posibilidad de una selección. Si el usuario aún no ha realizado una selección, un ventana de previsualización 3402 2302 despliega los metadatos 3404 y las tareas clave referentes a la carpeta o a la librería de la
15 *shell*. Si se desea, las tareas pueden separarse en tareas principales 3406 y otras tareas relevantes 3408. La ventan de bienvenida 3400 2300 también incluye un área de visualización 3410, en la cual archivos múltiples y otros objetos 3412 se pueden revisar. Los metadatos en la ventana de bienvenida 3404 pueden incluir información tal como las propiedades del contenedor (e.g.
20 Mis imágenes), en cuyo caso el despliegue de metadatos puede ser estático. Alternativamente, los metadatos en la ventana de bienvenida 3404 2304 pueden incluir información tal como muestreo de metadatos de cada uno de los

objetos dentro del contenedor, , en cuyo caso el despliegue de metadatos puede cambiar con frecuencia. Por ejemplo, el despliegue de metadatos puede limitarse a las propiedades de un objeto a la vez, mediante la alternación de un objeto al siguiente cíclicamente cada 30 segundos.

5 La FIGURA 61 es una ilustración esquemática de una ventana seleccionada 3500 en un navegador de la *shell* . A diferencia de la ventana de bienvenida, la ventana seleccionada representa una selección por parte del usuario. Si el usuario selecciona un contenedor o una carpeta, la ventana seleccionada no necesita ser idéntica a la ventana de bienvenida para ese
10 contenedor o carpeta. En la FIGURA 61 la ventana seleccionada 3500 incluye una ventana de previsualización 3502 que contiene un control de previsualización 3504 , un despliegue de metadatos 3506 y un despliegue de tareas 3508. Similar a la ventana de bienvenida, 3400 (en la FIGURA 60), la ventana de selección 3500 también incluye un área de visualización 3510, en la
15 cual múltiples archivos u otros objetos 3512 se pueden visualizar. Sin embargo en la FIGURA 61 el usuario ha seleccionado uno de los archivos. En consecuencia, el control de previsualización 3504 despliega una imagen de previsualización del archivo seleccionados, el despliegue de los metadatos 3506 y un despliegue de tareas 3508 proporciona un menú de tareas relevantes
20 para la operación del archivo seleccionado.

La FIGURA 62 es una representación esquemática de la ventana seleccionada 3500 de la FIGURA 61, pero con controles extendidos de acuerdo

con la modalidad de la presente invención. La ventana seleccionada 3600 incluye una ventana de previsualización 3602, que contiene un control de previsualización 3604 con controles extendidos 3614, un despliegue de metadatos 3606 y 3506 y un despliegue de tareas 3608. La ventana
5 seleccionada 3600 también incluye un área de visualización 3610, en la cual se pueden visualizar archivos múltiples, u otros objetos 3612 se pueden visualizar. El usuario ha seleccionado uno de los archivos 312, de tal forma que el control de previsualización 3604 despliega una imagen previsualizada del archivo seleccionado, el despliegue de los metadatos 3606 muestra las propiedades del
10 archivo seleccionado y el despliegue de tareas 3608 proporciona un menú de tareas relevantes para la operación del archivo seleccionado.

Los controles extendidos 3614 pueden estar disponibles para el usuario como parte de un previsualizador en un navegador de la *shell*. El término “previsualizador” se puede referir a un control de previsualización o a una
15 ventana de previsualización que incluye un control de previsualización. La presente invención contempla un navegador de la *shell* que le proporciona al usuario un previsualizador predeterminado que ofrece un nivel estándar de funcionalidad para tipos de objetos múltiples y uno o más previsualizadores predeterminados que ofrecen diferentes niveles de funcionalidad para tipos
20 particulares de objetos para mejorar la experiencia del usuario. Abriendo el desarrollo de previsualizadores a los vendedores independientes de programas de computador (ISVs por sus siglas en ingles: Independent Software vendors) y

a desarrolladores externos le da valor agregado a la experiencia de navegación a través de archivos mostrando aspectos relevantes de los archivos de una manera fácilmente reconocible. La presente invención contempla previsualizadores adaptados a la medida para numerosos tipos de archivos y tipos de objetos diferentes a archivos, incluyendo, sin limitarse a archivos de imágenes, archivos de videos, contactos, juegos, escáners, cámaras de video, archivos de documentos, archivos de hojas electrónicas, archivos de presentaciones de diapositivas, archivos de dibujo y archivos de tabletas de tinta.

La presente invención proporciona un número de escenarios que no eran posibles con los navegadores de *shell* convencionales, algunos de los cuales de han descrito arriba. Se les permite a terceros describir y demostrar sus tipos de archivos mediante el suministro del código que pueda mirar al interior del tipo de archivo y suministrar una imagen con significado que el usuario pueda entender. Por ejemplo, Apple pudo implementar el control de previsualización de QuickTime™ dentro del navegador de la *shell* . Este control de previsualización podría proporcionar una alternativa o un nivel extendido de funcionalidad más allá del que tiene el previsualizador predeterminado en el *shell* del sistema operacional, incluyendo la funcionalidad tal como la de mostrar los primeros cinco segundos de una película QuickTime™ v/o el ofrecer botones y controles para que el usuario lance el reproductor QuickTime™ . Un previsualizador alternativo para un archivo de música podría suministrar una

funcionalidad extendida similar. Como lo podrán apreciar aquellos versados en la técnica, las posibilidades de ampliar la funcionalidad en un previsualizador alternativo son ilimitadas.

La FIGURA 63 es una ilustración esquemática de una ventana seleccionada similar a la FIGURA 61 pero que también incluye un menú de contexto 3714 para permitirle al usuario modificar los metadatos en un navegador de la *shell* de acuerdo con la modalidad de la presente invención. La ventana seleccionada 3700 incluye una ventana de previsualización 3702 que contiene un control de previsualización 3704 , un despliegue de metadatos 3706 y un despliegue de tareas 3708. La ventana de selección 3700 también incluye un área de visualización 3710, en la cual múltiples archivos u otros objetos 3712 se pueden visualizar. Aquellos versados en la técnica apreciarán que, para los fines de la presente invención, el control de metadatos 3706 y el control de tareas 3708 no son características esenciales. La presente invención contempla la presencia de éstas y/o otras características opcionales que pueden ayudar al usuario para localizar más fácilmente un objeto en particular en el navegador de la *shell* o de otra forma mejorar la experiencia del usuario.

El menú de contexto 3714 en la FIGURA 63 le proporciona al usuario varias opciones, incluyendo la escogencia para seleccionar ya sea el previsualizador predeterminado o un previsualizador alternativo para un objeto seleccionado. El texto genérico mostrado en el menú 3714 es por supuesto solamente un ejemplo del tipo de opciones que se le pueden presentar al

usuario para la selección de un previsualizador. El menú de contexto se puede proporcionar en cualquier ventana, incluyendo una ventana de bienvenida, para mejorar la experiencia del usuario. Aquellos versados en la técnica podrán apreciar que cualquier número y variedad de menús de contexto pueden ser soportados por la presente invención. Para fines de la presente invención, uno de los medios para permitirle al usuario la selección de un previsualizador dentro de un navegador de la *shell* es proporcionándole un menú de contexto tal como el menú de contexto 3714. Un usuario puede convocar el menú de contexto, por ejemplo, mediante la selección sobre el texto u objeto correspondiente en la ventana de previsualización.

Aquellos versados en la técnica podrán apreciar que la presente invención contempla medios diferentes a los menús de contexto para seleccionar un previsualizador para los objetos desplegados, a partir de una pluralidad de visualizadores dentro del navegador de la *shell*. Otro de tales medios es que el usuario realice una selección (haga *click*) sobre el control de previsualización para determinar un modo de selección. En forma similar, el usuario podrá ser solicitado para seleccionar un previsualizador mediante la selección con el botón derecho dentro de la ventana de previsualización. En contraste, un usuario podría entrar al modo selección "rondando" sobre el texto u objeto relevante en la ventana de previsualización. Hay numerosos medios alternativos que se encuentran dentro del alcance de la presente invención.

La FIGURA 64A es un diagrama de flujo 3800 para permitirle al usuario seleccionar un previsualizador en un navegador de la *shell* que soporta múltiples tipos de objetos de acuerdo con la modalidad de la presente invención. El método 3800 proporciona una pluralidad de previsualizadores en el navegador de la *shell* en 3802. La pluralidad de previsualizadores puede incluir un previsualizador predeterminado para múltiples tipos de objetos y uno o más previsualizadores alternativos para múltiples tipos de objetos en particular. Estos previsualizadores alternativos pueden incluir aplicaciones instaladas desarrolladas por terceros. En 3804, el método 3800 le presenta al usuario una opción de dos o más previsualizadores para un tipo de objeto en particular. La solicitud para seleccionar un previsualizador se puede iniciar en el navegador de la *shell* (e.g. al desplegar un nuevo tipo de objeto) y/o por el usuario (e.g. mediante la selección de un nuevo tipo de objeto para desplegar un menú de contexto). Luego de recibir una entrada del usuario 3806 indicando una selección de uno de los previsualizadores para un tipo de objeto en particular, el método 3800 asocia el previsualizador seleccionado con el objeto particular en 3808. El previsualizador seleccionado permanecerá en uso hasta que el usuario seleccione uno diferente. Sin embargo, si el previsualizador seleccionado es una aplicación instalada, la desinstalación de la aplicación también terminará el uso del previsualizador seleccionado.

La FIGURA 64B es un diagrama de flujo que ilustra el método 3810 para la selección automática de un previsualizador en un navegador de la *shell* que

soporta múltiples tipos de objetos de acuerdo con la modalidad de la presente invención. El método 3810 proporciona una pluralidad de previsualizadores en el navegador de la *shell* en 3812. La pluralidad de previsualizadores puede incluir un previsualizador predeterminado para múltiples tipos de objetos y uno o
5 más previsualizadores alternativos para tipos de objetos en particular. Estos previsualizadores alternativos pueden incluir aplicaciones instaladas por terceros.

En 3814, el sistema (a diferencia del usuario) automáticamente y en forma transparente selecciona un previsualizador predeterminado entre dos o
10 más previsualizadores disponibles para un tipo particular de objeto. El sistema puede seleccionar un previsualizador como respuesta a un evento como el despliegue de un nuevo tipo de objeto o la presencia de un previsualizador alternativo. El sistema está configurado para seleccionar un previsualizador predeterminado con base en reglas lógicas. Bajo circunstancias excepcionales,
15 el sistema podrá decidir en 3816 de pasar por encima de las reglas y seleccionar un previsualizador que no hubiera sido seleccionado bajo las reglas aplicables. Por ejemplo, si la regla consiste en seleccionar un previsualizador recientemente disponible sobre el previsualizador predeterminado actual, una aplicación instalada puede generalmente tener la autorización de cambiar el
20 previsualizador predeterminado a el previsualizador que ahora está disponible a partir de la aplicación instalada. Sin embargo, el navegador de la *shell* , por ejemplo, puede reservarse el derecho de pasar por encima del cambio

propuesto por la aplicación recién instalada. Por ejemplo, este "pasar por encima" puede ser apropiado cuando la aplicación recién instalada no puede autenticarse como un usuario apropiado del tipo de objeto bajo estudio.

En cualquier caso, el método 3810 luego asocia el previsualizador
5 seleccionado con el tipo de objeto en particular en 3818. EL previsualizador seleccionado permanecerá en uso hasta que se seleccione uno diferente. Sin embargo. Si el previsualizador seleccionado es una aplicación instalada, la desinstalación de la aplicación también dará por terminado el uso del previsualizador seleccionado.

10 En relación con la siguiente FIGURA 65, un diagrama de flujo ilustra el método 3900 para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporte múltiples tipos de objetos de acuerdo con la modalidad de la presente invención. El método 3900 incluye el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples
15 tipos de objetos en 3902. El método 3900 además incluye el suministro de un mecanismo de ampliación para el desarrollo de un previsualizador alternativo por parte de terceros para por lo menos uno de los tipos múltiples de objetos en 3904. El previsualizador alternativo puede registrarse en el navegador de la *shell* en 3906. En el caso de una aplicación instalada, el registro puede ocurrir
20 sustancialmente en el momento de la instalación. Por ejemplo, si la aplicación es instalada por parte de un OEM, el previsualizador alternativo puede registrarse antes de que el usuario haya adquirido el computador.

Alternativamente, el usuario puede instalar la aplicación en forma local o remota.

Existen muchos enfoques para el mecanismo de ampliación referenciado arriba en 3904. Uno de tales enfoques involucra la exposición de un conjunto de interfaces de programas de aplicación (APIs por sus siglas en inglés *Application Program Interfaces*) de tal forma que los vendedores independientes de programas (ISVs) y otros desarrolladores externos puedan desarrollar previsualizadores alternativos. Con el enfoque API, el mecanismo de registro existe, permitiéndole a un ISV asociar su control de previsualización con un tipo de objeto de propiedad del ISV. Cuando un objeto de ese tipo de archivo se selecciona dentro del navegador de la *shell*, el control de previsualización del ISV se activa instantáneamente vía el mecanismo de registro y mediante la ampliación API. El API suministra datos al control de previsualización: datos que representan los objetos seleccionados en la vista y datos que representan el contenedor paterno de los objetos en la vista. El control de previsualización opera sobre estos datos y proporciona una interface a través del API, la cual se presenta en el navegador de la *shell*. El usuario puede proporcionar entradas con acciones de teclas y eventos del ratón que son transmitidos por el navegador de la *shell* hacia el control de previsualización, que puede operar sobre esos eventos de entrada por parte del usuario.

Aquellos versados en la materia podrán apreciar que muchos enfoques son posibles en el contexto del mecanismo de ampliación de la presente

invención. Adicionalmente al enfoque API, se puede lograr una funcionalidad similar mediante la configuración por parte del usuario, por medio de un apuntador hacia HTML o alojando un "flash". Más aún, el modelo de ampliación puede requerir que una sola aplicación que sea propietaria del tipo de objeto
5 seleccionado pueda proporcionar solamente un previsualizador. En otras palabras, el número de previsualizadores disponibles puede estar limitado a un previsualizador predeterminado y un previsualizador alternativo para evitar una experiencia pobre del usuario en la cual múltiples previsualizadores registrados ampliados están compitiendo entre sí. Sin embargo, otro modelo sería permitir
10 cualquier aplicación que pudiera manejar el tipo de objeto seleccionado para suministrar un previsualizador adicional. Un modelo alternativo sería permitir que cualquier código corriera para suministrar un previsualizador adicional para cualquier tipo de objeto. También puede ser deseable bajo ciertas circunstancias permitir el reemplazo o el retiro del previsualizador
15 predeterminado. Muchos otros modelos son posibles y están contemplados dentro el alcance de la presente invención.

La FIGURA 66 es un diagrama de bloques de una estructura de datos 4000 almacenada en uno o más medios legibles por computador y que contiene información que indica una pluralidad de previsualizadores en un navegador de
20 la *shell*. La estructura de datos 4000 incluye un campo para un previsualizador predeterminado 4002 que contiene información que indica un previsualizador predeterminado que soporta varios tipos de objetos. Un campo para un

previsualizador alternativo 4004 contiene información que indica un previsualizador alternativo para el primer tipo de objeto. Otro campo de previsualizador alternativo 4006 puede contener información que indica sobre un segundo previsualizador alternativo para el primer tipo de objeto, o puede
5 contener información que indica sobre un previsualizador alternativo para un segundo tipo de objeto. Aquellos versados en la técnica apreciarán que en algunos casos puede haber solamente un campo para previsualizador alternativo, y en otros casos puede haber dos o más campo de previsualizadores alternativos. El campo de previsualizador seleccionado 4008
10 contiene información que indica si se invoca el previsualizador predeterminado o un previsualizador alterno cuando los objetos de un determinado tipo se despliegan en el navegador de la *shell*. En ese evento el campo 4006 contiene información que indica un previsualizador alterno para un segundo tipo de objetos, un campo de visualizador seleccionado 4010 puede contener
15 información que indica si se debe invocar el previsualizador predeterminado o el previsualizador alterno cuando uno o más objetos del segundo tipo de objetos se despliegan en el navegador de la *shell*. La información contenida en los campos 4002, 4004 y/o 4006 puede comprender el código del previsualizador que está configurado para correr cuando un usuario selecciona un tipo de
20 objeto.

Mientras que la modalidad preferida de la invención ha sido ilustrada y descrita, se apreciará que varios cambios se pueden hacer en ella sin alejarse

del espíritu ni del alcance de la invención. Por ejemplo, se apreciará que las localizaciones de varias características de la interface de usuario (UI) que se muestran aquí son ilustrativas y pueden ser modificadas, y que diferentes colocaciones de varias características de la UI seguirán estando dentro del
s espíritu y alcance de la invención. Además, los diferentes aspectos de la invención aquí descritos se pueden conformar en varias combinaciones, también sin alejarse del espíritu y del alcance de la invención.

Las modalidades de la invención en la cual se reivindica una propiedad exclusiva o privilegio se definen a continuación:

1. En un sistema de computador con un elemento de despliegue y memoria para almacenar objetos, un método para presentar colecciones de los
5 objetos, que comprende:

la creación de una librería que contiene los objetos con una o más propiedades de metadatos:

el agrupamiento automático de objetos con uno o más propiedades de metadatos especificada en la librería; y

10 que suministra una o más herramientas para manipular los objetos de la librería.

2. El método de la Reivindicación 1, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell* , el método del navegador de la
15 *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto
20 seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

3. El método de la Reivindicación 1, que además comprende el suministro de un navegador de la *shell*, el navegador de la *shell* que
5 comprende:

una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
10 por lo menos una porción de los valores de metadatos desplegados en la ventana.

4. El método de la Reivindicación 1, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el
15 método de previsualización que comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los
20 múltiples tipos de objetos.

5. El método de la Reivindicación 1, que además comprende un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

- 5 un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

- 10 6. El método de la Reivindicación 1, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

- 15 el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

- la adición de uno o más segmentos interactivos a la barra de direcciones
20 virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

7. El método de la Reivindicación 1, que además comprende un control de interface de usuario de la barra de direcciones virtuales para
5 seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

8. El método de la Reivindicación 1, que además comprende un
10 método para filtrar objetos, dicho método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado
15 de los mismos.

9. El método de la Reivindicación 1, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir
20 de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

5 10. El método de la Reivindicación 1, que además comprende un método para representar objetos ante el usuario, dicho método para representar objetos ante el usuario que comprende:

el suministro de una base de datos que permite la búsqueda tanto de objetos tipo archivo como de objetos diferentes a los archivos, a partir de una
10 consulta;

recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos tanto de objetos tipo archivo como de objetos diferentes a los archivos que coinciden con la consulta; y

presentar tanto de objetos tipo archivo como de objetos diferentes a los
15 archivos que coinciden con la consulta en el despliegue.

11. Un sistema para exponer ante un usuario los objetos que están almacenados en memoria, que comprende:

una carpeta de procesamiento que obtiene las consultas del usuario:

una base de datos relacional para almacenar la información referente a
20 los objetos; y

donde:

la carpeta de procesamiento obtiene una consulta de un usuario y la pasa a la base de datos relacional;

la base de datos relacional suministra los resultados de regreso a la carpeta de procesamiento; y

5 con base en los resultados de la base de datos relacional, la carpeta de procesamiento suministra los resultados al usuario en forma de carpetas virtuales.

12. El sistema de la Reivindicación 11, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los
10 metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

15 el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

13. El sistema de la Reivindicación 11, que además comprende un
20 navegador de la *shell*, el navegador de la *shell* que comprende:

una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
5 por lo menos una porción de los valores de metadatos desplegados en la ventana.

14. El sistema de la Reivindicación 11, que además comprende un previsualizador para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el previsualizador
10 que comprende:

un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de
15 objetos.

15. El sistema de la Reivindicación 11, que además comprende un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

20 un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

16. El sistema de la Reivindicación 11, que además comprende una
5 barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador, dicha barra de direcciones virtuales que comprende:

una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los
10 objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

15 donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

17. El sistema de la Reivindicación 11, que además comprende un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso
20 para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

18. El sistema de la Reivindicación 11, que además comprende un método para filtrar objetos, dicho método para filtrar objetos que comprende:

5 el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado de los mismos.

19. El sistema de la Reivindicación 11, que además comprende un método para presentar colecciones de objetos, el método para presentar colecciones de objetos que comprende:

15 la creación de una librería que contiene objetos con una o más de las propiedades de metadatos especificadas;

el agrupamiento automático de objetos con una o más propiedades de metadatos especificadas, dentro de la librería; y

el suministro de una o más herramientas para manipular los objetos en la librería.

20. El sistema de la Reivindicación 11, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir
5 de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el
10 despliegue.

21. En un sistema de computador con un despliegue y una memoria para almacenar objetos, los objetos con propiedades de metadatos asociados, un método para exponer los objetos ante el usuario, el método que comprende:

la selección de una primera propiedad de metadatos;

15 la búsqueda de objetos que tienen dicha propiedad de metadatos seleccionada; y

el suministro de un despliegue de un objeto de la primera carpeta virtual que representa la colección de objetos que tienen la primera propiedad de metadatos seleccionada.

20 22. El método de la Reivindicación 21, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los

metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de
5 por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto
seleccionado; y

el suministro de un control para las modificaciones por parte del usuario
de los metadatos desplegados.

10 23. El método de la Reivindicación 21, que además comprende el
suministro de un navegador de la *shell*, el navegador de la *shell* que
comprende:

una ventana que despliega un grupo de objetos, donde la ventana
también despliega los valores de metadatos asociados a uno o más de los
15 objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
por lo menos una porción de los valores de metadatos desplegados en la
ventana.

24. El método de la Reivindicación 21, que además comprende un
20 método de previsualización para permitir el uso de previsualizadores de

terceros en un navegador de la *shell* que soporta múltiples tipos de objetos. el método del previsualizador que comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

- 5 el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de objetos.

25. El método de la Reivindicación 21, que además comprende el suministro de un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

- un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

26. El método de la Reivindicación 21, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

- 5 la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

- 10 27. El método de la Reivindicación 21, que además comprende el suministro de un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

- 15 una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

28. El método de la Reivindicación 21, que además comprende un método para filtrar objetos que comprende:

- 20 el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado
5 de los mismos.

29. El método de la Reivindicación 21, que además comprende un método para presentar colecciones de objetos, el método para presentar colecciones de objetos que comprende:

la creación de una librería que contiene objetos con una o más de las
10 propiedades de metadatos especificadas;

el agrupamiento automático de objetos con una o más propiedades de metadatos especificadas, dentro de la librería; y

el suministro de una o más herramientas para manipular los objetos en la librería.

15 30. El método de la Reivindicación 21, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir de las cuales se obtienen los objetos, el alcance comprendiendo la memoria
20 actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

5 31. En un sistema de computador con un despliegue y una memoria para almacenar objetos, un método para manipular los objetos, que comprende:

representar grupos de objetos como carpetas virtuales; y

suministrar por lo menos una primera acción definida que puede desarrollarse para la manipulación directa de una carpeta virtual, donde dicha
10 primera acción definida se desarrolla, la carpeta virtual se manipula exactamente en la forma indicada por la acción desarrollada.

32. El método de la Reivindicación 31, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell* , el método del navegador de la
15 *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto
20 seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

33. El método de la Reivindicación 31, que además comprende el suministro de un navegador de la *shell*, el navegador de la *shell* que
5 comprende:

una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
10 por lo menos una porción de los valores de metadatos desplegados en la ventana.

34. El método de la Reivindicación 31, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el
15 método del previsualizador que comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los
20 múltiples tipos de objetos.

35. El método de la Reivindicación 31, que además comprende el suministro de un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

5 un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

10 36. El método de la Reivindicación 31, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

15 el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones
20 virtuales, donde cada segmento interactivo además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

37. El método de la Reivindicación 31, que además comprende el suministro de un control de interface de usuario de la barra de direcciones
5 virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

10 38. El método de la Reivindicación 31, que además comprende un método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos
15 desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado de los mismos.

39. El método de la Reivindicación 31, que además comprende un
20 método para presentar colecciones de objetos, el método para presentar colecciones de objetos que comprende:

la creación de una librería que contiene objetos con una o más de las propiedades de metadatos especificadas;

el agrupamiento automático de objetos con una o más propiedades de metadatos especificadas, dentro de la librería; y

5 el suministro de una o más herramientas para manipular los objetos en la librería.

40. El método de la Reivindicación 31, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de objetos que comprende:

10 la definición del alcance de las localizaciones de la memoria física a partir de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos desde las localizaciones físicas definidas en el alcance; y

15 presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

41. En un sistema de computador con un despliegue y una memoria para almacenar objetos con propiedades de metadatos, un método para filtrar objetos, que comprende:

20 el suministro de una pluralidad de objetos de despliegue en el despliegue en el cual que cada uno represente uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado
5 de los mismos.

42. El método de la Reivindicación 41, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

10 el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

15 el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

43. El método de la Reivindicación 41, que además comprende el suministro de un navegador de la *shell*, el navegador de la *shell* que comprende:

una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
5 por lo menos una porción de los valores de metadatos desplegados en la ventana.

44. El método de la Reivindicación 41, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el
10 método del previsualizador que comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los
15 múltiples tipos de objetos.

45. El método de la Reivindicación 41, que además comprende el suministro de un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

20 un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

46. El método de la Reivindicación 41, que además comprende un
5 método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma
10 que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones
virtuales, donde cada segmento interactivo adicional además restringe los
15 objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

47. El método de la Reivindicación 41, que además comprende el suministro de un control de interface de usuario de la barra de direcciones
20 virtuales para seleccionar objetos a los cuales el sistema de computador puede

tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

5 48. El método de la Reivindicación 41, que además comprende un método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

10 el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado de los mismos.

15 49. El método de la Reivindicación 41, que además comprende un método para presentar colecciones de objetos, el método para presentar colecciones de objetos que comprende:

la creación de una librería que contiene objetos con una o más de las propiedades de metadatos especificadas;

20 el agrupamiento automático de objetos con una o más propiedades de metadatos especificadas, dentro de la librería; y

el suministro de una o más herramientas para manipular los objetos en la librería.

50. El método de la Reivindicación 41, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de
5 objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener
10 los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

51. En un sistema de computador con un despliegue y una memoria para almacenar objetos con propiedades de metadatos, un método para
15 visualizar objetos seleccionados, que comprende:

el almacenamiento de un enlace hacia una colección de objetos deseada hacia la cual el usuario ya ha navegado;

presentación del enlace en el despliegue de tal forma que mediante la selección del ("click" sobre el) enlace el usuario puede regresar a la vista
20 correspondiente a la colección deseada de objetos

52. El método de la Reivindicación 51, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

5 el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

10 el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

53. El método de la Reivindicación 51, que además comprende el suministro de un navegador de la *shell*, el navegador de la *shell* que comprende:

15 una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de por lo menos una porción de los valores de metadatos desplegados en la
20 ventana.

54. . El método de la Reivindicación 51, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el método del previsualizador que comprende:

5 el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de objetos.

10 55. El método de la Reivindicación 51, que además comprende el suministro de un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

un previsualizador predeterminado que suministra un nivel estándar de
15 funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

56. El método de la Reivindicación 51, que además comprende un
20 método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador

usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

57. El método de la Reivindicación 51, que además comprende el suministro de un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

58. El método de la Reivindicación 51, que además comprende un método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

- 5 la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado de los mismos.

59. El método de la Reivindicación 51, que además comprende un método para presentar colecciones de objetos, el método para presentar
10 colecciones de objetos que comprende:

la creación de una librería que contiene objetos con una o más de las propiedades de metadatos especificadas;

el agrupamiento automático de objetos con una o más propiedades de metadatos especificadas, dentro de la librería; y

- 15 el suministro de una o más herramientas para manipular los objetos en la librería.

60. El método de la Reivindicación 51, que además comprende un método para el despliegue de objetos, dicho método para el despliegue de objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener
5 los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

61. En un sistema de computador con un despliegue y una memoria para almacenar objetos un método para representar los objetos ante el usuario,
10 que comprende:

el suministro de una base de datos que permita búsquedas de objetos de archivo y objetos diferentes a archivos de acuerdo con una consulta;

recepción de la consulta, y en respuesta a ella, obtener tanto los objetos tipo archivo como los objetos diferentes a los archivos que corresponden a la
15 consulta; y

presentar tanto los objetos tipo archivo como los objetos diferentes a los archivos que corresponden a la consulta en el despliegue.

62. El método de la Reivindicación 61, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los
20 metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto
5 seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

63. El método de la Reivindicación 61, que además comprende el suministro de un navegador de la *shell*, el navegador de la *shell* que
10 comprende:

una ventana que despliega un grupo de objetos, donde la ventana también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de
15 por lo menos una porción de los valores de metadatos desplegados en la ventana.

64. El método de la Reivindicación 61, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el
20 método del previsualizador que comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los
5 múltiples tipos de objetos.

65. El método de la Reivindicación 61, que además comprende el suministro de un navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

10 un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

15 66. El método de la Reivindicación 61, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

20 el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento

interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los
5 objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

67. El método de la Reivindicación 61, que además comprende el suministro de un control de interface de usuario de la barra de direcciones
10 virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

15 68. El método de la Reivindicación 61, que además comprende un método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos en el despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos
20 desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado de los mismos.

69. El método de la Reivindicación 61, que además comprende un
5 método para presentar colecciones de objetos, el método para presentar colecciones de objetos que comprende:

la creación de una librería que contiene objetos con una o más de las propiedades de metadatos especificadas;

el agrupamiento automático de objetos con una o más propiedades de
10 metadatos especificadas, dentro de la librería; y

el suministro de una o más herramientas para manipular los objetos en la librería.

70. El método de la Reivindicación 61, que además comprende un
método para el despliegue de objetos, dicho método para el despliegue de
15 objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener
20 los objetos desde las localizaciones físicas definidas en el alcance; y

presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

71. Un método para suministrar una trayectoria virtual hacia los objetos almacenados en un sistema de archivos de computador usando una barra de direcciones virtuales, el método comprendiendo:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

72. El método de la Reivindicación 71, donde se proporciona un filtro en el despliegue que puede ser seleccionado por parte de un usuario, para filtrar los objetos con base en las propiedades de metadatos especificados por el filtro.

73. El método de la Reivindicación 71, donde se proporciona un enlace rápido en el despliegue, tal que pueda ser seleccionado por parte de un usuario para visualizar una colección específica de objetos.

74. El método de la Reivindicación 71, donde se proporciona una librería que incluye una colección de objetos y un conjunto de herramientas para manipular los objetos de la librería.

75. El método de la Reivindicación 71, donde los objetos en una carpeta virtual pueden ser físicamente almacenados en memorias en diferentes localizaciones físicas.

76. El método de la Reivindicación 75, donde las diferentes localizaciones físicas comprenden el computador actual y por lo menos otro computador diferente, una localización sobre una red, y un dispositivo de almacenamiento externo.

77. El método de la Reivindicación 71, donde los objetos comprenden tanto objetos de archivos como objetos diferentes a archivos.

78. El método de la Reivindicación 77, donde los objetos diferentes a archivos comprenden por lo menos uno entre contactos o correos electrónicos.

79. El método de la Reivindicación 71, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell*, el método del navegador de la *shell* que comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

5 80. El método de la Reivindicación 71, que además comprende un método de previsualización para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el método del previsualizador que comprende:

10 el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de objetos.

15 81. Un método implementado por computador para permitirle al usuario modificar los metadatos en un navegador de la *shell* , el método comprende:

el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

20 el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

82. El método de la Reivindicación 81, donde se proporciona un filtro en el despliegue que puede ser seleccionado por parte de un usuario, para
5 filtrar los objetos con base en las propiedades de metadatos especificados por el filtro.

83. El método de la Reivindicación 81, donde se proporciona un enlace rápido en el despliegue, tal que pueda ser seleccionado por parte de un usuario para visualizar una colección específica de objetos.

10 84. El método de la Reivindicación 81, donde se proporciona una librería que incluye una colección de objetos y un conjunto de herramientas para manipular los objetos de la librería.

85. El método de la Reivindicación 81, donde los objetos en una carpeta virtual pueden ser físicamente almacenados en memorias en diferentes
15 localizaciones físicas.

86. El método de la Reivindicación 85, donde las diferentes localizaciones físicas comprenden el computador actual y por lo menos otro computador diferente, una localización sobre una red, y un dispositivo de almacenamiento externo.

20 87. El método de la Reivindicación 81, donde los objetos comprenden tanto objetos de archivos como objetos diferentes a archivos.

88. El método de la Reivindicación 87, donde los objetos diferentes a archivos comprenden por lo menos uno entre contactos o correos electrónicos.

89. El método de la Reivindicación 81, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual
5 hacia los objetos almacenados en el sistema de archivos de computador usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento
10 interactivo referencia los objetos en el sistema de archivos de computador de acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

15 donde el primer segmento interactivo y el otro u otros segmentos interactivos adicionales comprenden una trayectoria virtual.

90. El método de la Reivindicación 81, que además comprende el suministro de un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede
20 tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

91. Un método implementado por computador para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporte múltiples
5 tipos de objetos, el método comprende:

el suministro de un navegador de la *shell* con un previsualizador predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los
10 múltiples tipos de objetos.

92. El método de la Reivindicación 91, donde se proporciona un filtro en el despliegue que puede ser seleccionado por parte de un usuario, para filtrar los objetos con base en las propiedades de metadatos especificados por el filtro.

15 93. El método de la Reivindicación 91, donde se proporciona un enlace rápido en el despliegue, tal que pueda ser seleccionado por parte de un usuario para visualizar una colección específica de objetos.

94. El método de la Reivindicación 91, donde se proporciona una librería que incluye una colección de objetos y un conjunto de herramientas
20 para manipular los objetos de la librería.

95. El método de la Reivindicación 91, donde los objetos en una carpeta virtual pueden ser físicamente almacenados en memorias en diferentes localizaciones físicas.

96. El método de la Reivindicación 95, donde las diferentes
5 localizaciones físicas comprenden el computador actual y por lo menos otro computador diferente, una localización sobre una red, y un dispositivo de almacenamiento externo.

97. El método de la Reivindicación 91, donde los objetos comprenden tanto objetos de archivos como objetos diferentes a archivos.

10 98. El método de la Reivindicación 97, donde los objetos diferentes a archivos comprenden por lo menos uno entre contactos o correos electrónicos.

99. El método de la Reivindicación 91, que además comprende un método de barra de direcciones virtuales para suministrar una trayectoria virtual hacia los objetos almacenados en el sistema de archivos de computador
15 usando una barra de direcciones virtuales, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de
20 acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos
5 interactivos adicionales comprenden una trayectoria virtual.

100. El método de la Reivindicación 91, que además comprende el suministro de un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que
10 comprende:

una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

101. En un sistema computacional con un despliegue y una memoria para almacenar objetos, siendo los objetos de una pluralidad de diferentes
15 tipos, un método para presentar librerías de objetos que comprende:

suministro de librerías para contener objetos de tipos similares; y

suministro de un conjunto similar de Interfaces de Usuario (UI) para cada una de las librerías.

102. El método de la Reivindicación 101, donde los diferentes tipos
20 comprenden por lo menos dos clases de documentos, imágenes y música.

103. El método de la Reivindicación 101, donde los conjuntos de características similares de Interfaces de Usuario incluyen una característica para edición.

104. El método de la Reivindicación 101, donde los conjuntos de
5 características similares de Interfaces de Usuario incluyen una característica para crear categorías.

105. El método de la Reivindicación 101, donde los conjuntos de características similares de Interfaces de Usuario incluyen una característica para editar los metadatos de los objetos.

106. El método de la Reivindicación 101, donde los conjuntos de
10 características similares de Interfaces de Usuario incluyen una característica para alterar los pivotes.

107. El método de la Reivindicación 101, donde adicionalmente a
contener objetos de tipos similares, las librerías también pueden contener
15 objetos de diferentes tipos.

108. El método de la Reivindicación 101, que además comprende un método de navegador de la *shell* para permitirle al usuario modificar los metadatos dentro del navegador de la *shell* , el método del navegador de la *shell* que comprende:

20 el despliegue de una pluralidad de objetos;

recibir una primera entrada del usuario que representa la selección de por lo menos uno de los objetos desplegados;

el despliegue de los metadatos asociados con por lo menos el objeto seleccionado; y

5 el suministro de un control para las modificaciones por parte del usuario de los metadatos desplegados.

109. El método de la Reivindicación 101, que además comprende un navegador de la *shell*, el navegador de la *shell* que comprende:

una ventana que despliega un grupo de objetos; donde la ventana
10 también despliega los valores de metadatos asociados a uno o más de los objetos desplegados; y

un control de edición que permite la modificación por parte del usuario de por lo menos una porción de los valores de metadatos desplegados en la ventana.

15 110. El método de la Reivindicación 101, que además comprende un previsualizador para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporta múltiples tipos de objetos, el previsualizador que comprende:

un navegador de la *shell* con un previsualizador predeterminado para
20 múltiples tipos de objetos; y

un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de objetos.

111. El método de la Reivindicación 101, que además comprende un
5 navegador de la *shell* configurado para desplegar una pluralidad de objetos representando múltiples tipos de objetos, el navegador de la *shell* que comprende:

un previsualizador predeterminado que suministra un nivel estándar de funcionalidad para múltiples tipos de objetos; y

10 un mecanismo de ampliación que permite una funcionalidad superior a aquella suministrada por el nivel estándar proporcionado por el previsualizador predeterminado para uno o más de los múltiples tipos de objetos.

112. El método de la Reivindicación 101, que además comprende una barra de direcciones virtuales para suministrar una trayectoria virtual hacia los
15 objetos almacenados en el sistema de archivos de computador, dicha barra de direcciones virtuales que comprende:

el despliegue de una barra de direcciones virtuales operable en tal forma que incluya un primer segmento interactivo, donde el primer segmento interactivo referencia los objetos en el sistema de archivos de computador de
20 acuerdo con un filtro correspondiente para la selección de objetos; y

la adición de uno o más segmentos interactivos a la barra de direcciones virtuales, donde cada segmento interactivo adicional además restringe los objetos referenciados por los segmentos interactivos precedentes;

donde el primer segmento interactivo y el otro u otros segmentos
5 interactivos adicionales comprenden una trayectoria virtual.

113. El método de la Reivindicación 101, que además comprende un control de interface de usuario de la barra de direcciones virtuales para seleccionar objetos a los cuales el sistema de computador puede tener acceso para su despliegue, la barra de direcciones virtuales que comprende:

10 una pluralidad de segmentos interactivos, donde cada segmento interactivo corresponde a un filtro para seleccionar objetos para despliegue.

114. El método de la Reivindicación 101, que además comprende un método para filtrar objetos, dicho método para filtrar objetos que comprende:

el suministro de una pluralidad de despliegues de objetos en el
15 despliegue en el cual cada uno representa uno o más objetos;

el filtrado de los elementos representado en el despliegue por los objetos desplegados con base en el término de filtro; y

la modificación de los objetos desplegados que se suministran en el despliegue, tal que representen lo objetos que permanecen después del filtrado
20 de los mismos.

115. El método de la Reivindicación 101, que además comprende un método para desplegar objetos, el método para desplegar objetos que comprende:

la definición del alcance de las localizaciones de la memoria física a partir
5 de las cuales se obtienen los objetos, el alcance comprendiendo la memoria actual del computador y por lo menos una localización física diferente;

la recepción de una consulta, y en respuesta a dicha consulta, obtener los objetos desde las localizaciones físicas definidas en el alcance; y

10 presentar los objetos obtenidos a partir de la consulta en un vista en el despliegue.

116. El método de la Reivindicación 101, que además comprende un método para representar objetos ante un usuario, el método representar objetos ante el usuario que comprende:

el suministro de una base de datos que permita búsquedas de objetos de
15 archivo y objetos diferentes a archivos de acuerdo con una consulta;

recepción de la consulta, y en respuesta a ella, obtener tanto los objetos tipo archivo como los objetos diferentes a los archivos que corresponden a la
consulta; y

20 presentar tanto los objetos tipo archivo como los objetos diferentes a los archivos que corresponden a la consulta en el despliegue.

117. Un método implementado por computador para permitir el uso de previsualizadores de terceros en un navegador de la *shell* que soporte múltiples tipos de objetos, el método comprende:

el suministro de un navegador de la *shell* con un previsualizador
5 predeterminado para múltiples tipos de objetos; y

el suministro de un mecanismo de ampliación que permite que terceros desarrollen un previsualizador alternativo para por lo menos uno de los múltiples tipos de objetos.

118. El método de la Reivindicación 101, donde se proporciona un filtro
10 en el despliegue que puede ser seleccionado por parte de un usuario, para filtrar los objetos con base en las propiedades de metadatos especificados por el filtro.

119. El método de la Reivindicación 101, donde se proporciona un enlace rápido en el despliegue, tal que pueda ser seleccionado por parte de un
15 usuario para visualizar una colección específica de objetos.

120. El método de la Reivindicación 101, donde se proporciona una librería que incluye una colección de objetos y un conjunto de herramientas para manipular los objetos de la librería.

121. El método de la Reivindicación 101, donde los objetos en una
20 carpeta virtual pueden ser físicamente almacenados en memorias en diferentes localizaciones físicas.

122. El método de la Reivindicación 121, donde las diferentes localizaciones físicas comprenden el computador actual y por lo menos otro computador diferente, una localización sobre una red, y un dispositivo de almacenamiento externo.

5 123. El método de la Reivindicación 101, donde los objetos comprenden tanto objetos de archivos como objetos diferentes a archivos.

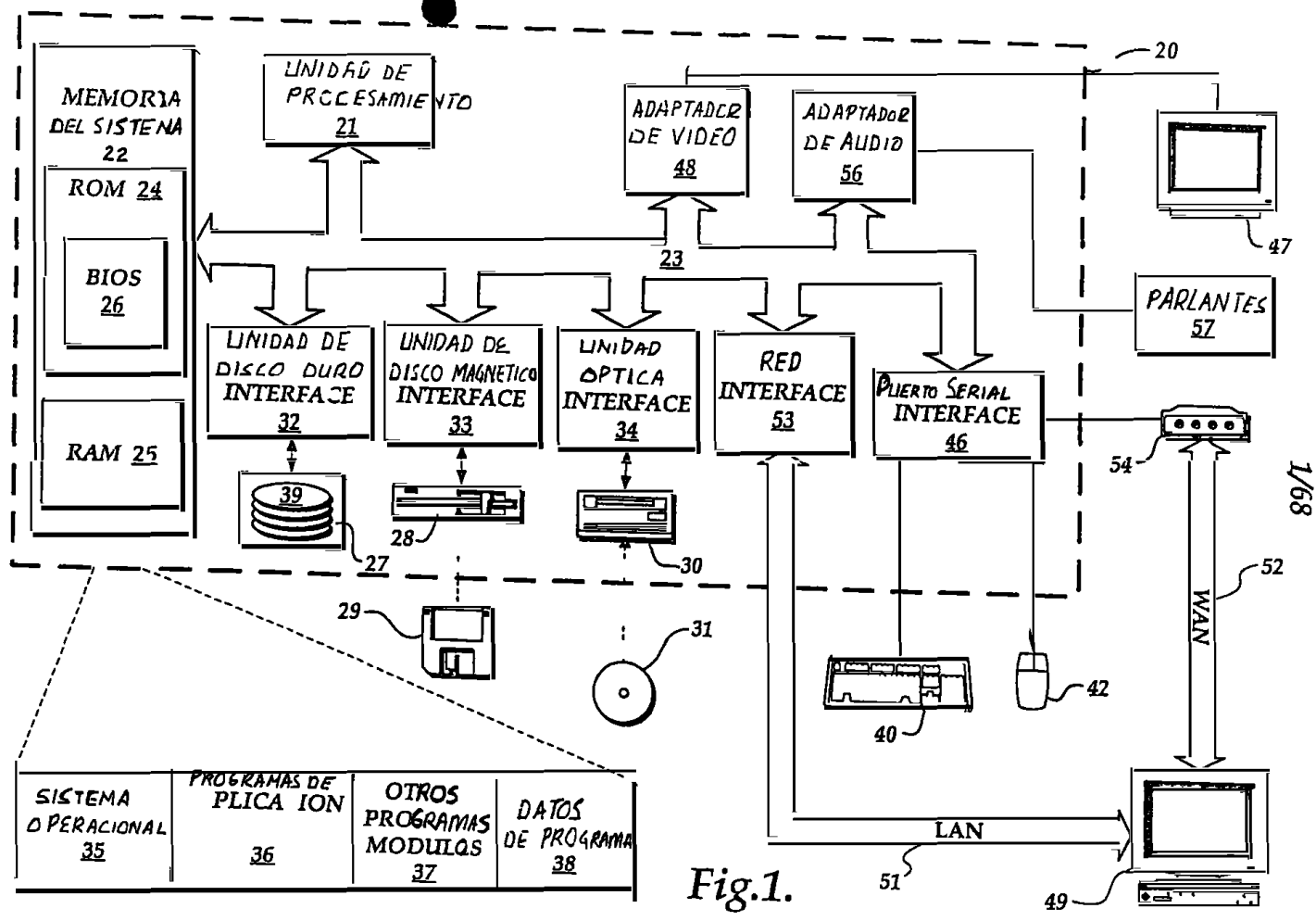
124. El método de la Reivindicación 123, donde los objetos diferentes a archivos comprenden por lo menos uno entre contactos o correos electrónicos.

RESUMEN DE LA INVENCION

10 Se proporciona un shell del sistema de archivos (N.T. el "shell" se refiere al espacio donde se manejan las instrucciones a ejecutar, a nivel del sistema operativo, y en este caso, al navegador que permite recorrer el sistema de archivos). Uno de los aspectos del *shell* consiste en que proporciona carpetas virtuales que exponen archivos comunes y carpetas ante los usuarios en
15 diferentes vistas, con base en sus metadatos en lugar de basarse en la estructura física actual del sistema de archivos subyacente en el disco. Los usuarios pueden trabajar con carpetas virtuales a través de manipulación directa, (e.g. selección y arrastre, copiado, pegado, etc.). Se suministran filtros para reducir los conjuntos de objetos. Se proporcionan enlaces rápidos, los
20 cuales se pueden seleccionar para general vistas útiles de los conjuntos de objetos. Se suministran librerías que consisten en grandes grupos de tipos de objetos utilizables que pueden asociarse en grupos, junto con funciones y

herramientas relacionadas con estos objetos. Se proporciona una barra de direcciones virtuales que comprende una pluralidad de segmentos, cada segmento corresponde a un filtro para la selección de contenido. Un navegador de la *shell* se suministra, con el cual los usuarios pueden identificar fácilmente

5 un objeto con base en los metadatos asociados con tal objeto. Se proporciona un previsualizador de objetos en el navegador de la *shell* , el cual está configurado para desplegar o representar una pluralidad de objetos que representan múltiples tipos de objetos.



178

177

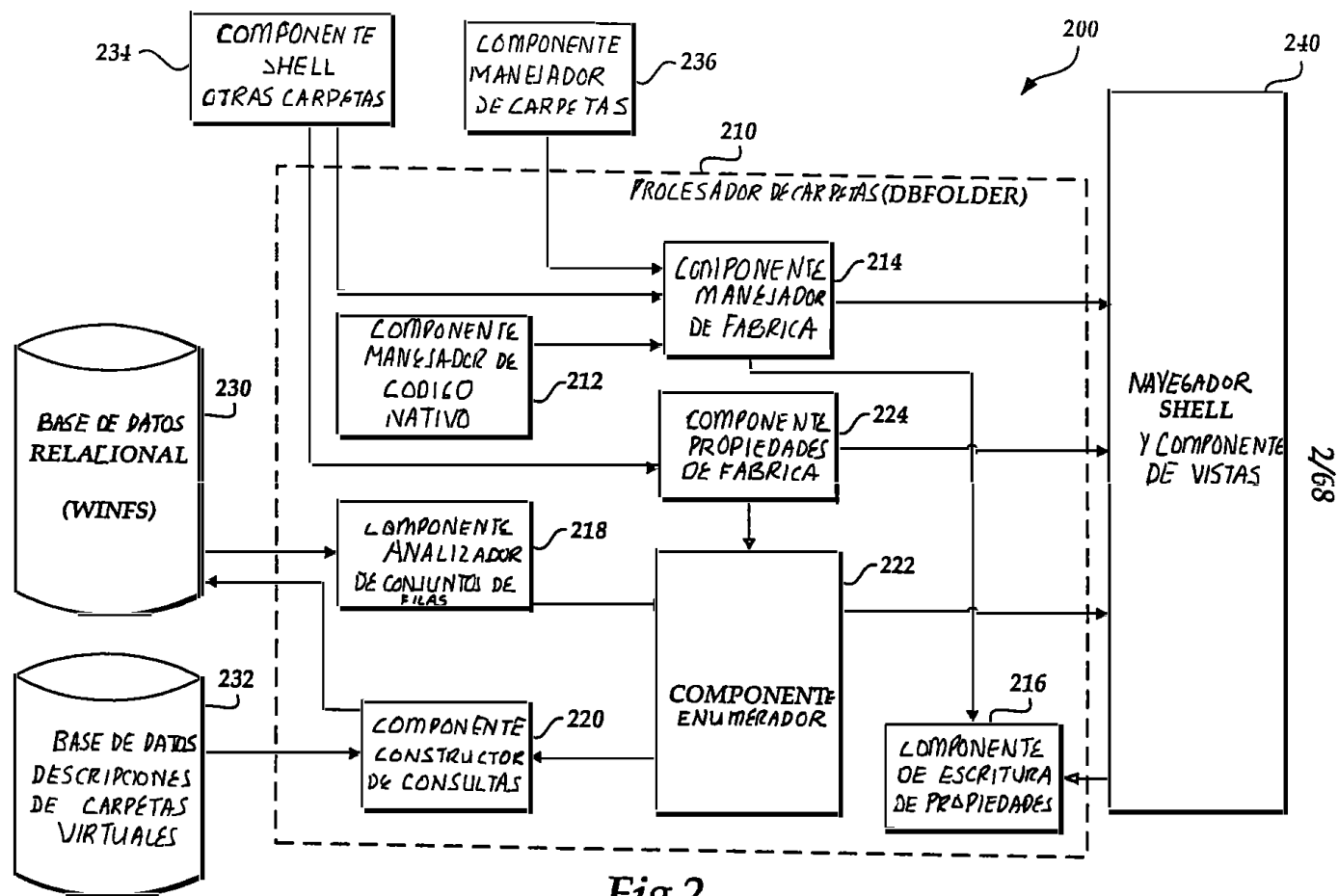


Fig.2.

3/68

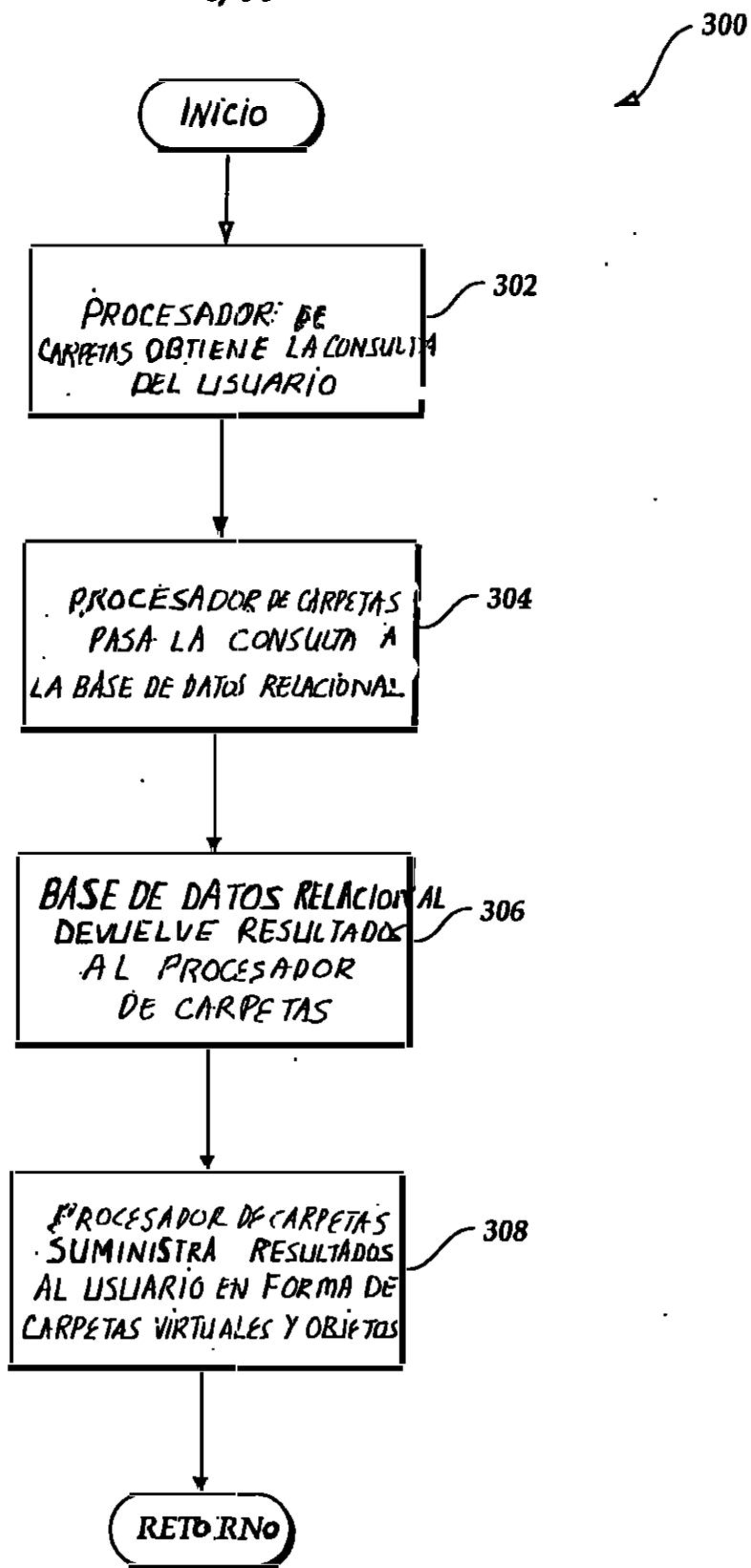


Fig.3.

4/68

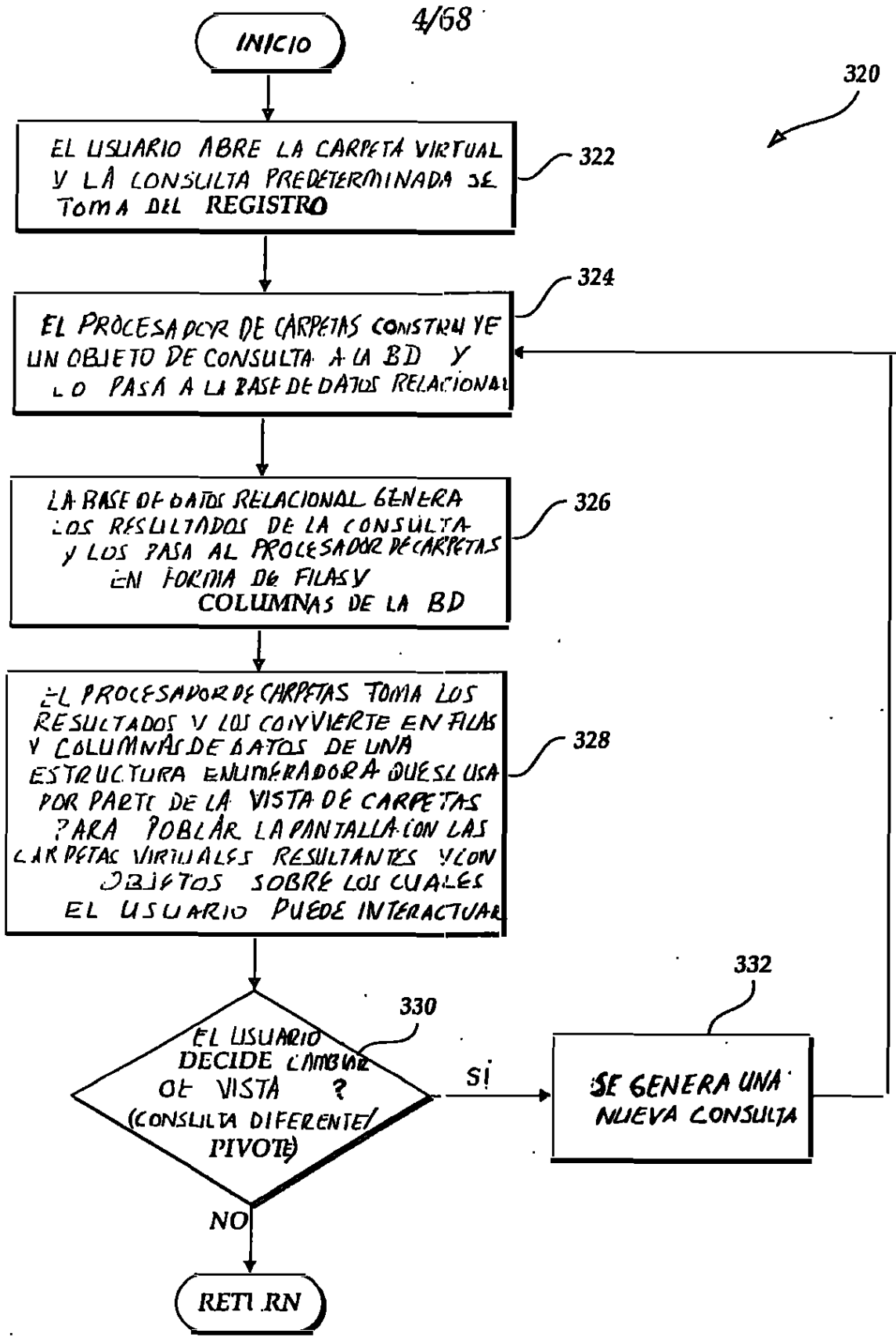


Fig.4.

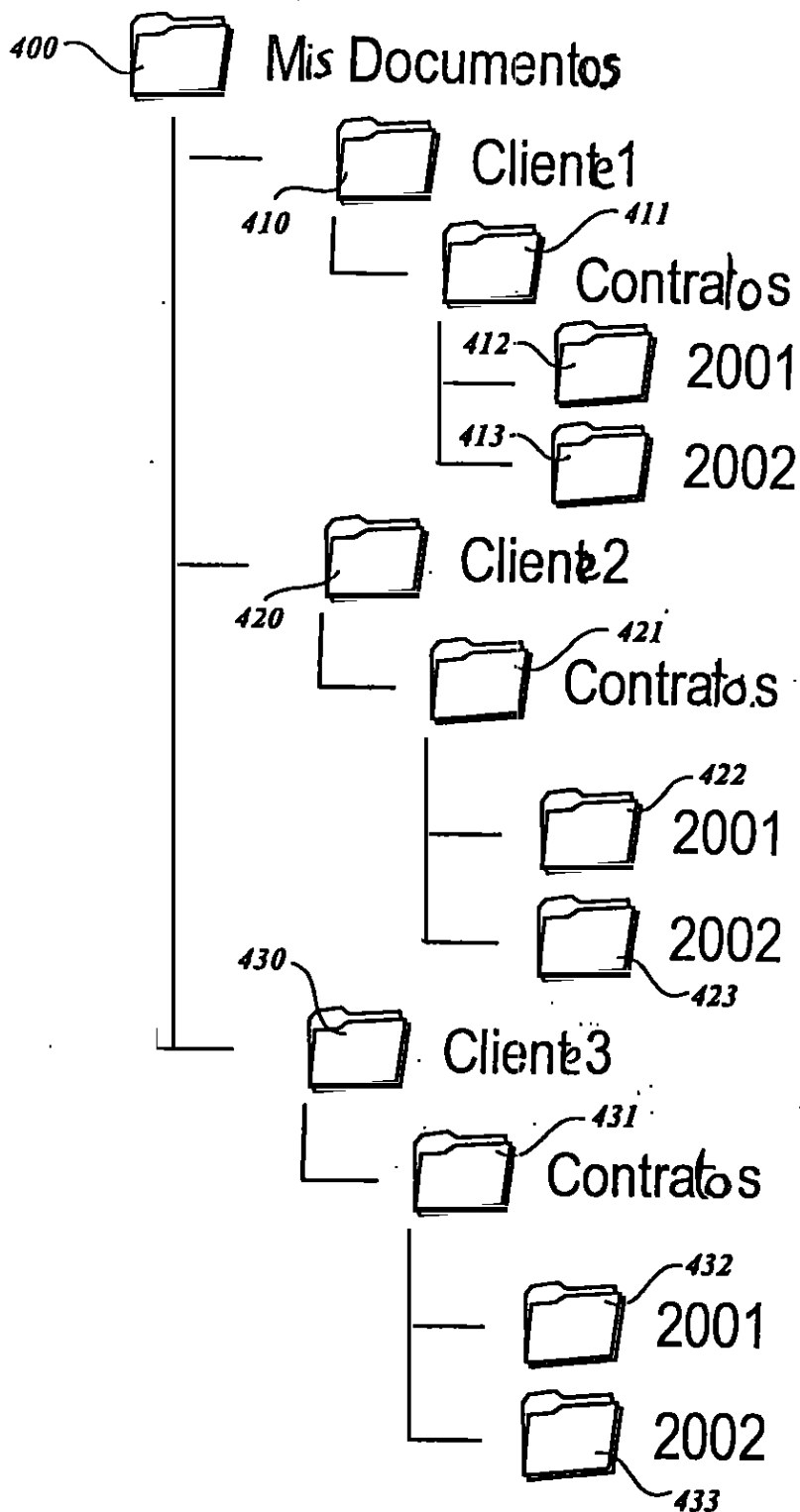


Fig.5.

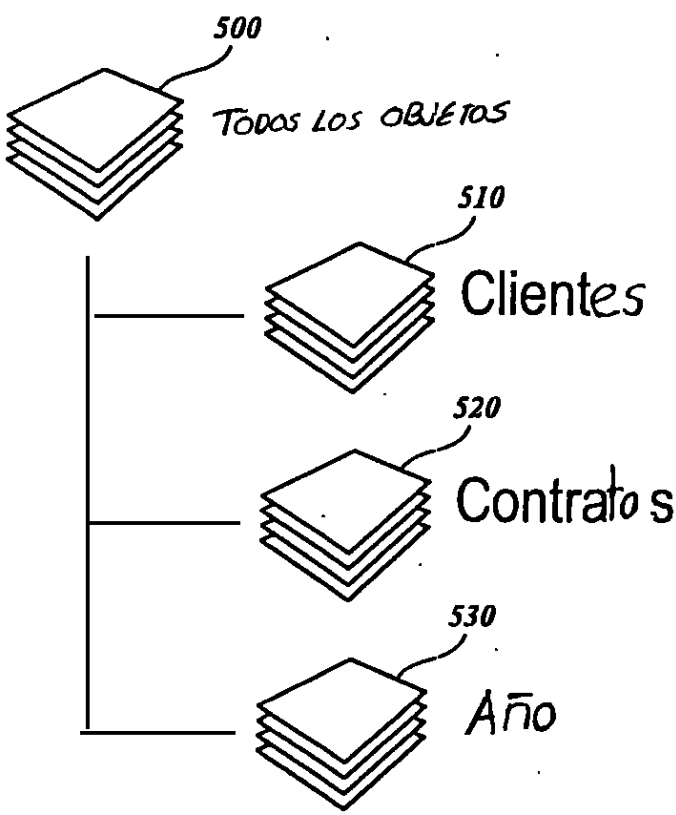


Fig.6.

7/68

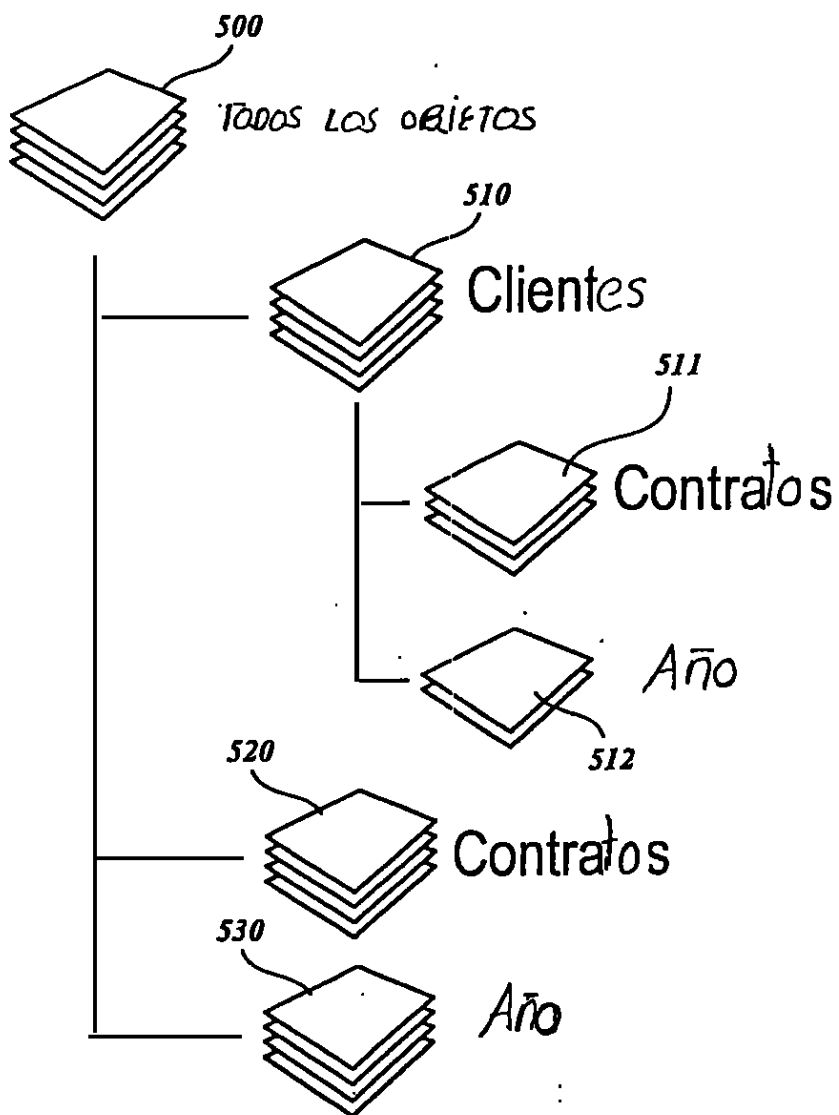


Fig.7.

8/68

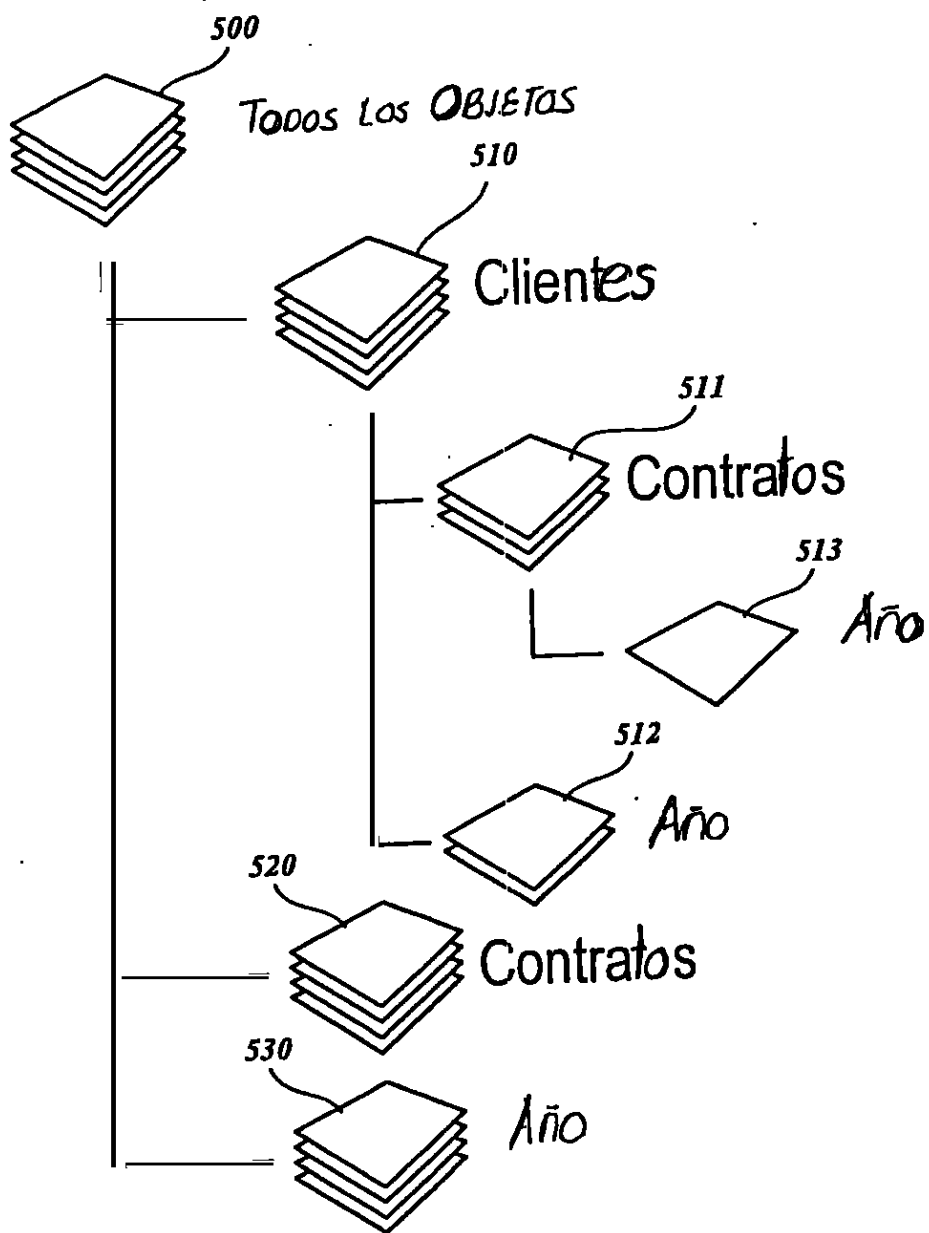


Fig.8.

9/68

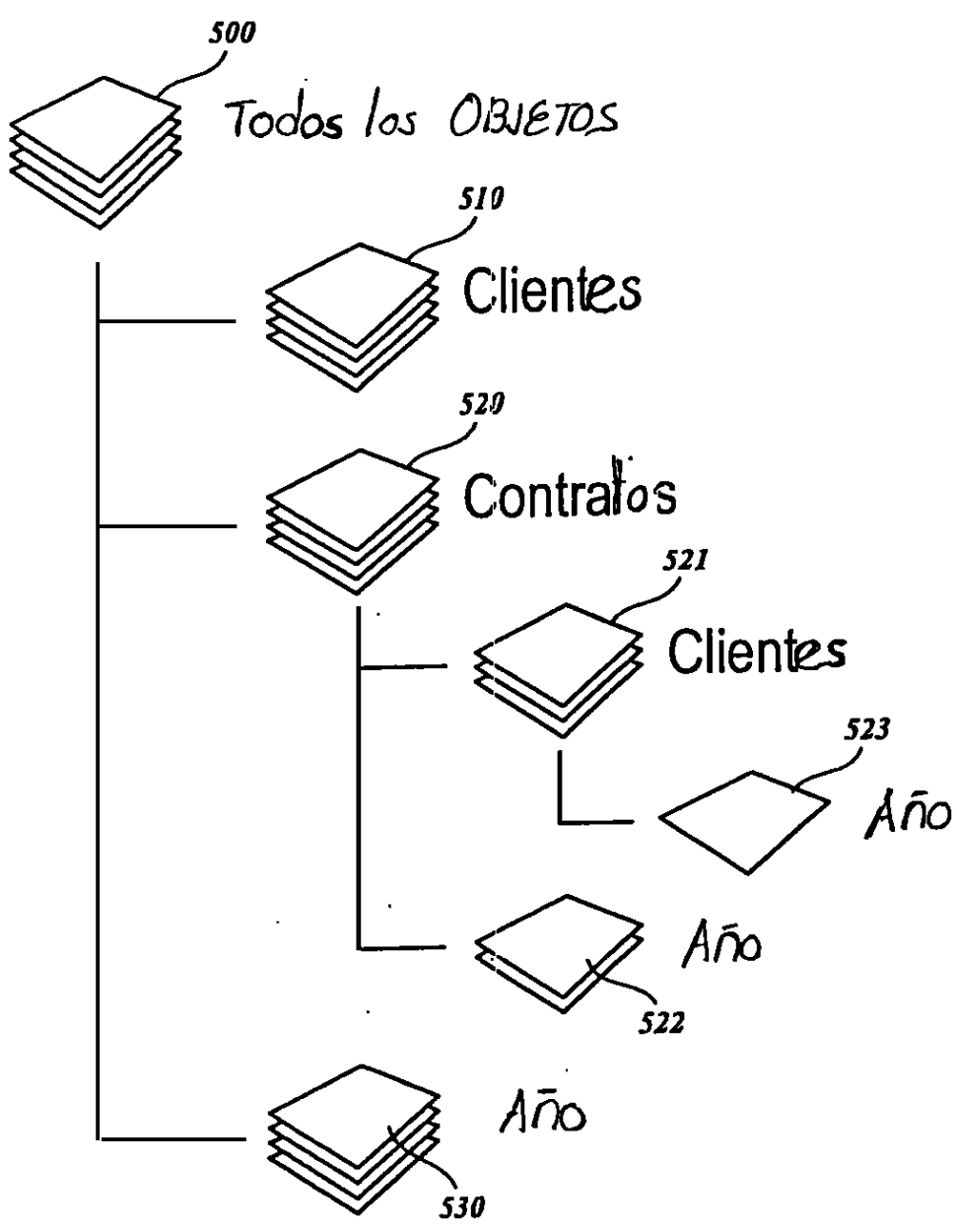


Fig.9.

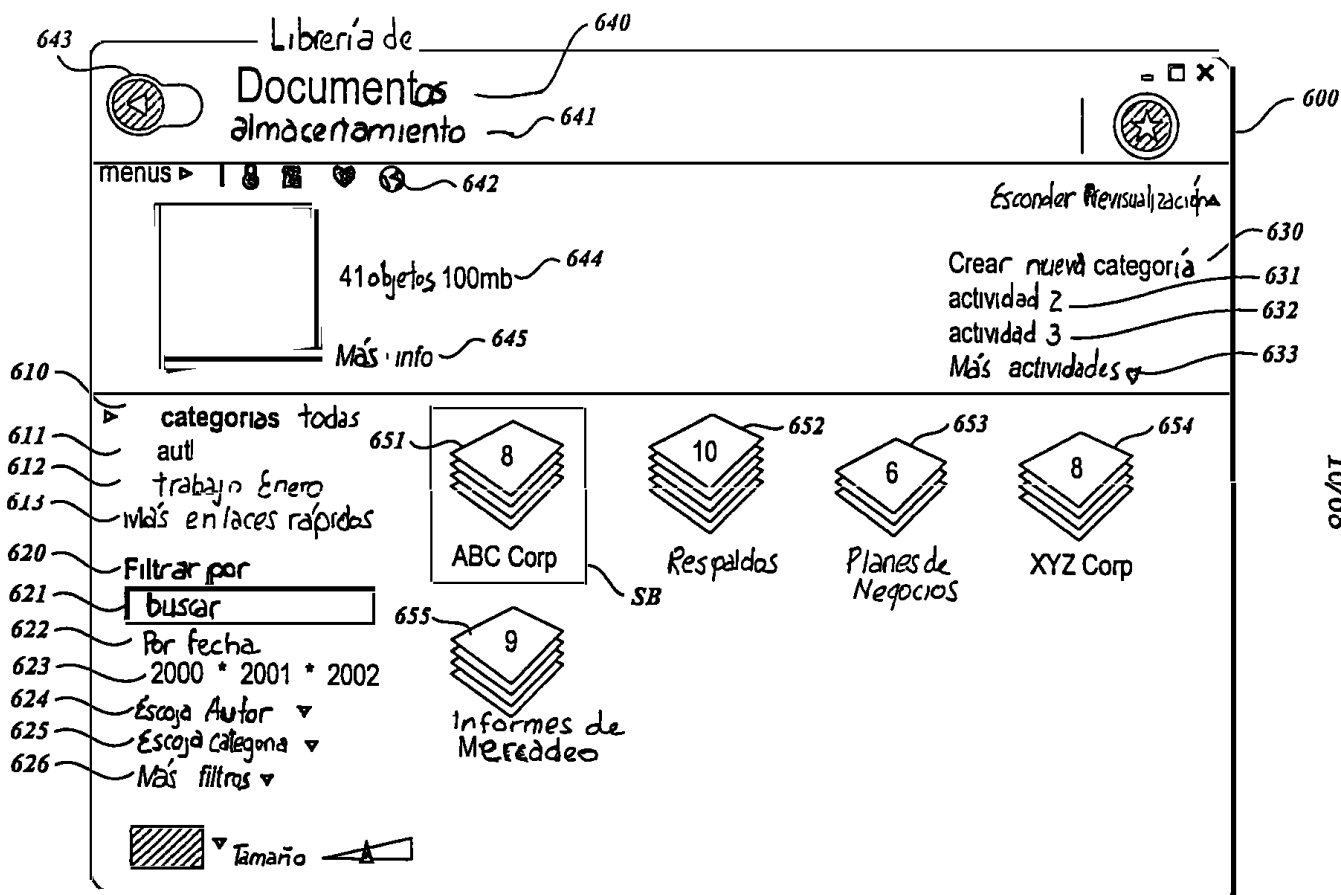


Fig.10.

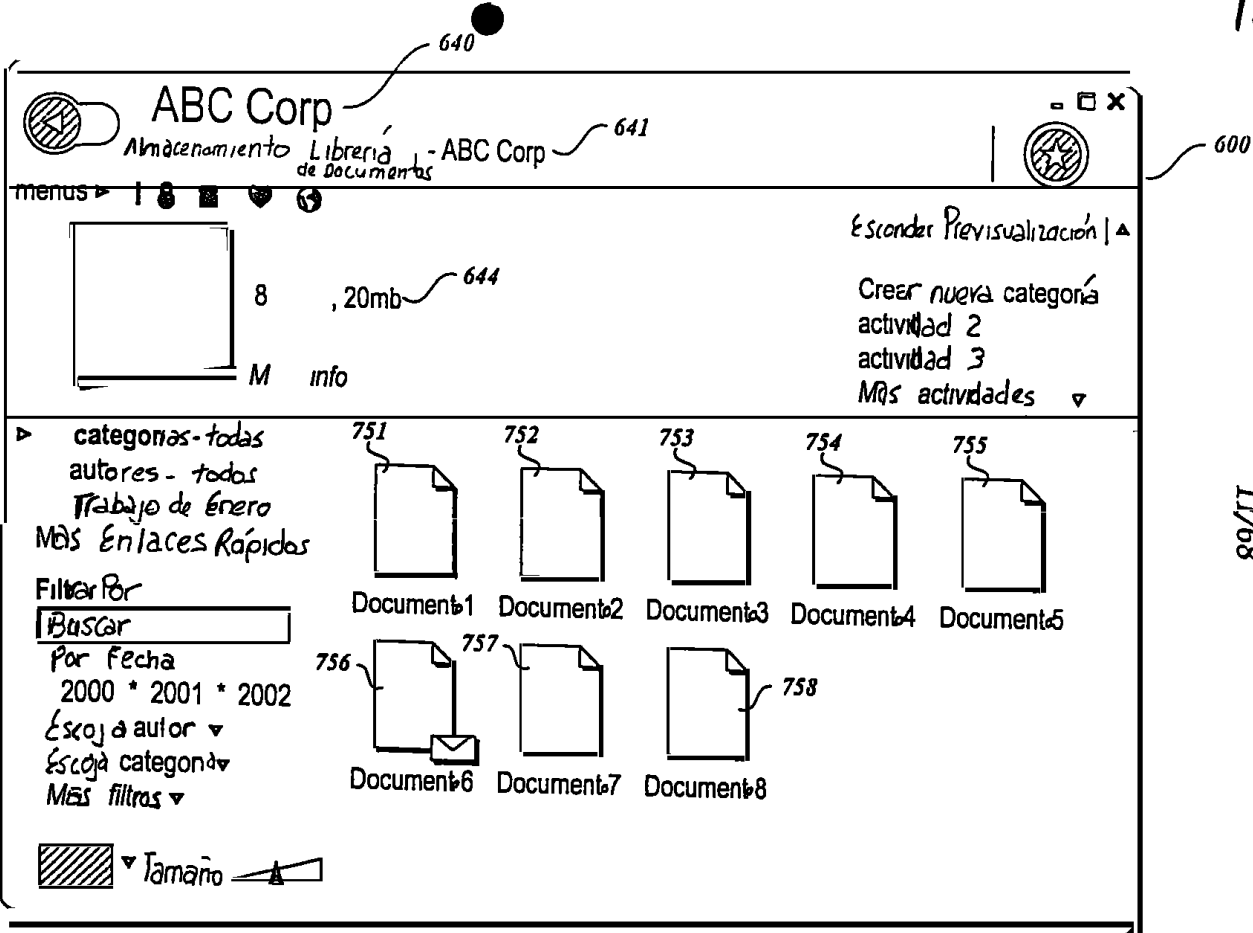


Fig.11.

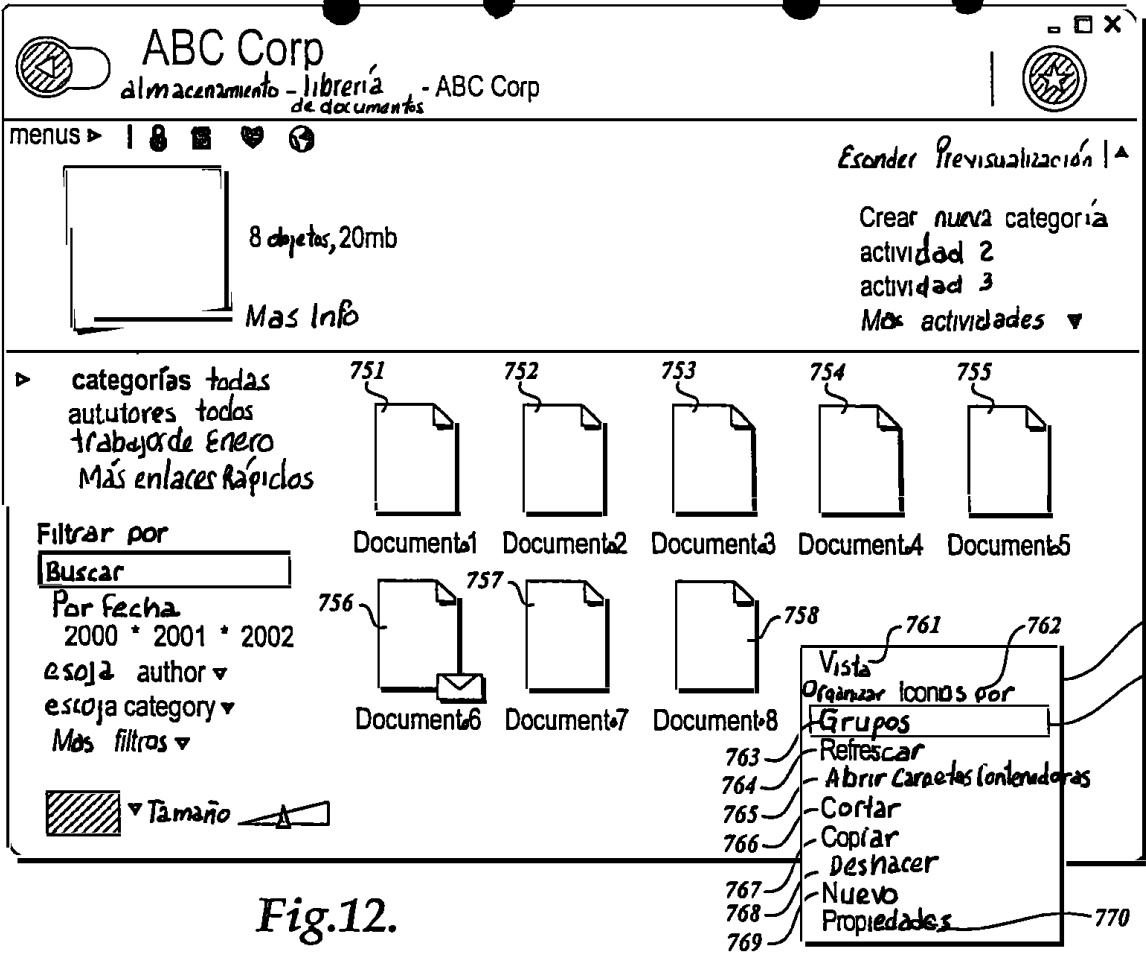
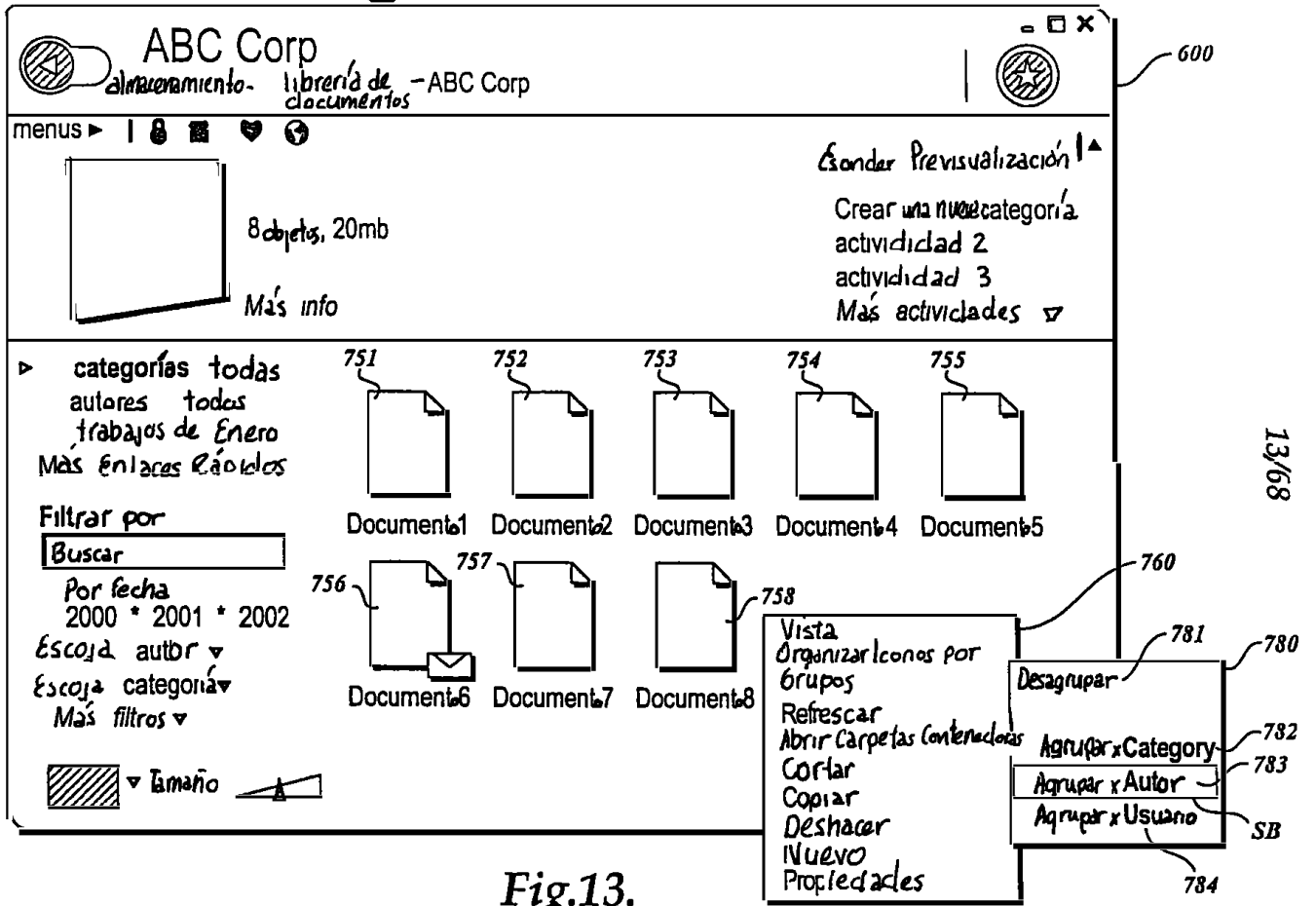
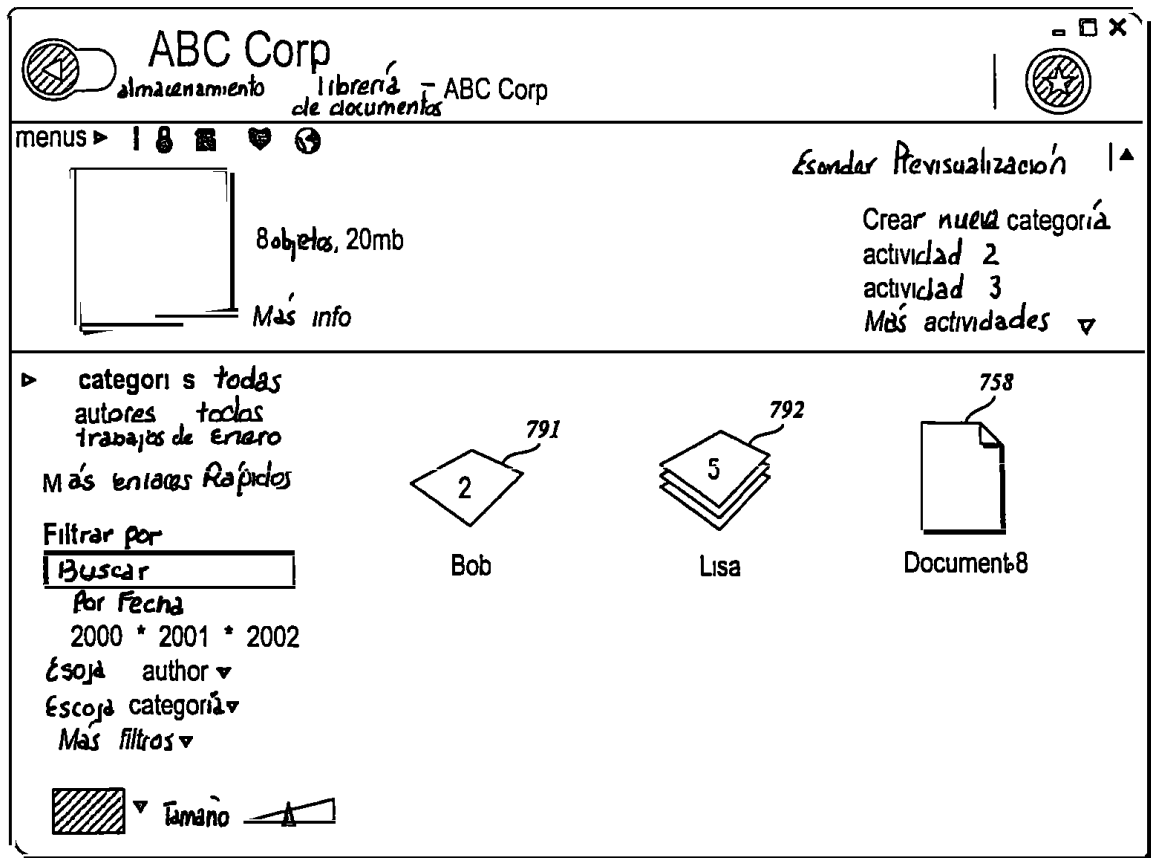


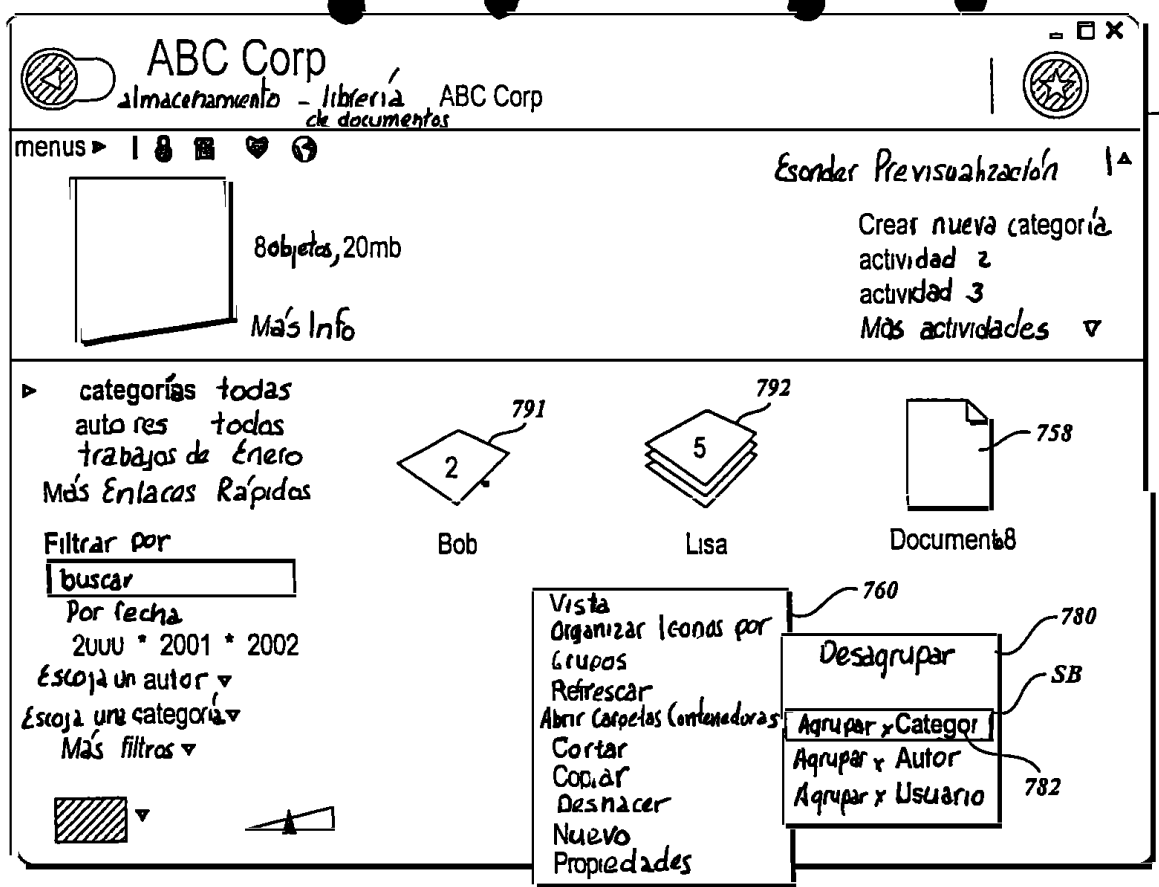
Fig.12.





14/68

Fig.14.



600

15/68

Fig.15.

191

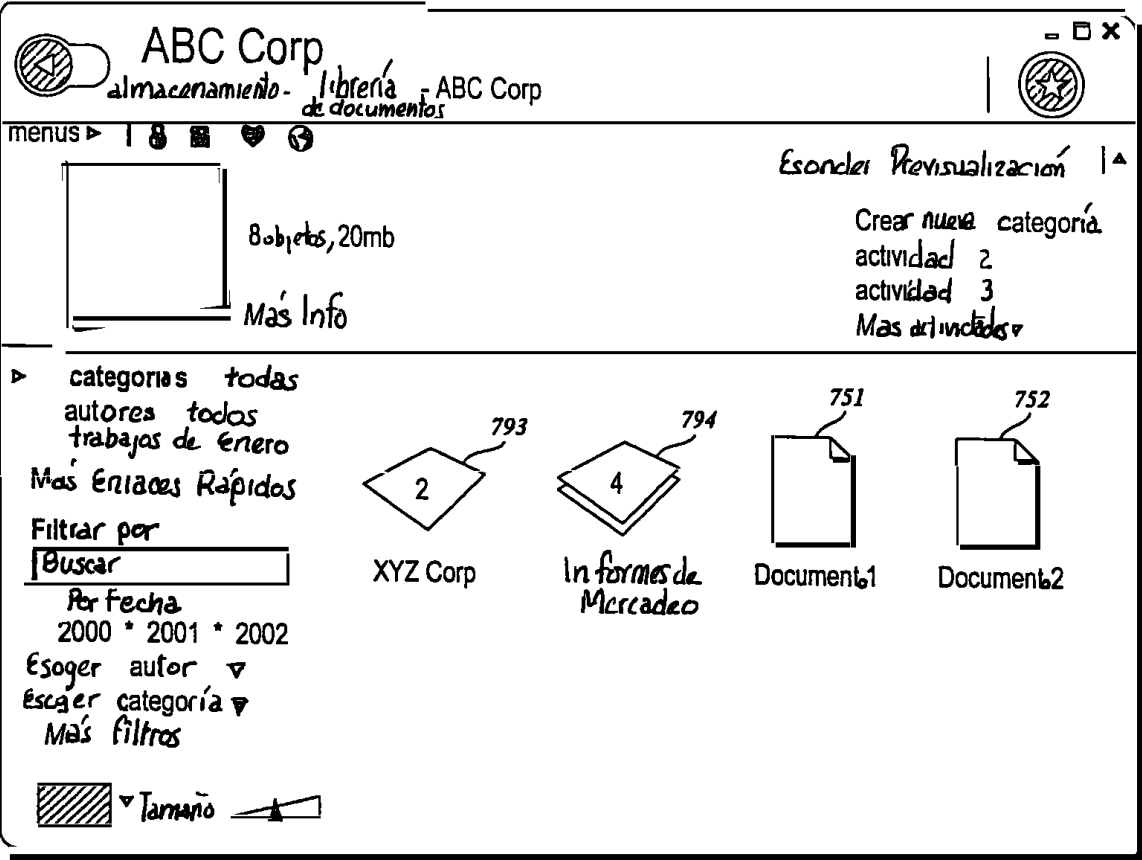


Fig.16.

192

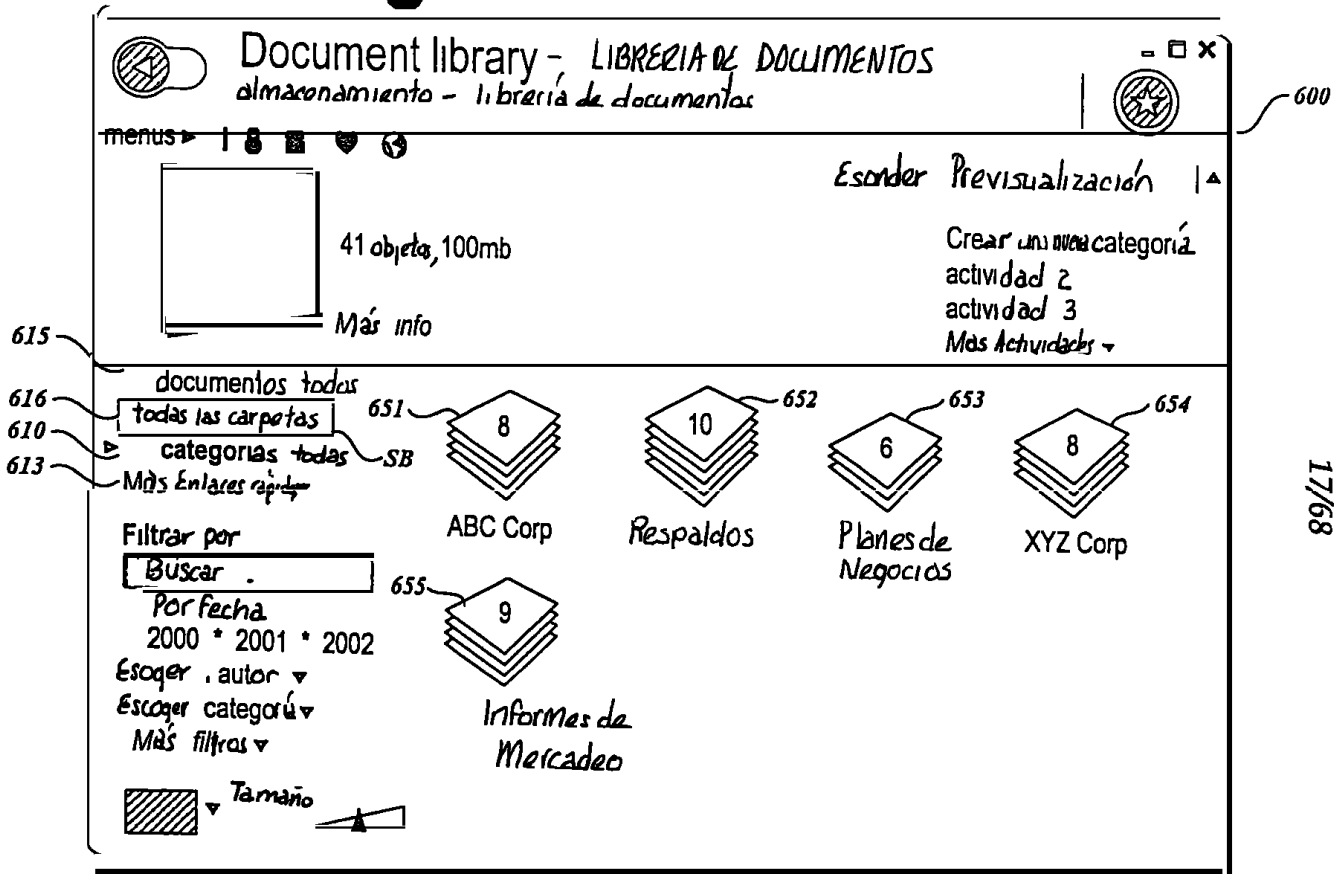


Fig.17.

17/68

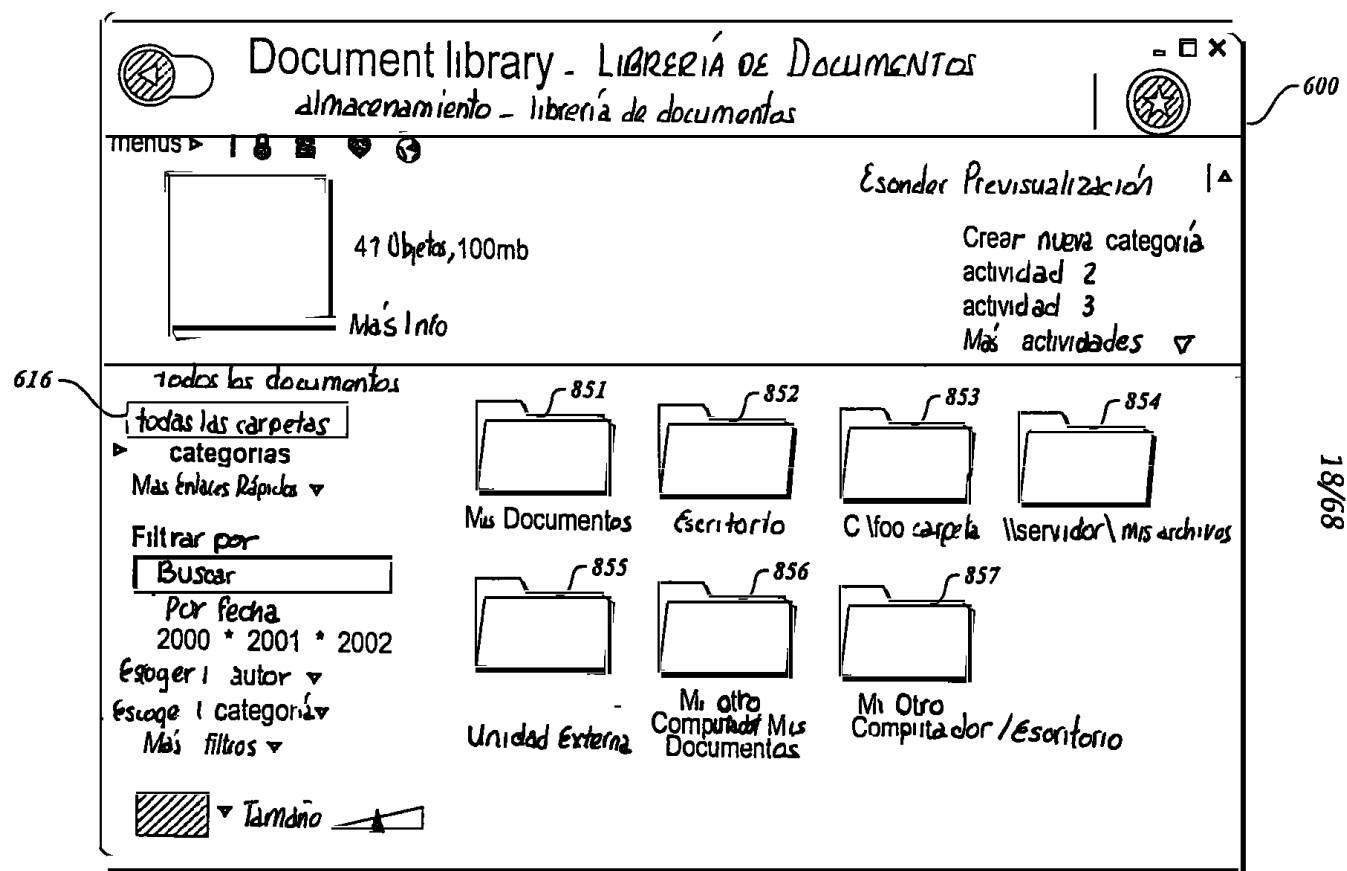


Fig.18.

194

19/68

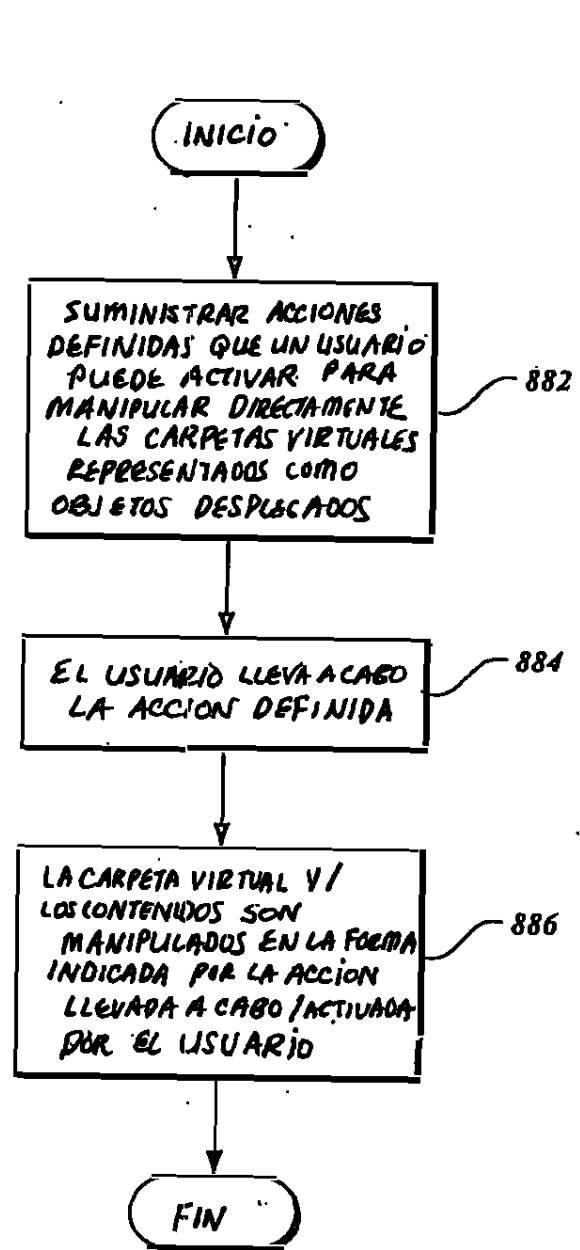


Fig.19.

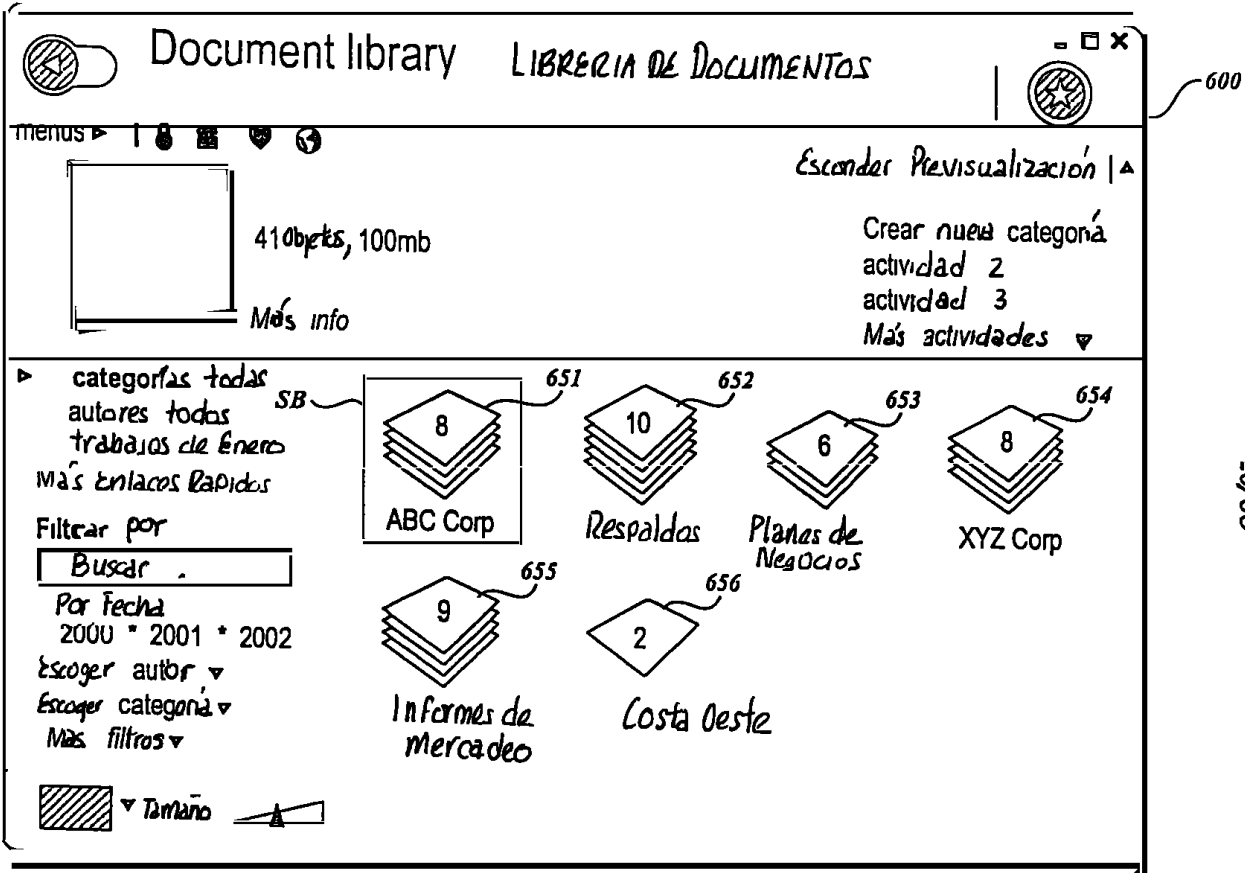


Fig.20.

196

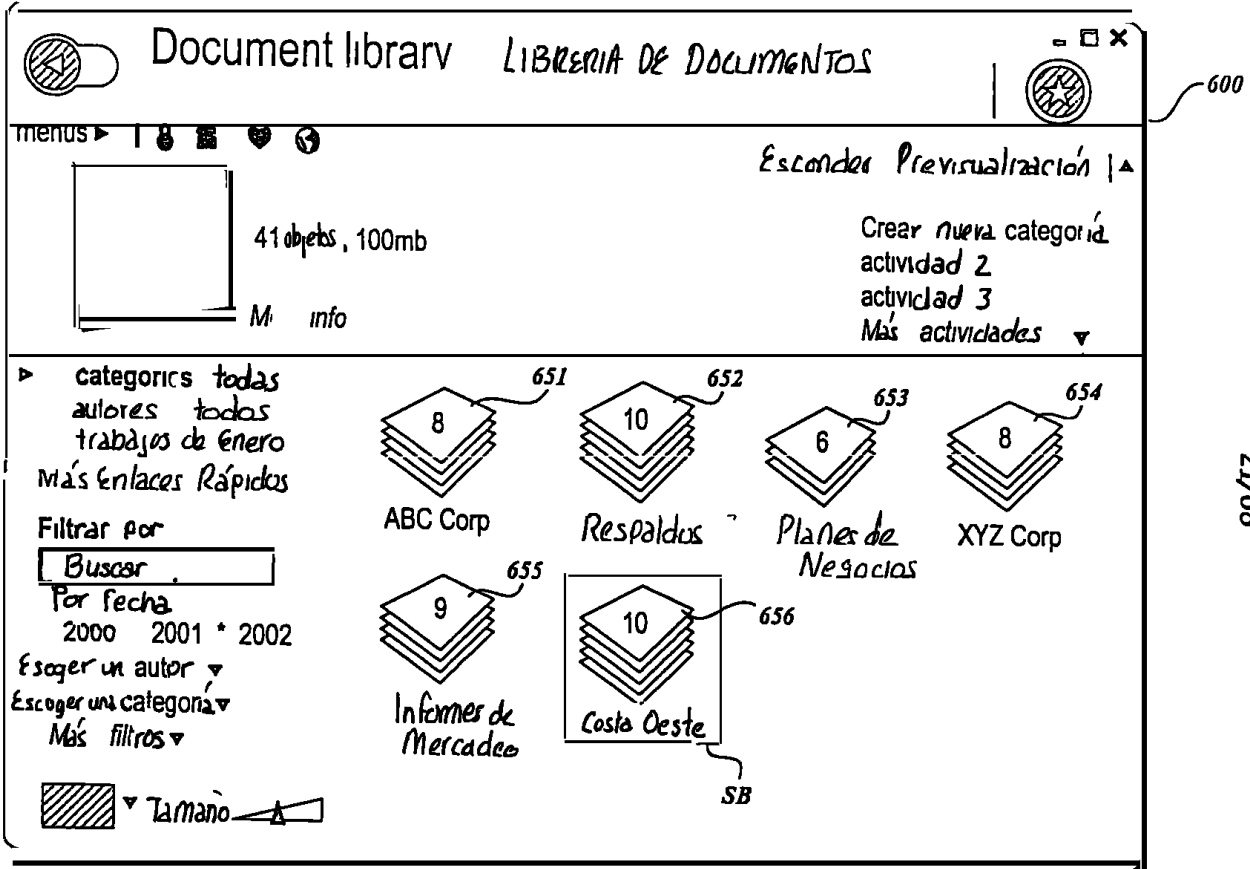


Fig.21.

22/68

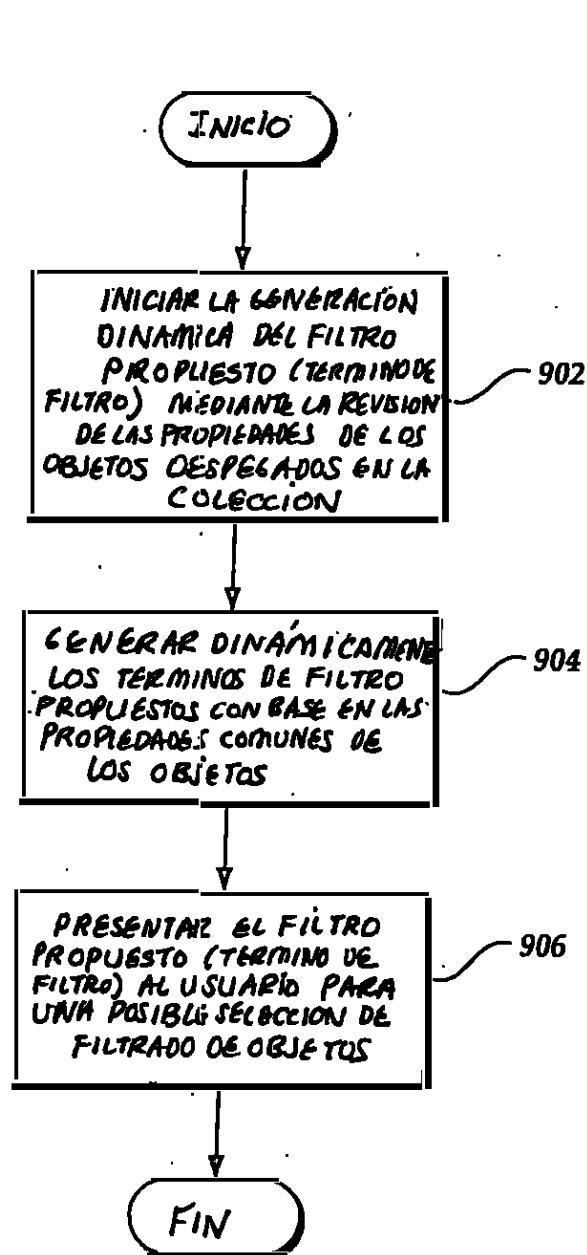


Fig.22.

23/68

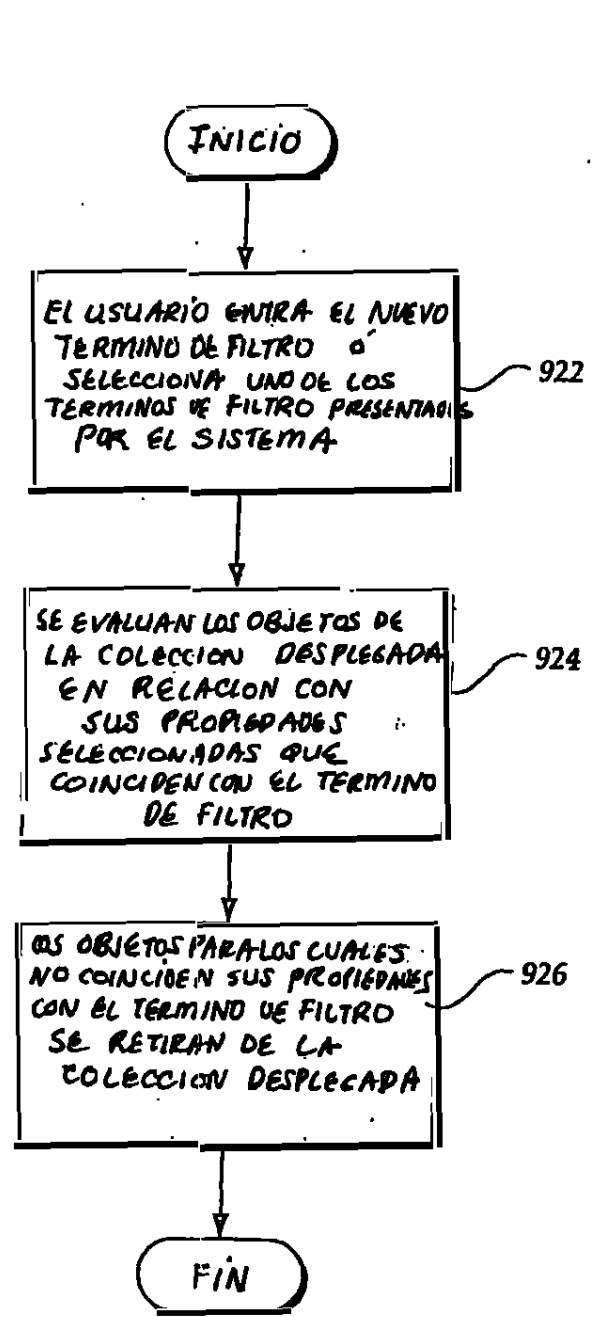
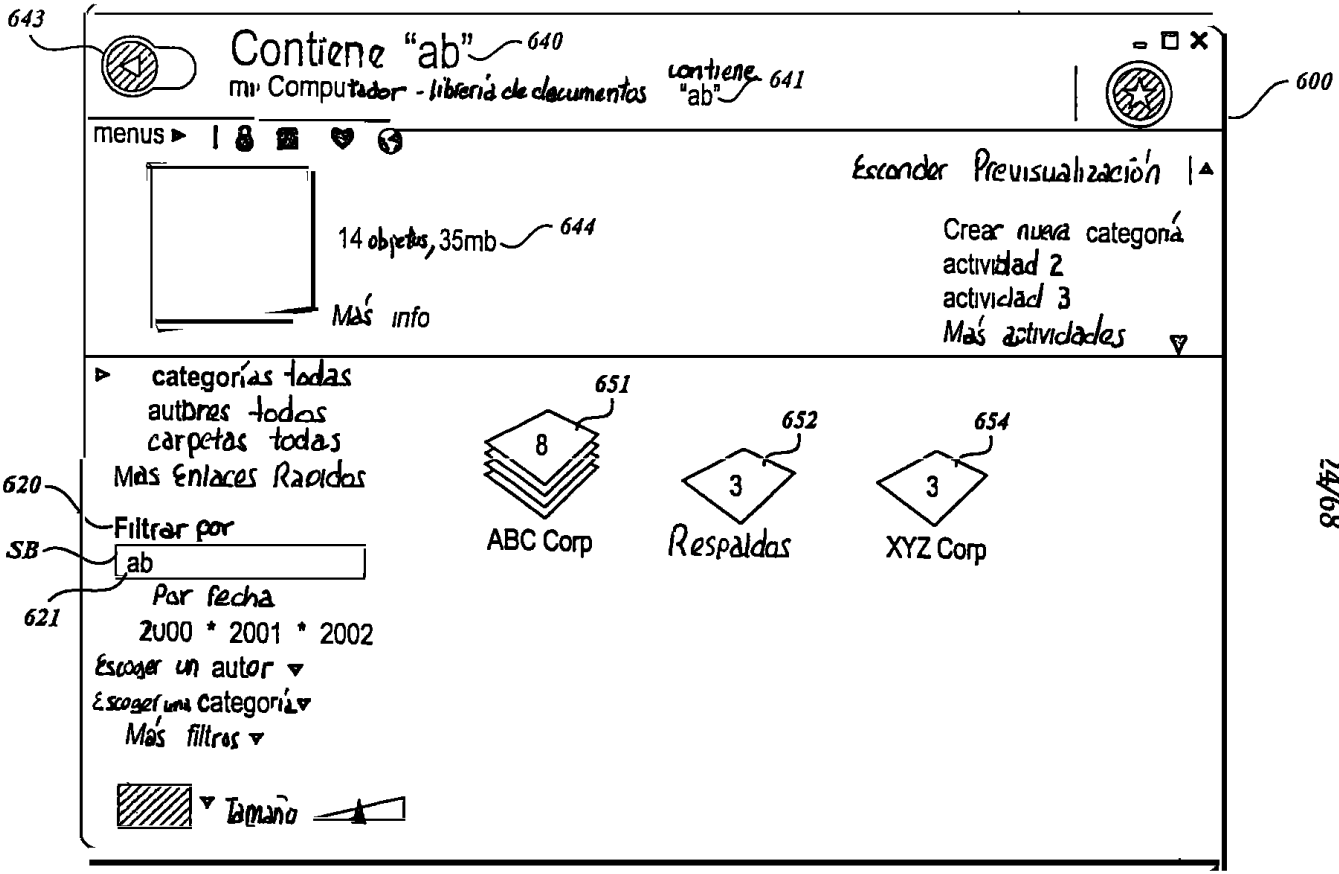


Fig.23.



24/68

Fig.24.

200

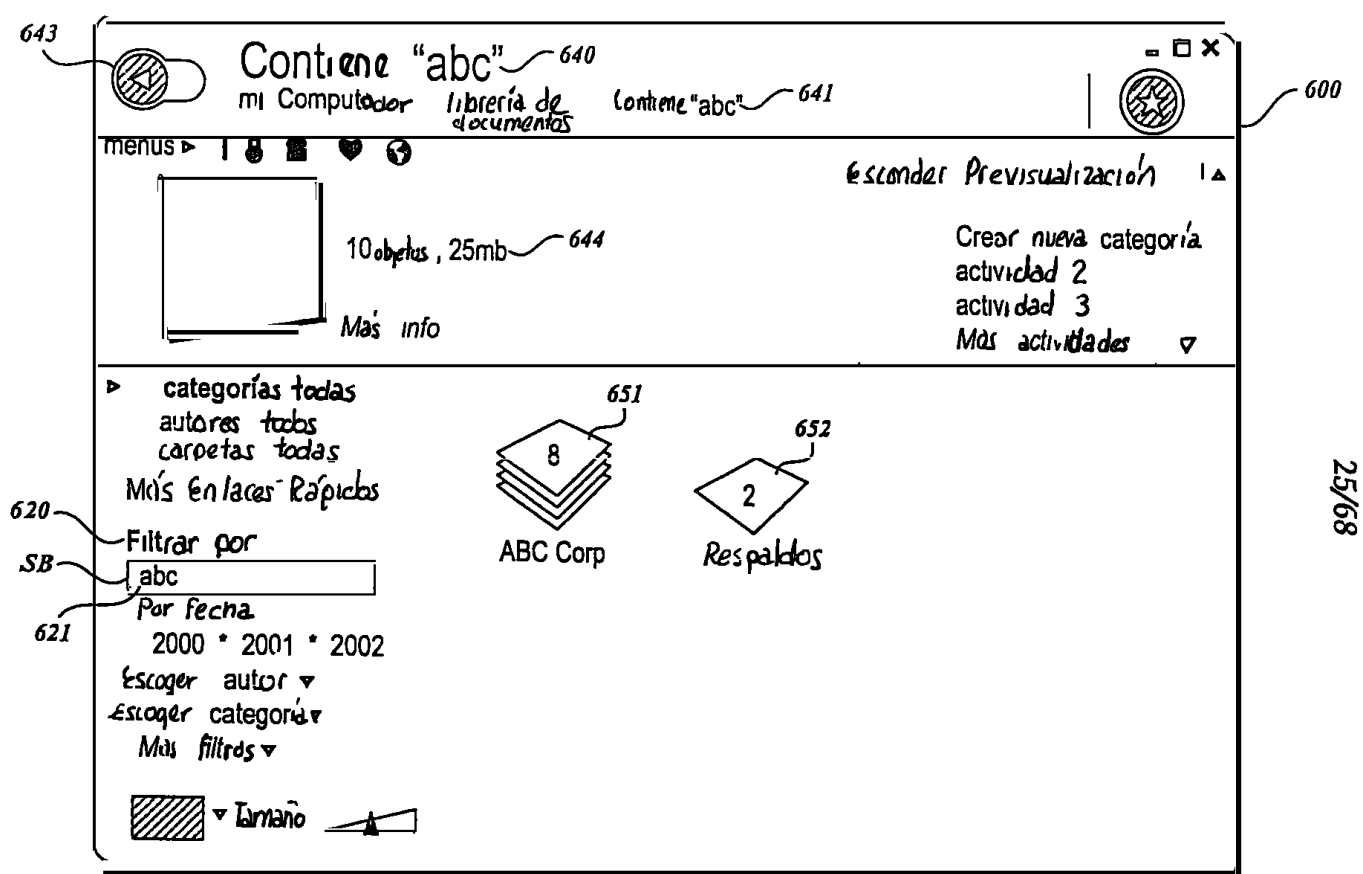


Fig.25.

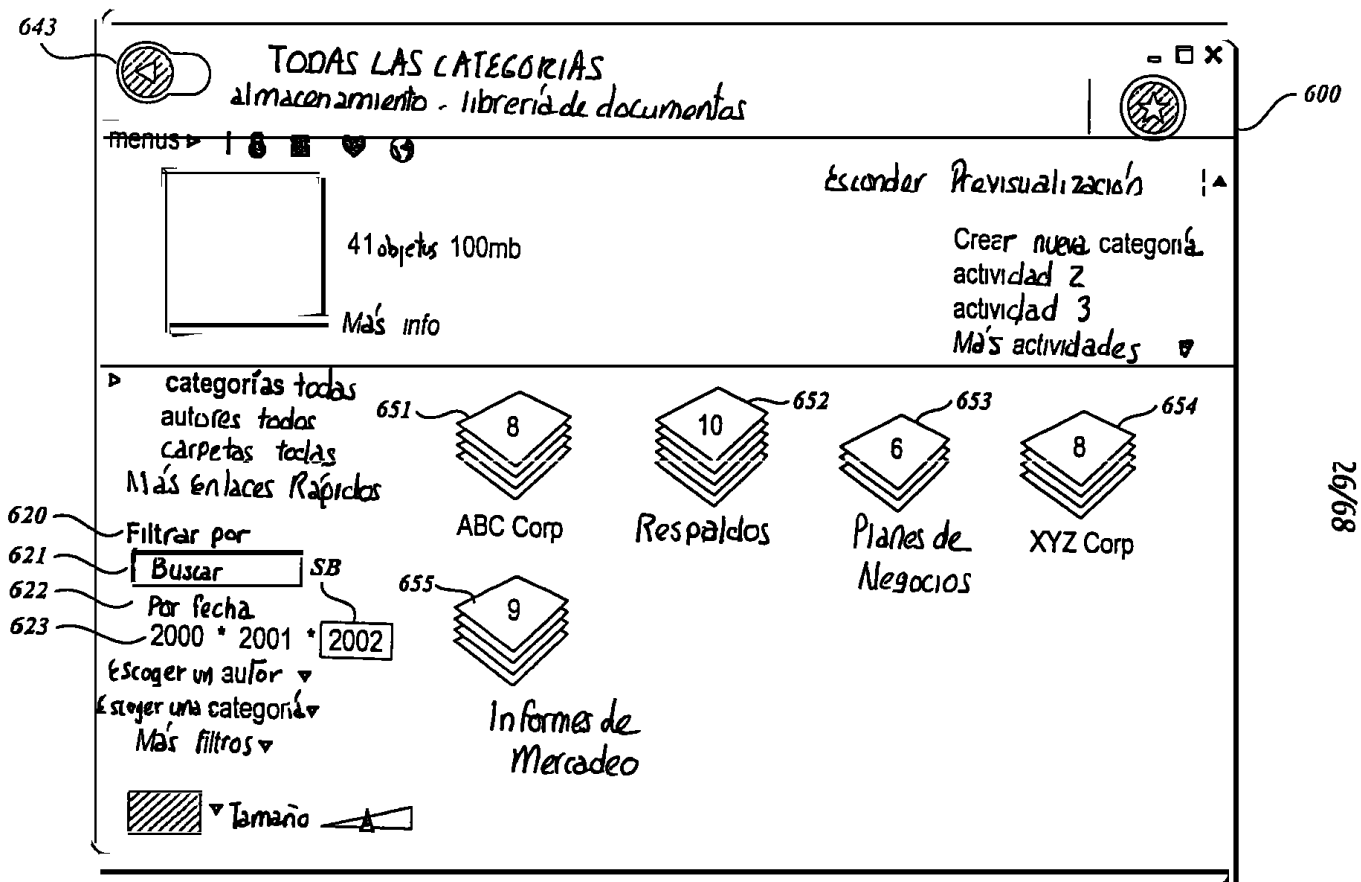


Fig.26.

202

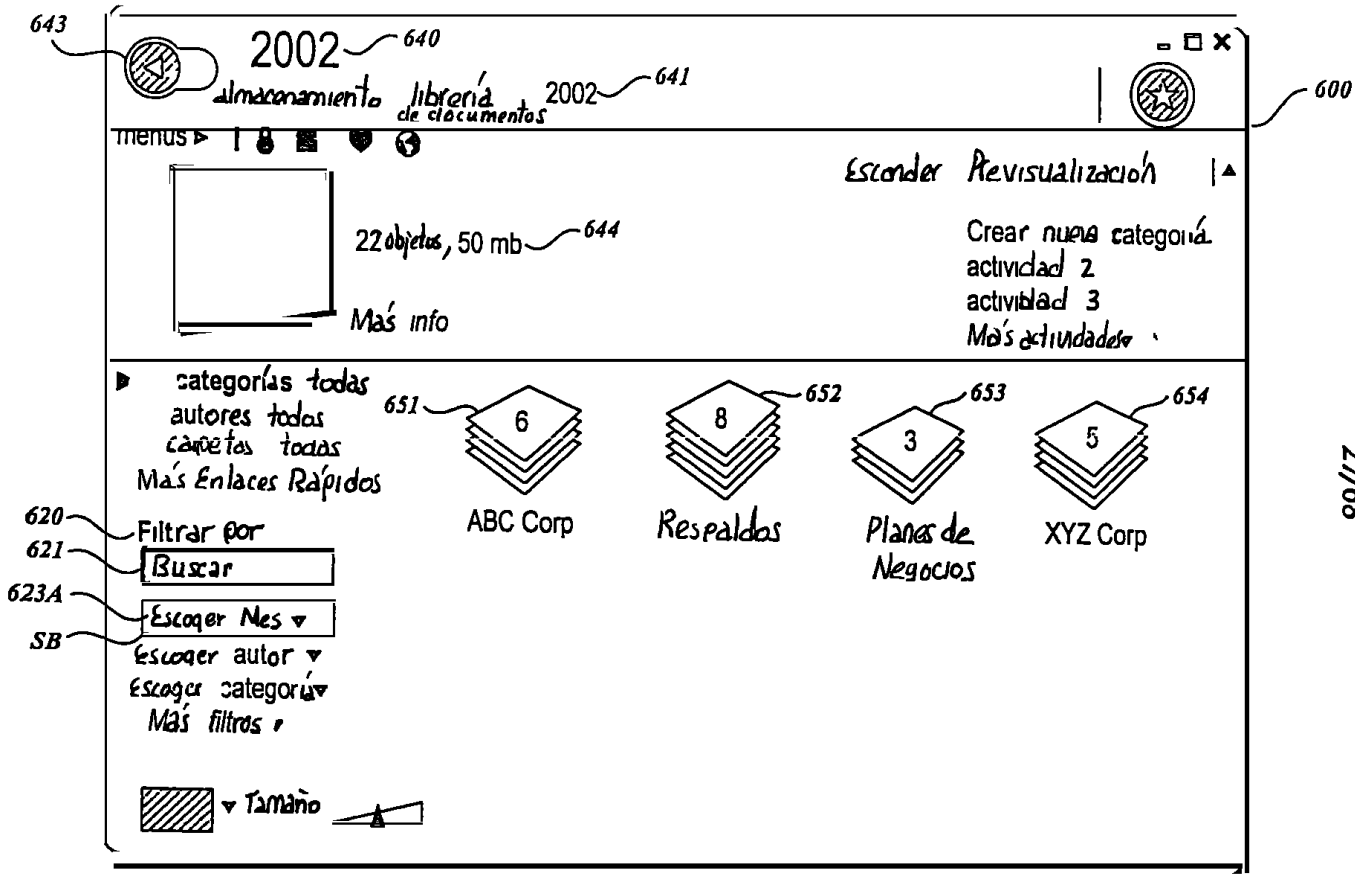


Fig.27.

203

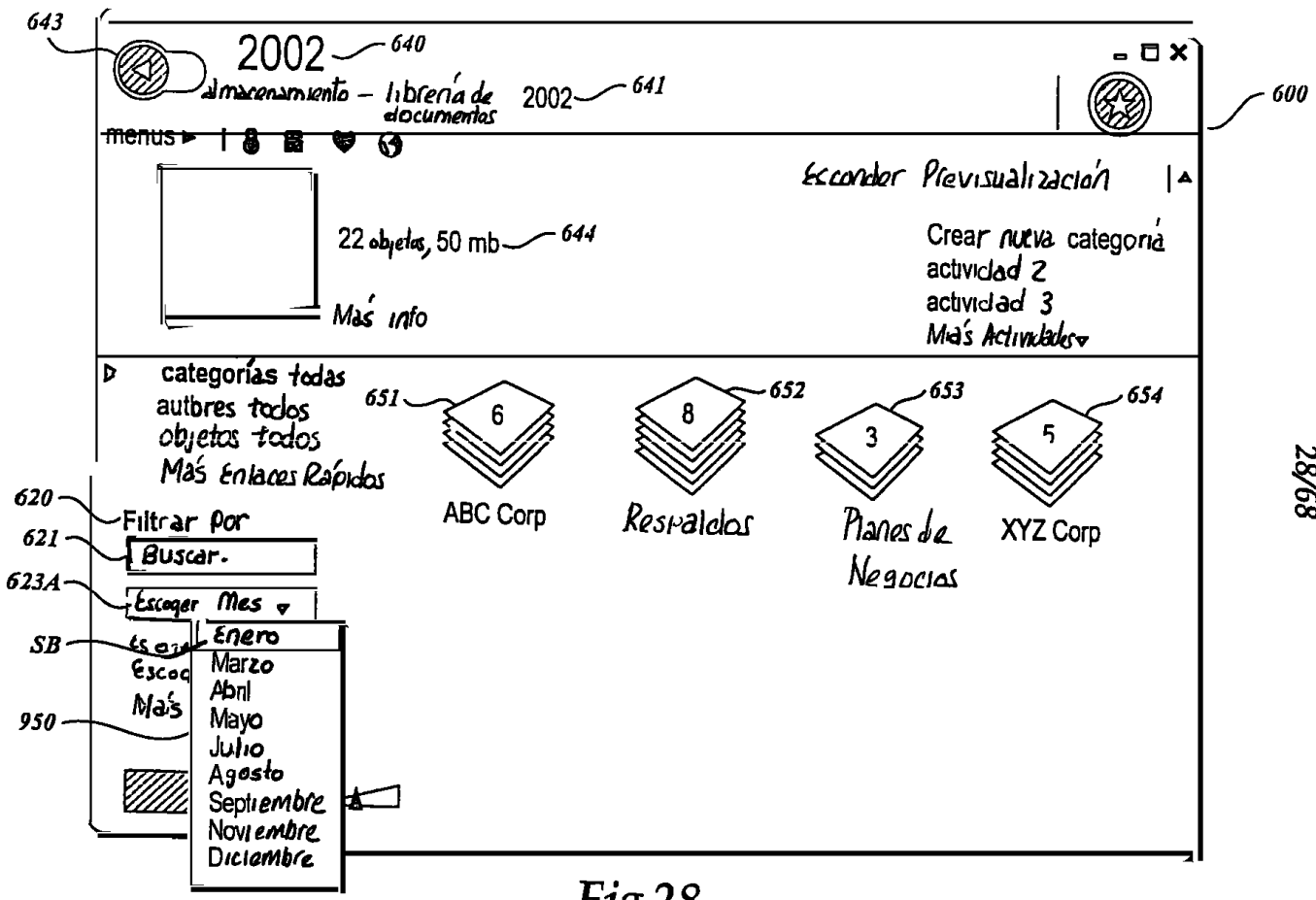


Fig.28.

28/68

204

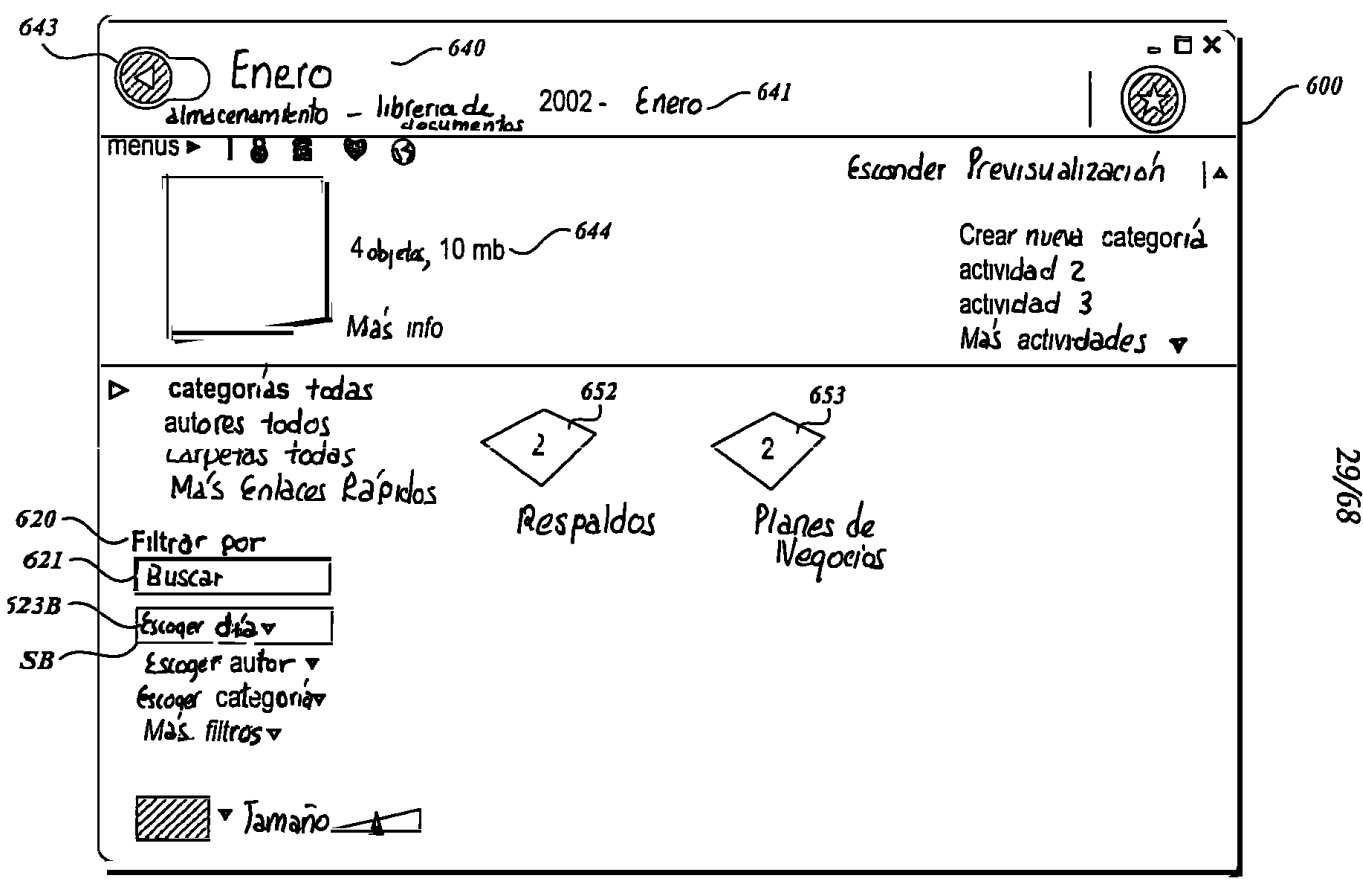


Fig.29.

29/68

405

30/68

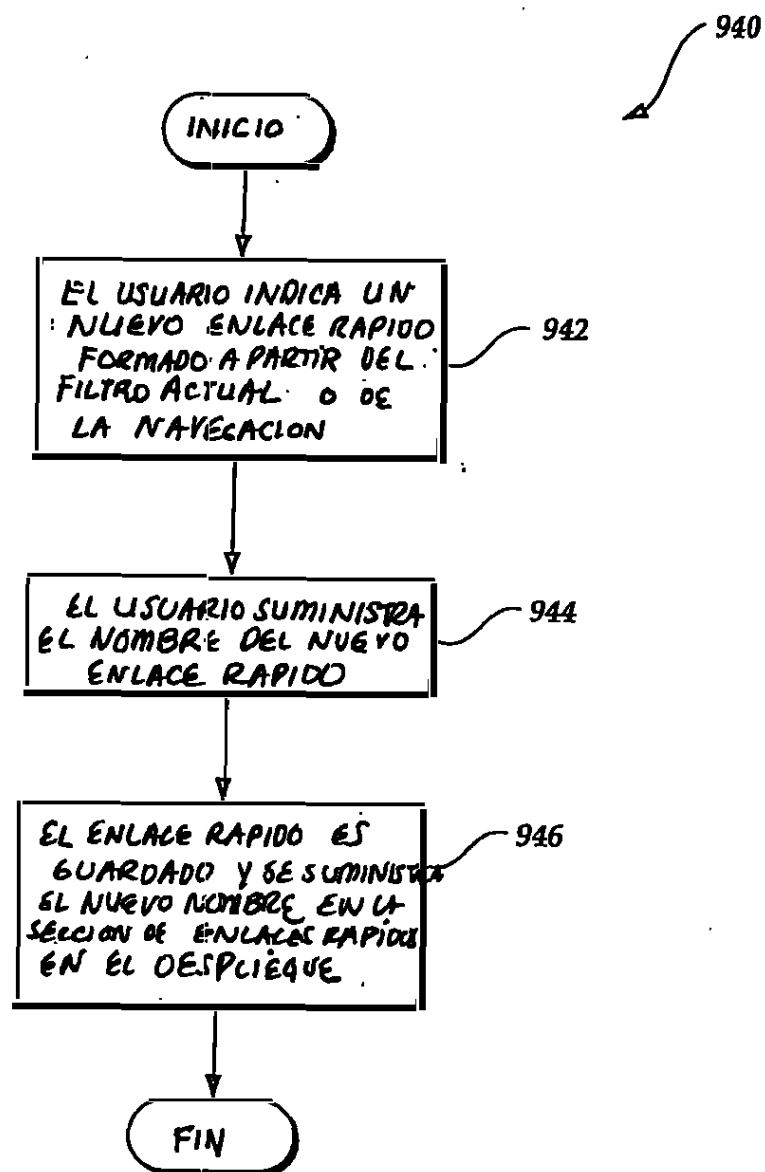


Fig.30.

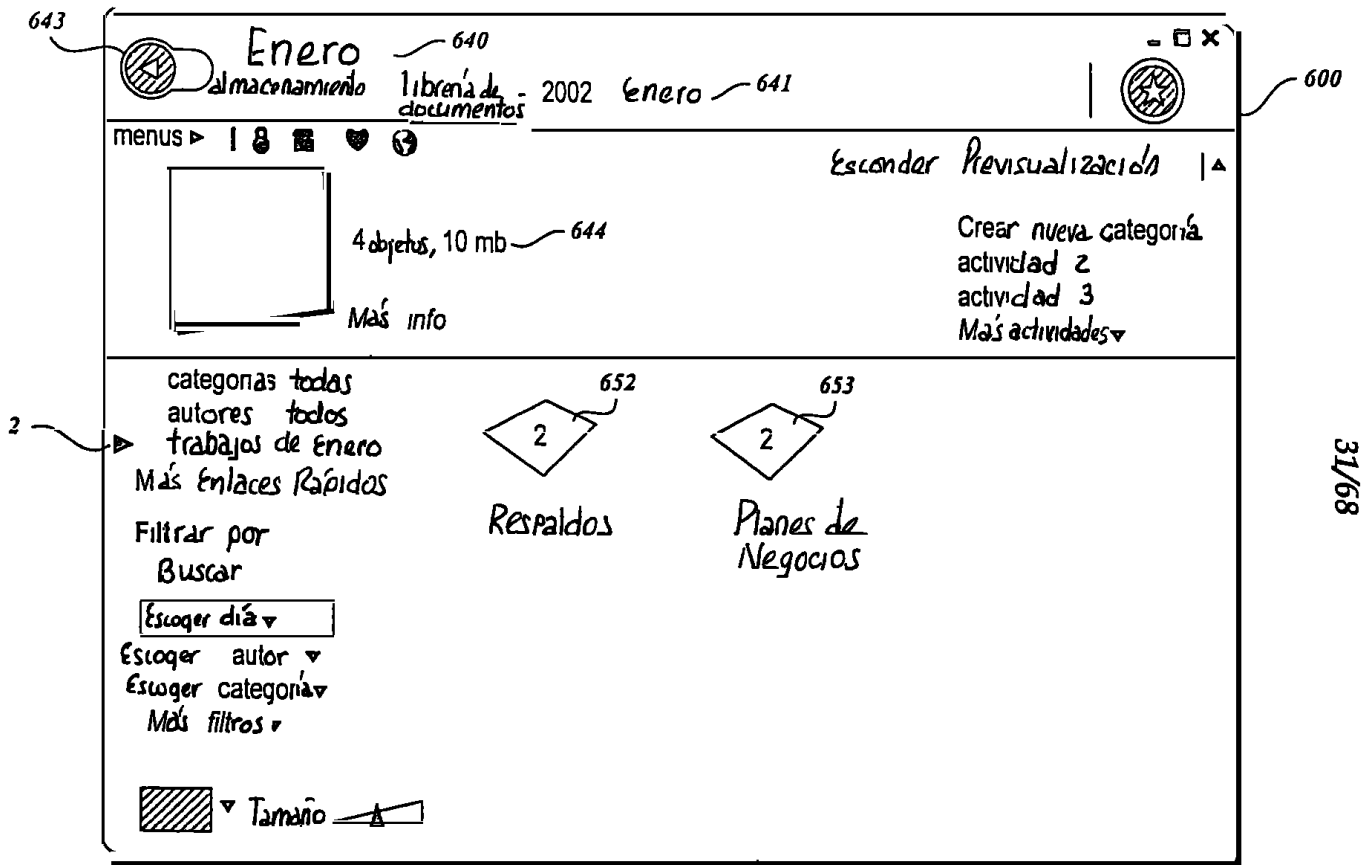


Fig.31.

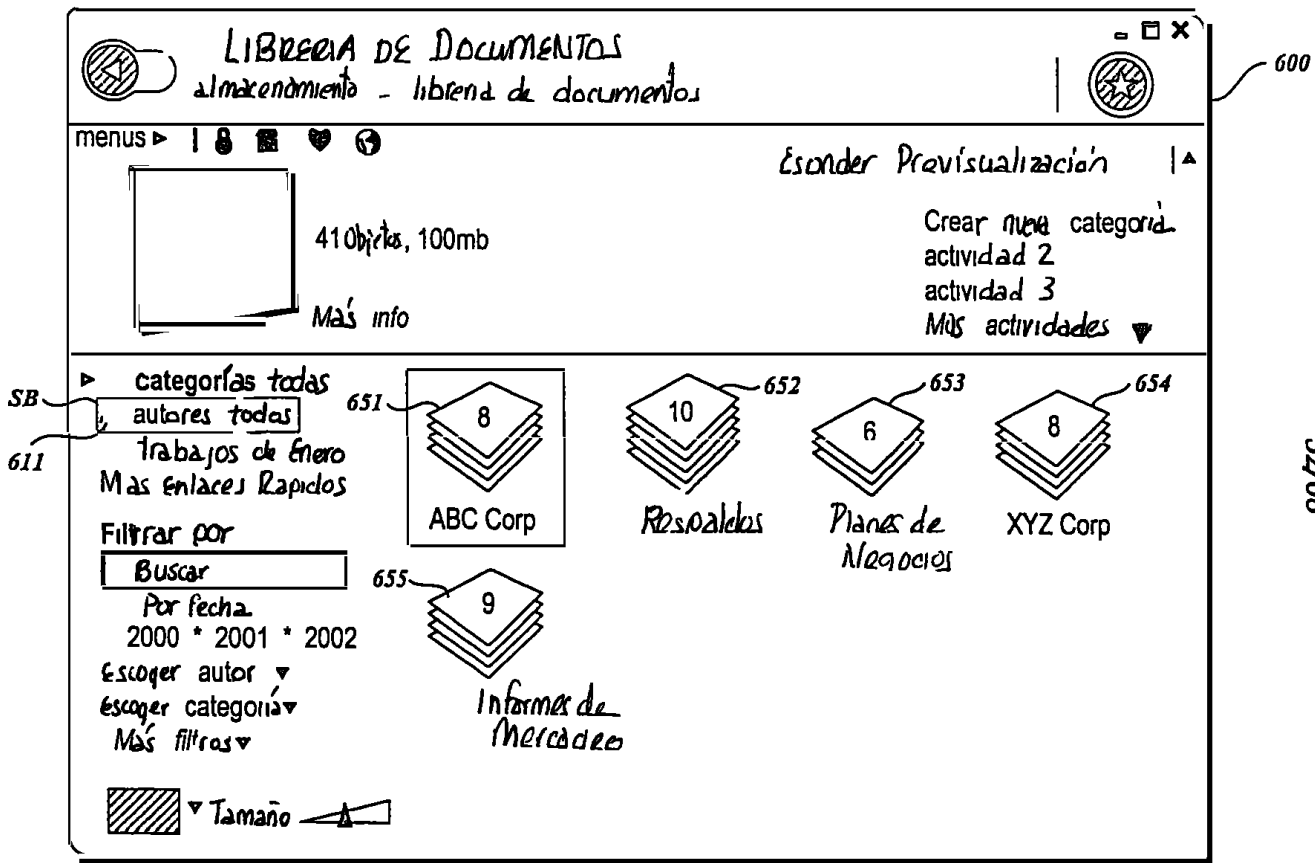


Fig.32.

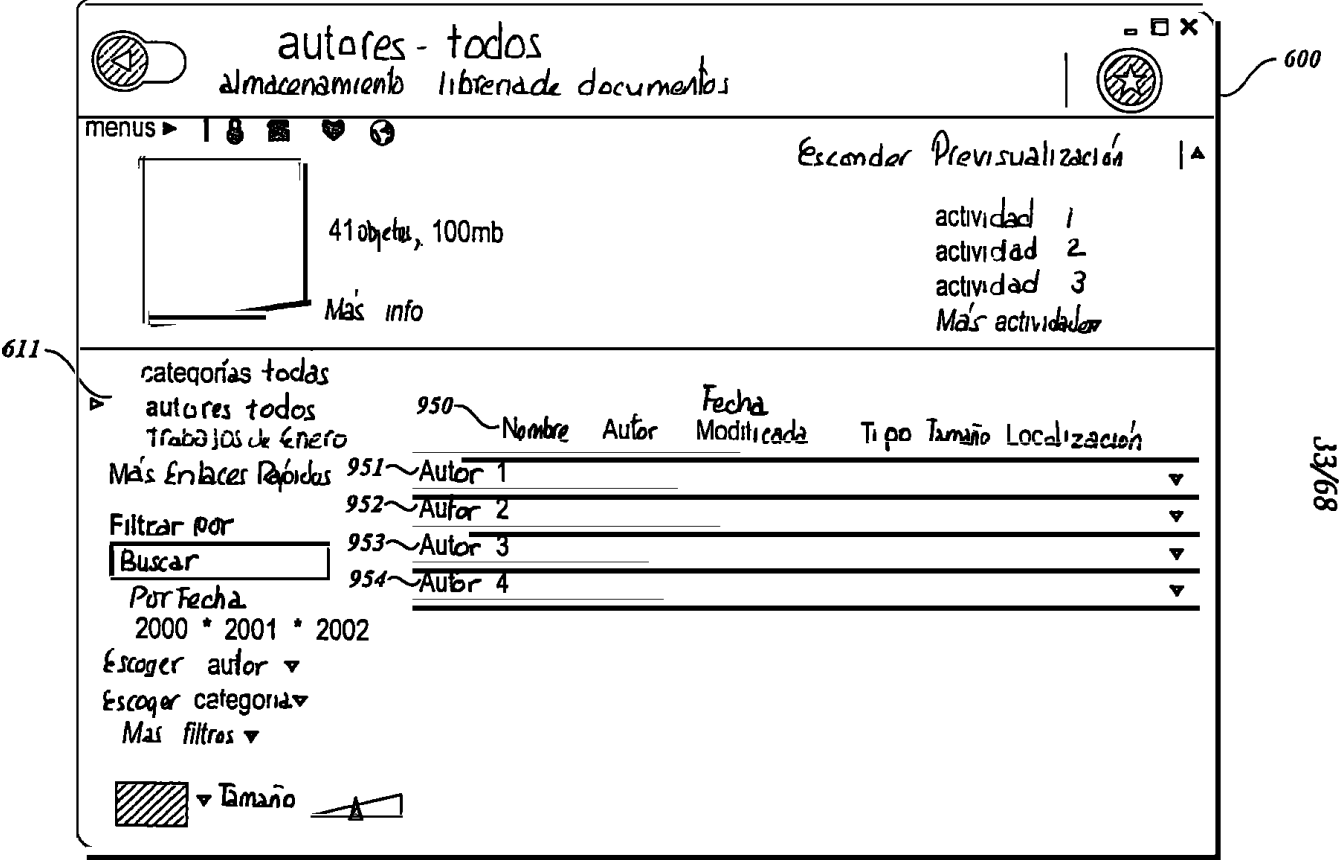


Fig.33.

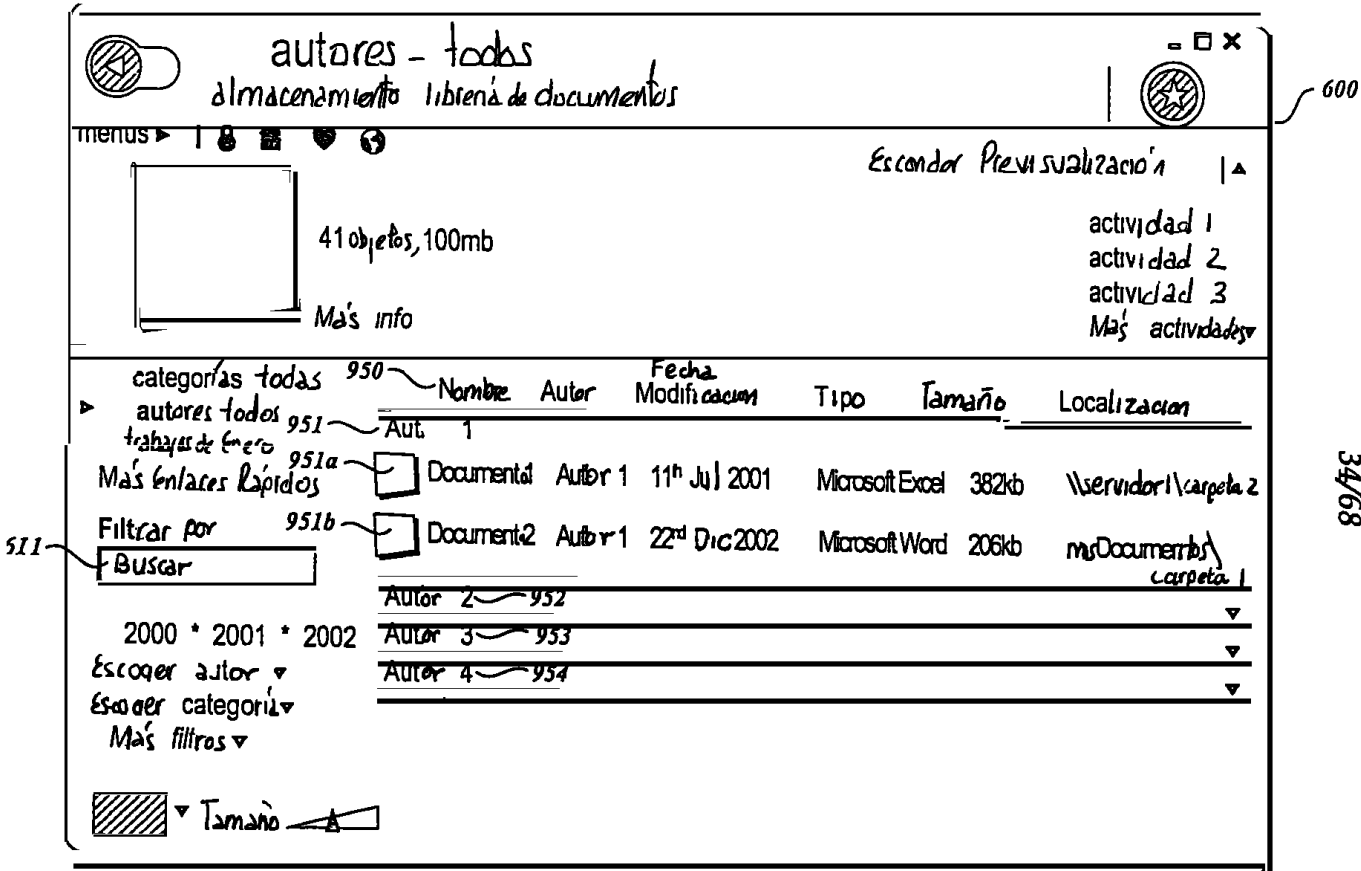


Fig.34.

210

35/68

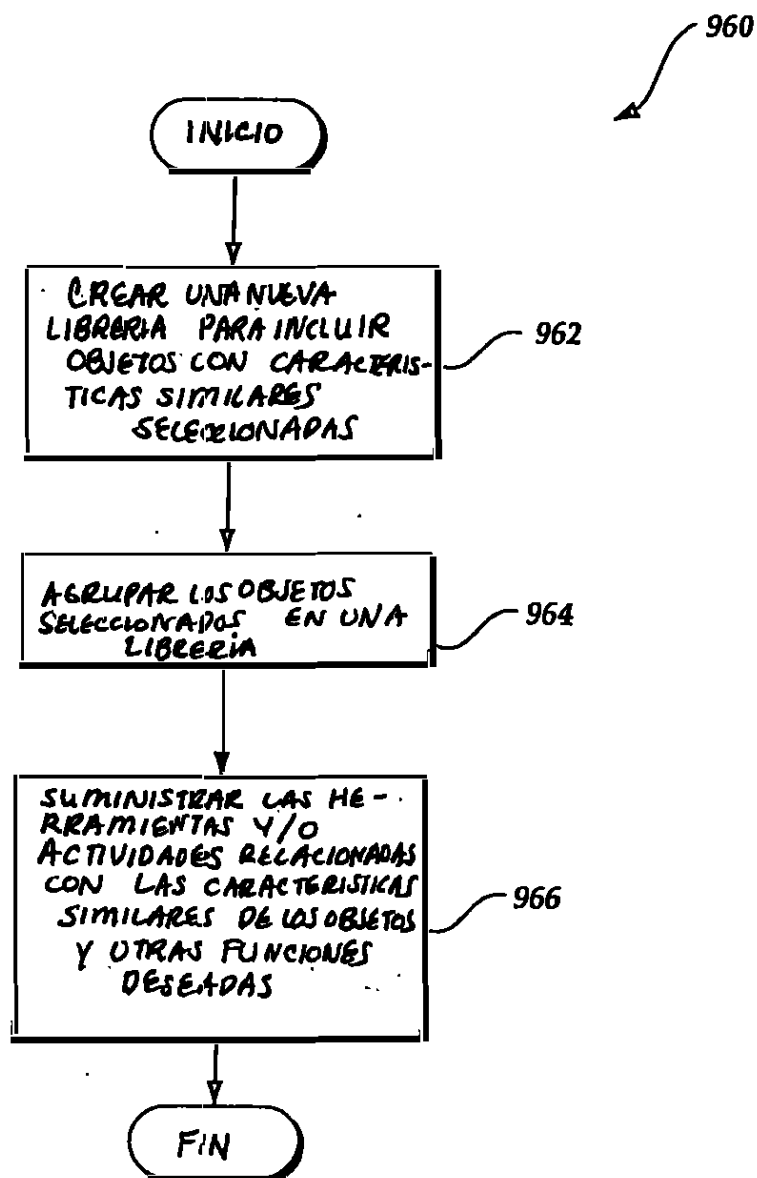


Fig.35.

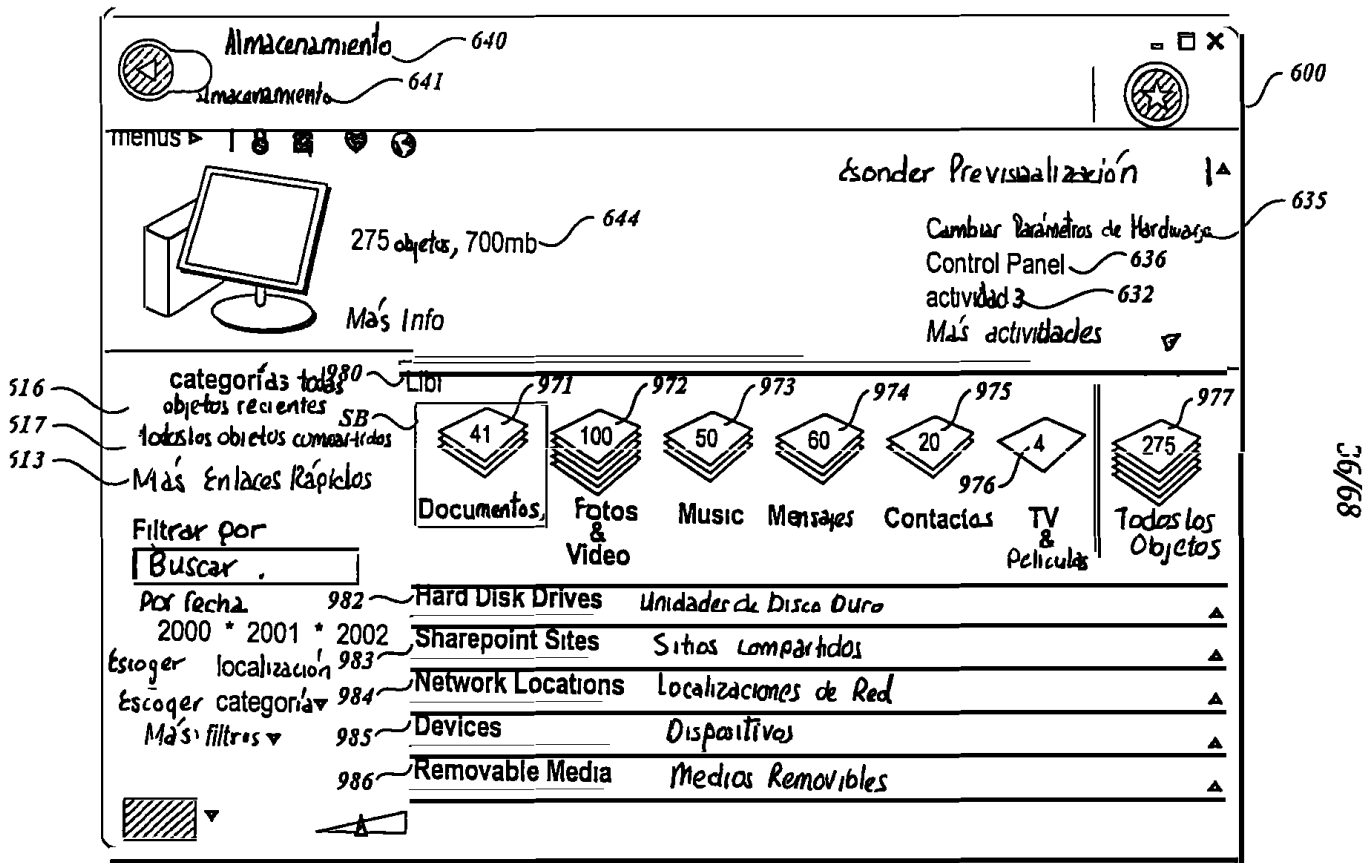


Fig.36.

37/68

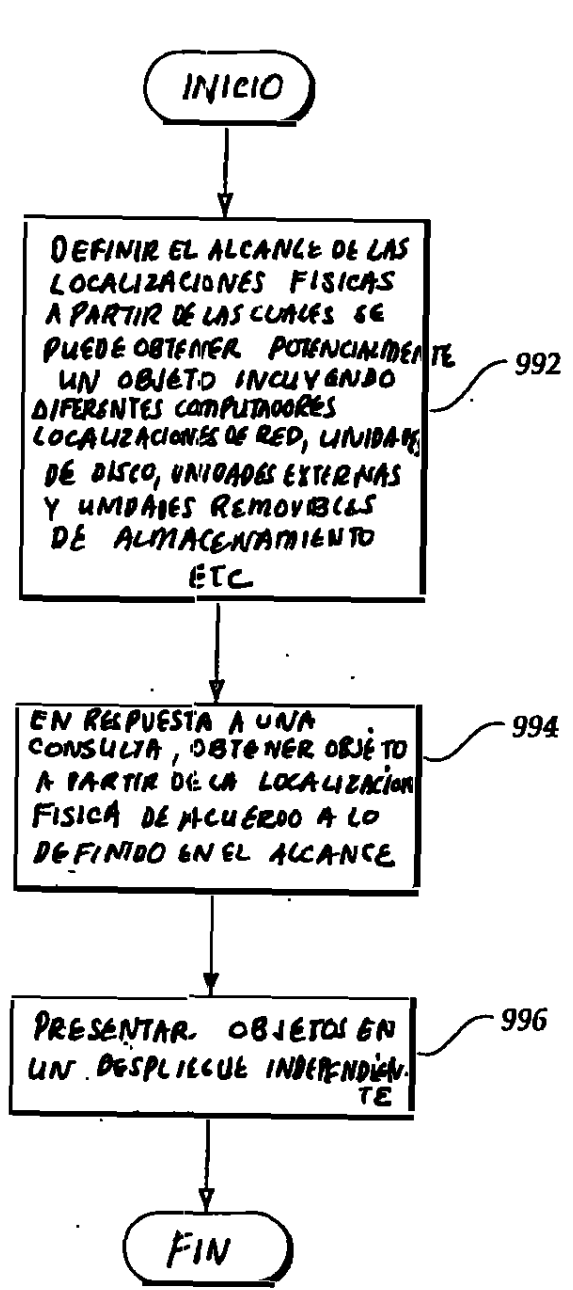


Fig.37.

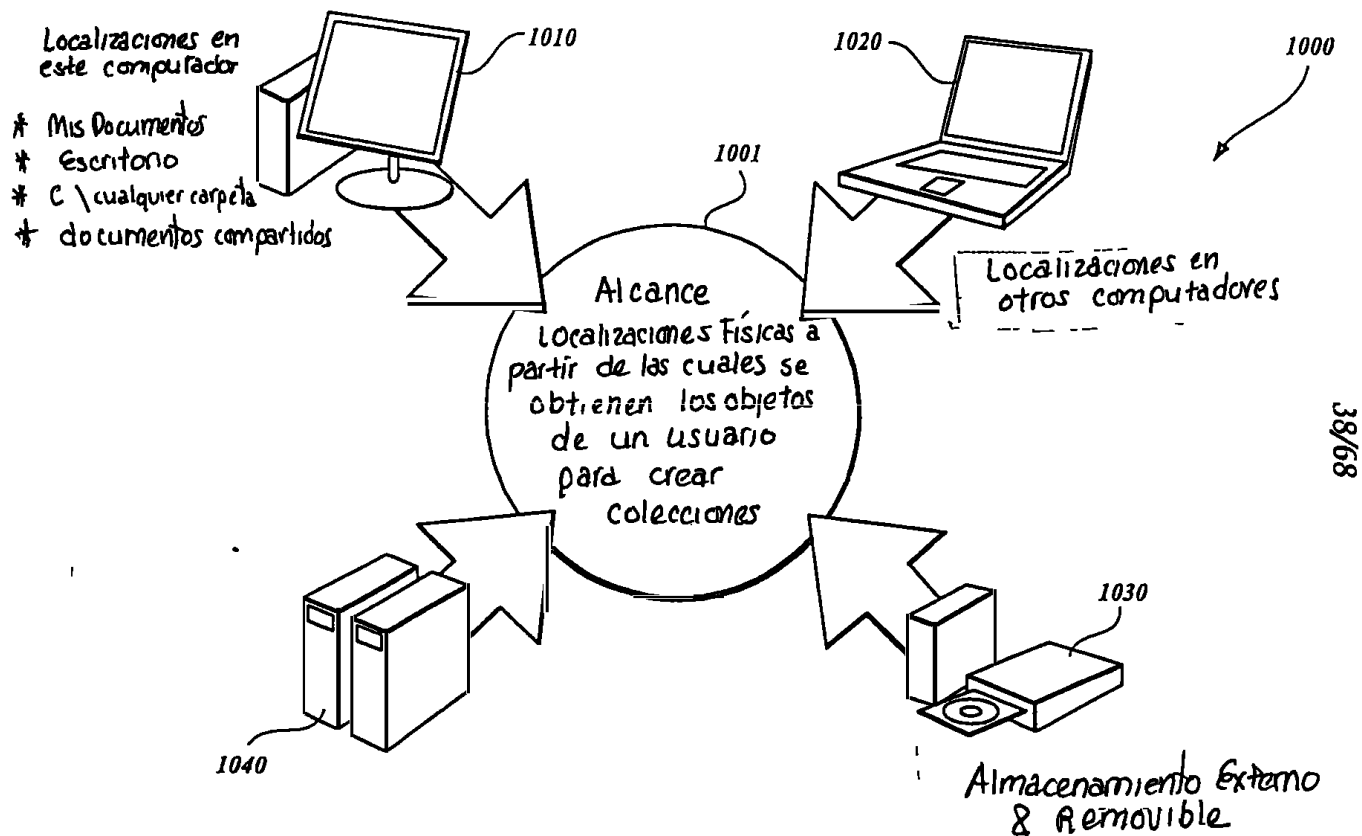


Fig.38.

39/68

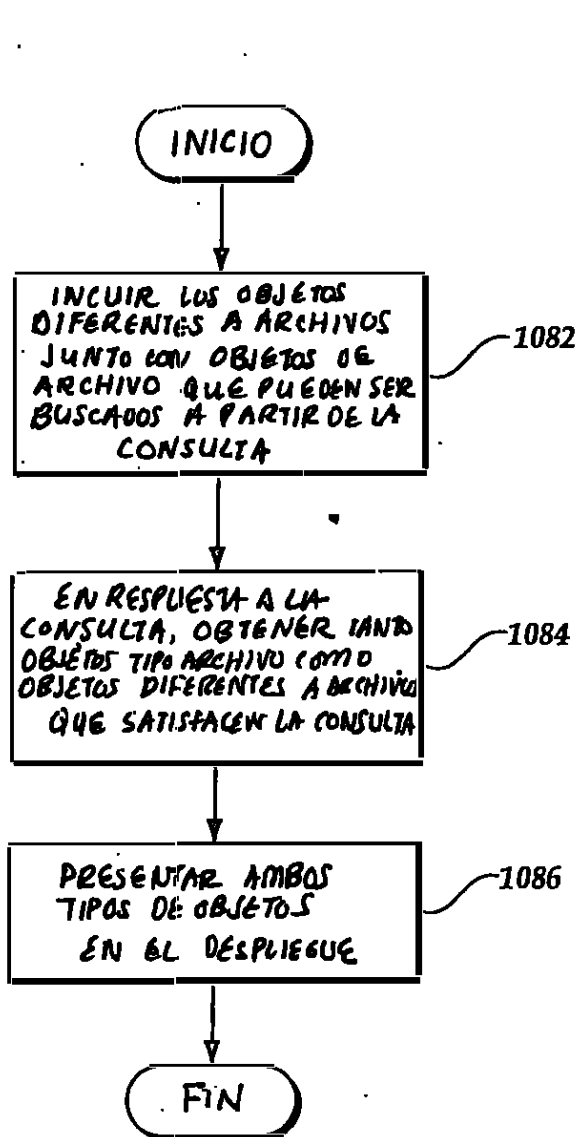


Fig.39.

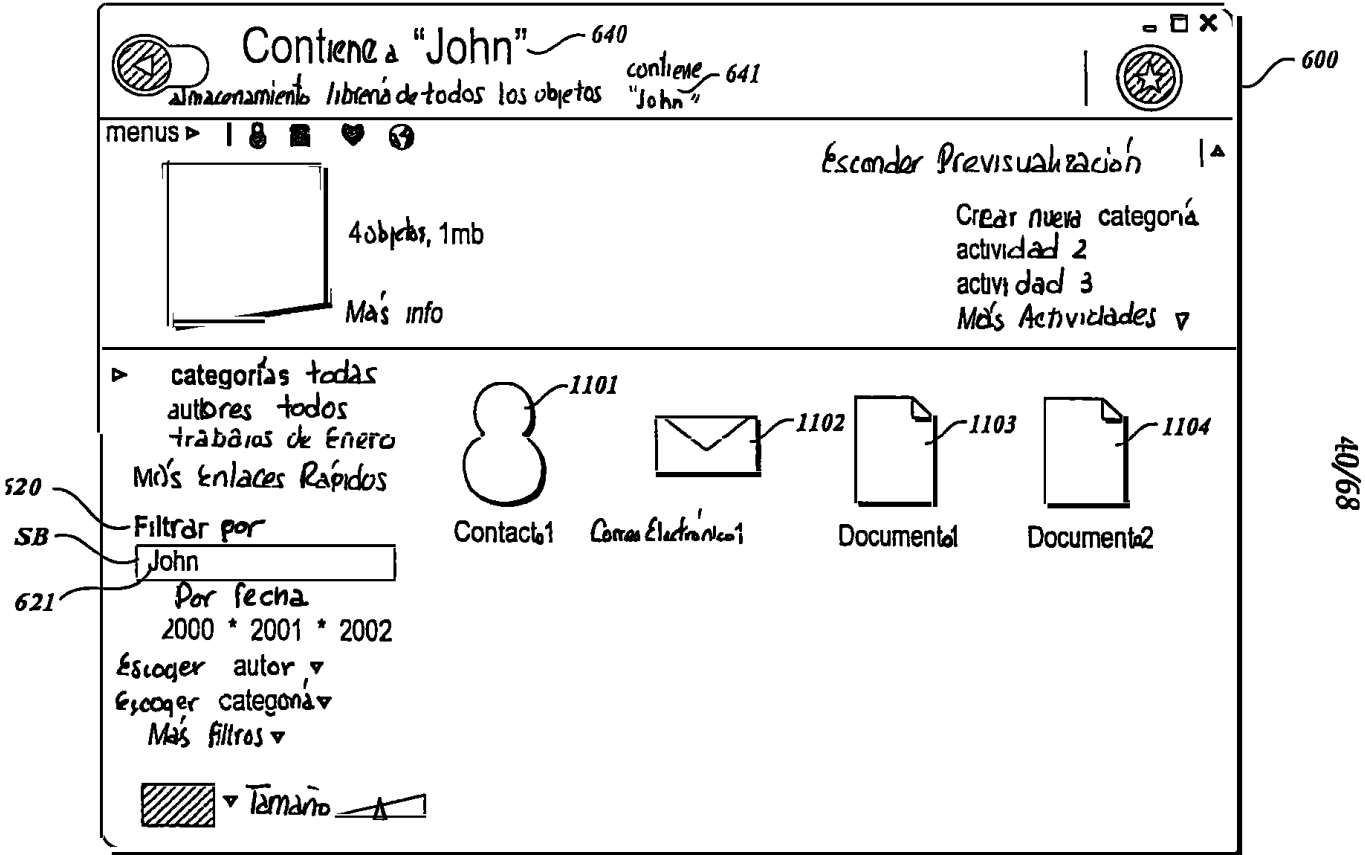


Fig.40.

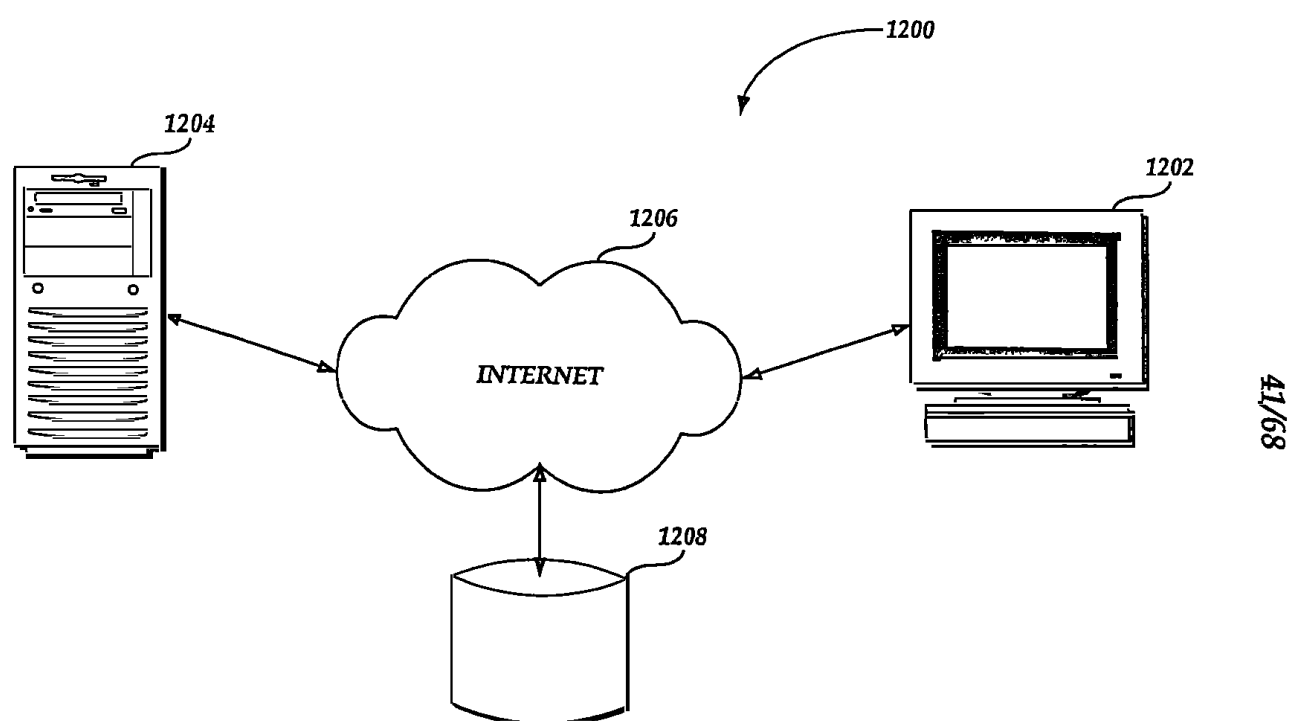
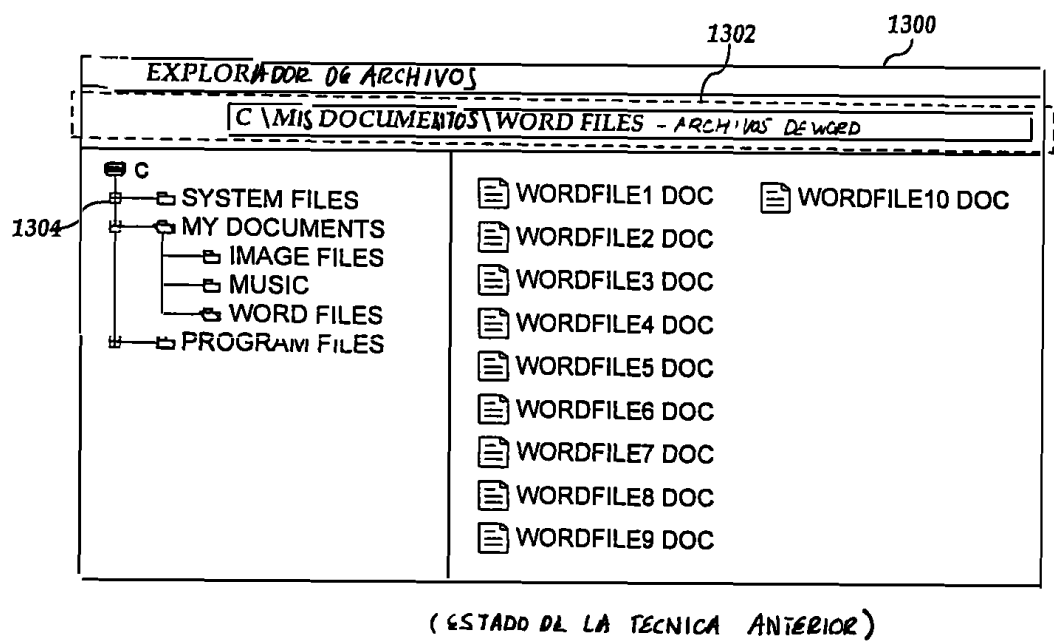


Fig.41.

41/68

217

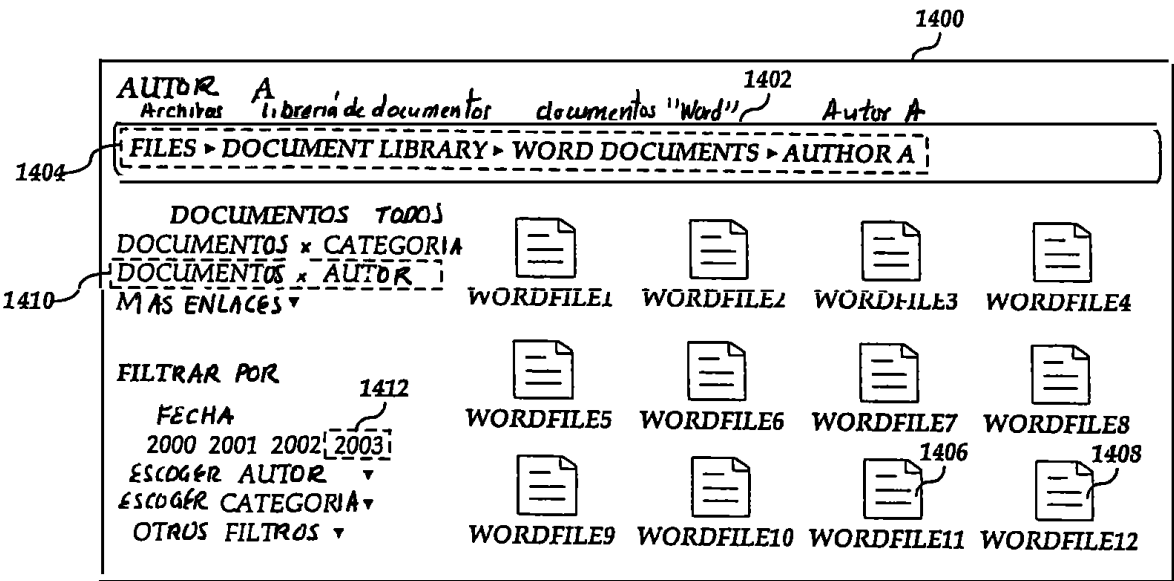


42/68

Fig.42.

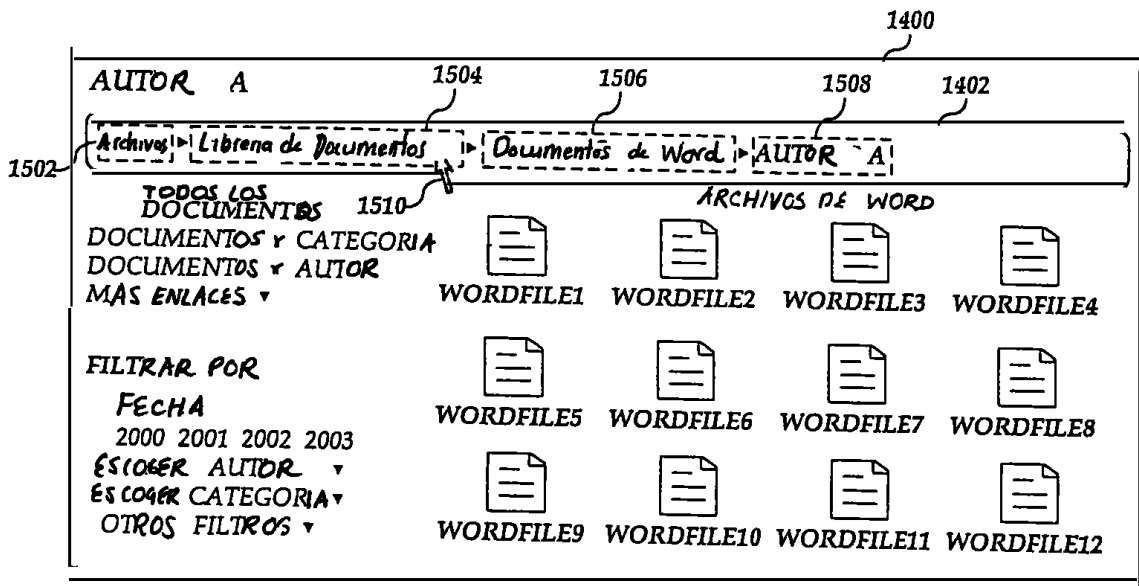
225

218



43/68

Fig.43.



44/68

Fig.44A.

220

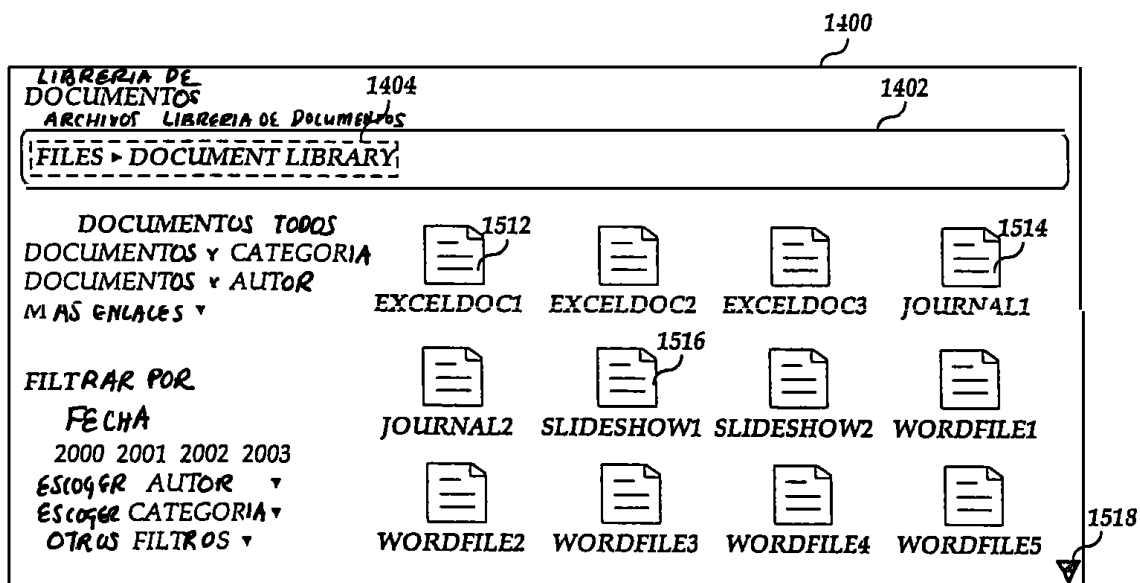


Fig.44B.

46/68

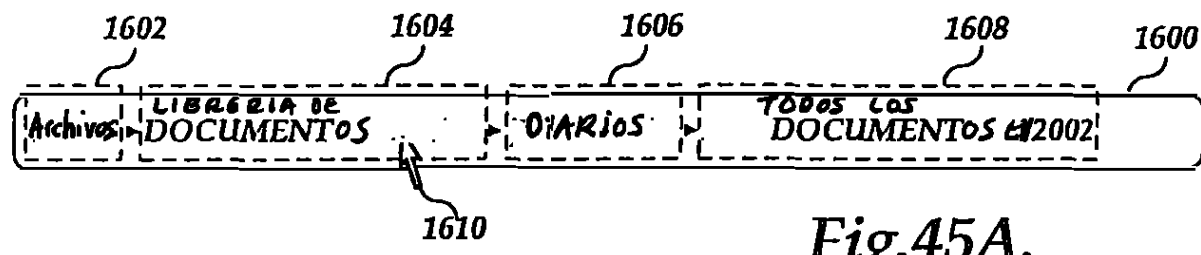


Fig.45A.

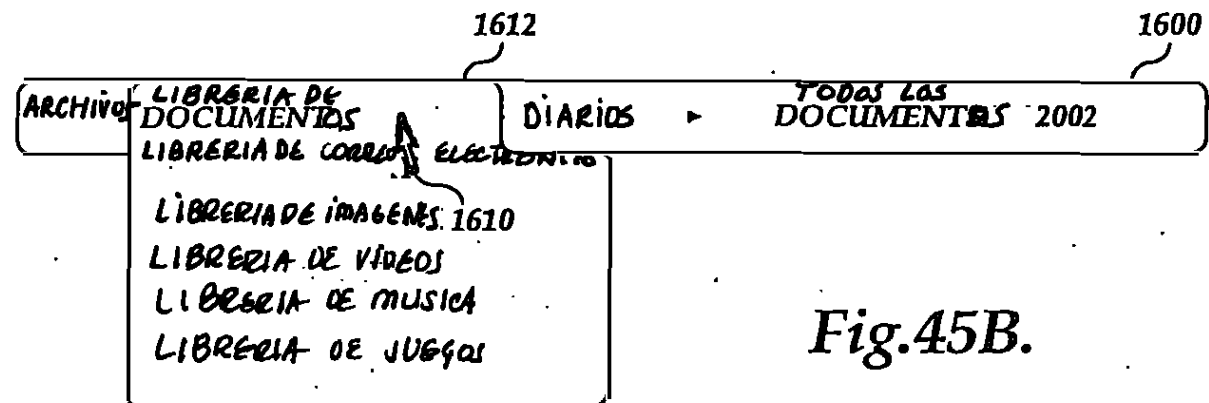


Fig.45B.

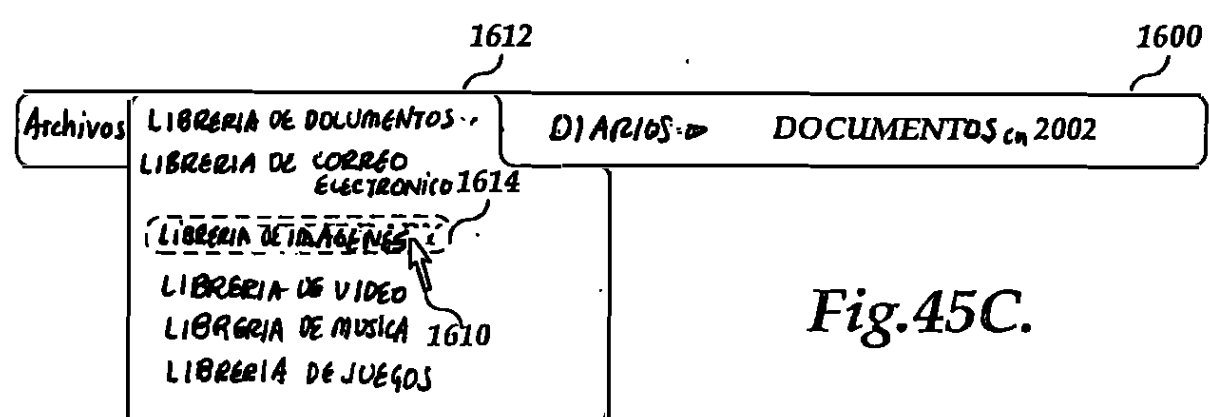


Fig.45C.

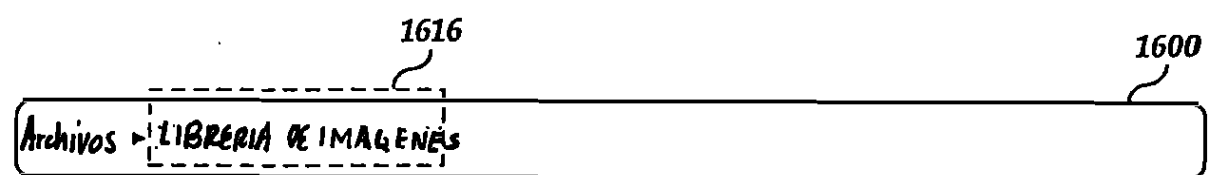


Fig.45D.

47/68

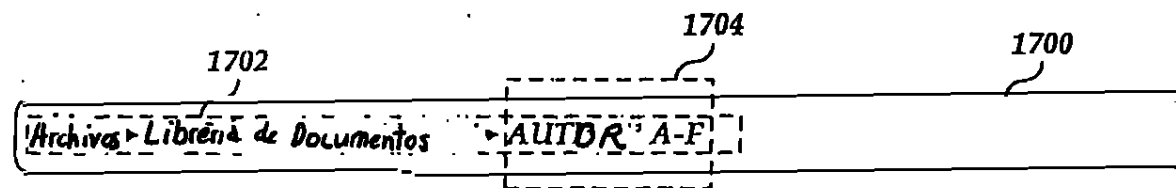


Fig.46A.

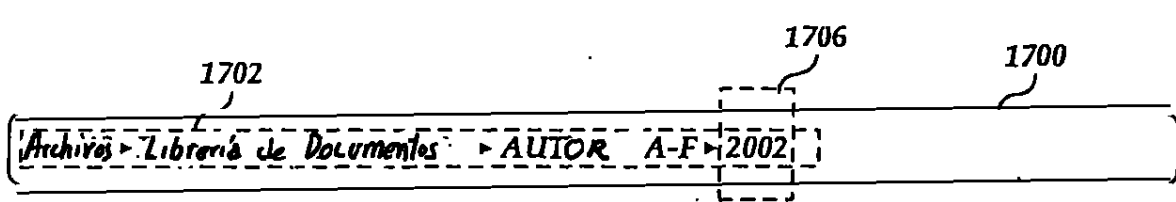


Fig.46B.

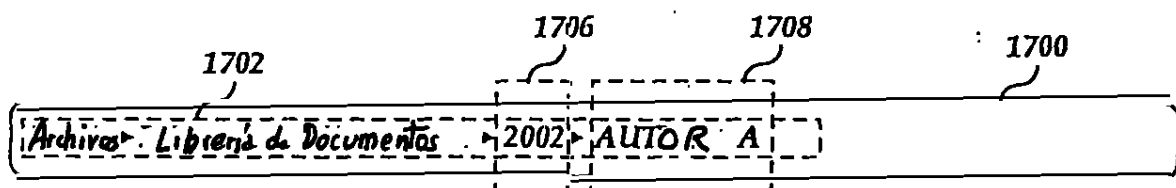


Fig.46C.

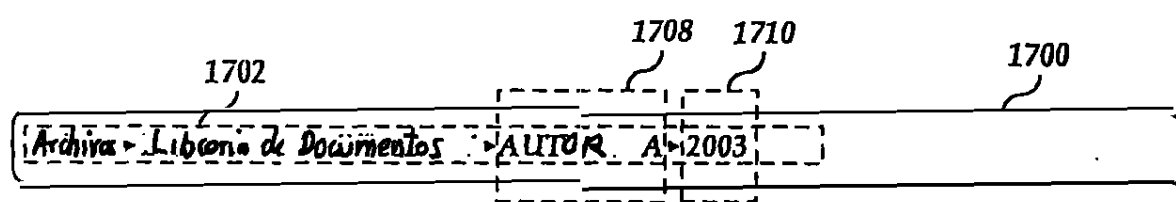


Fig.46D.

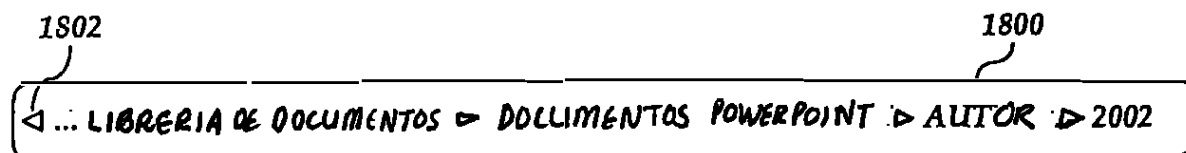


Fig.47A.

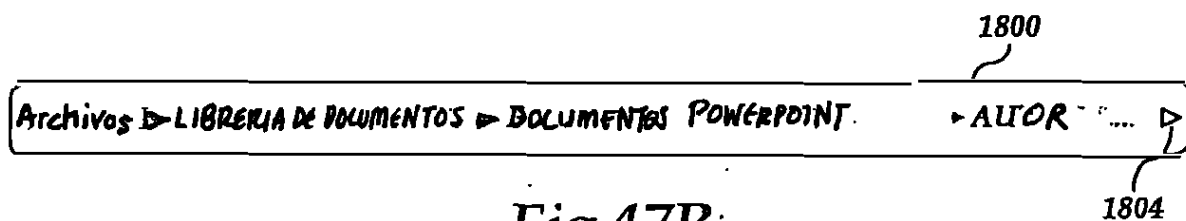


Fig.47B.

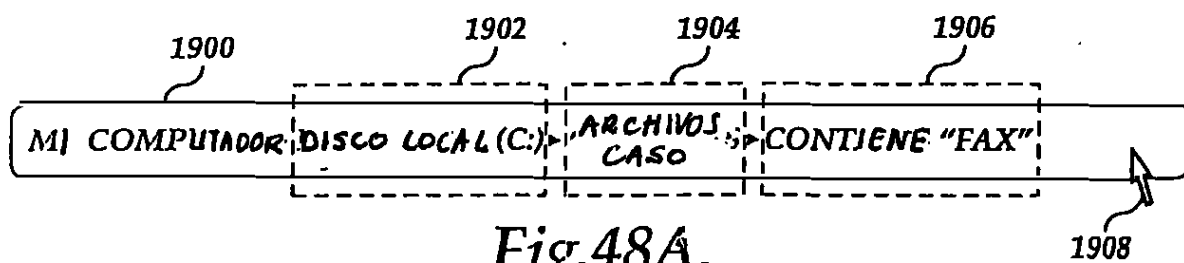


Fig.48A.

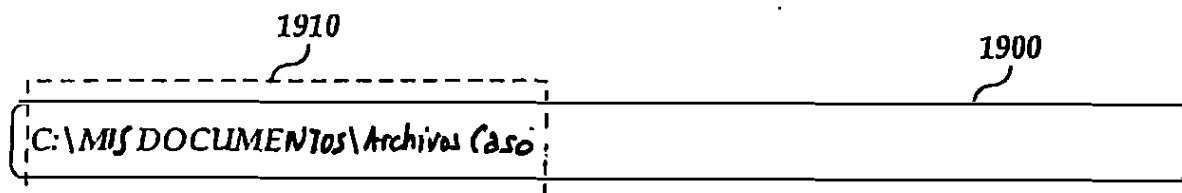
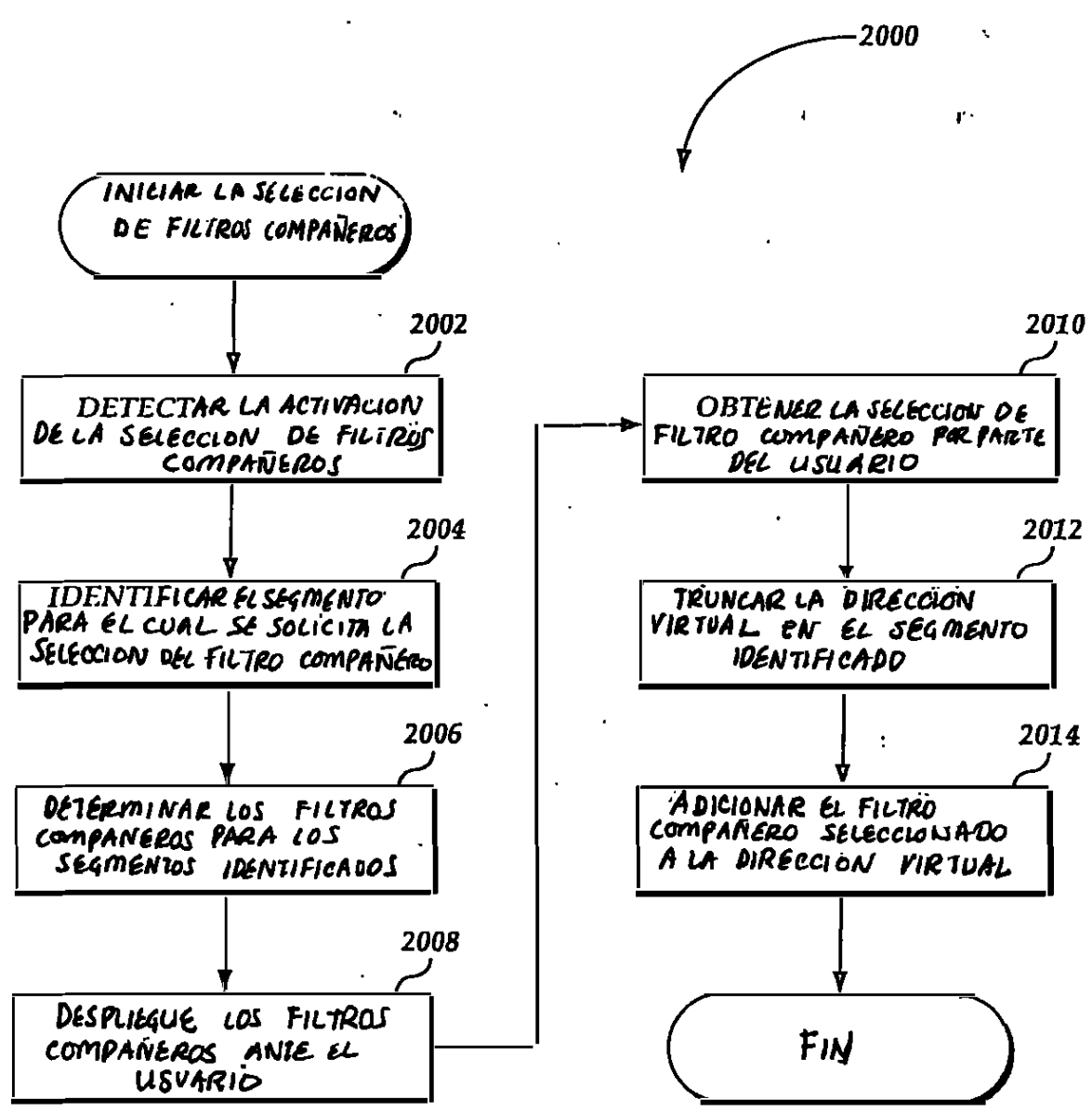
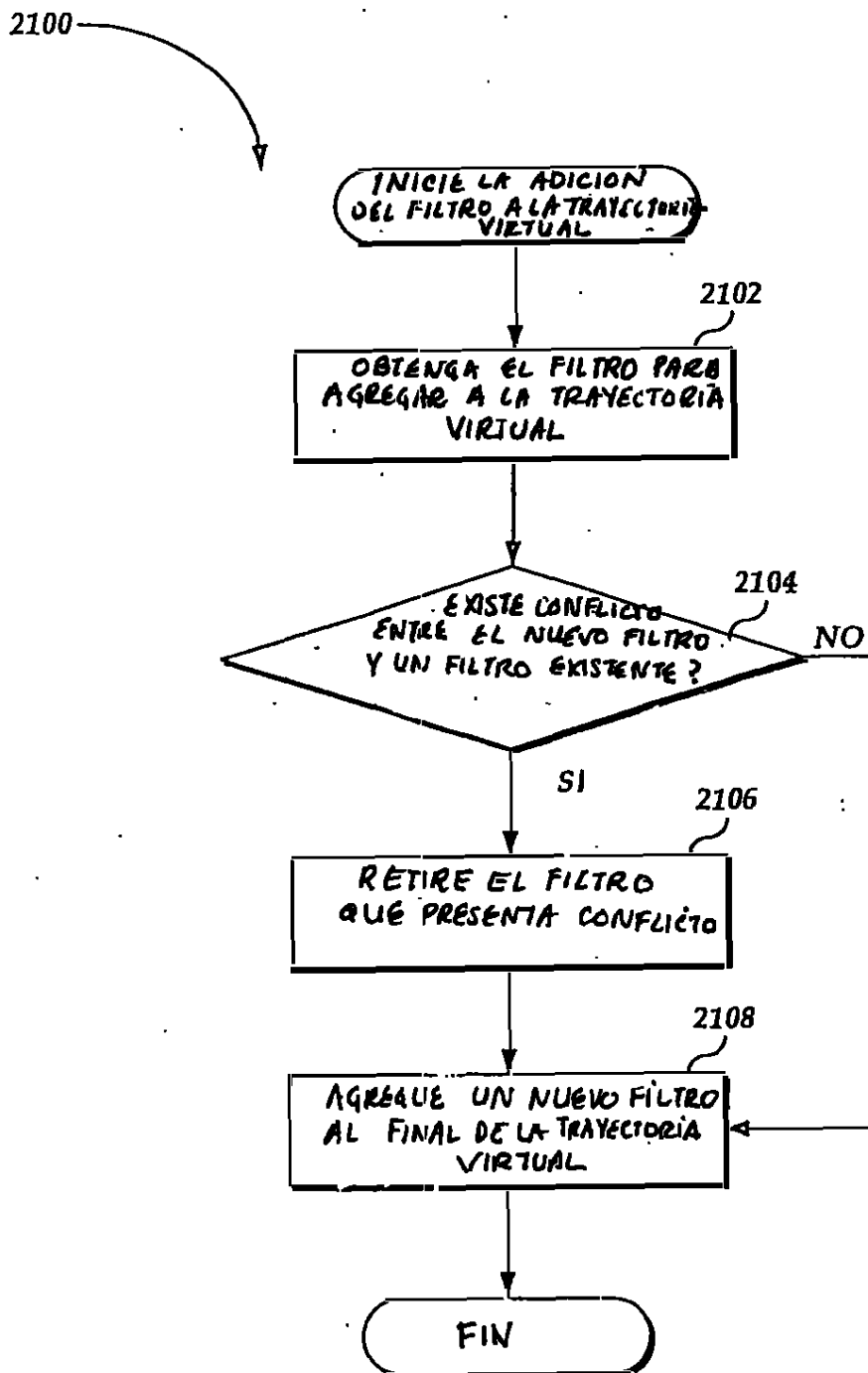
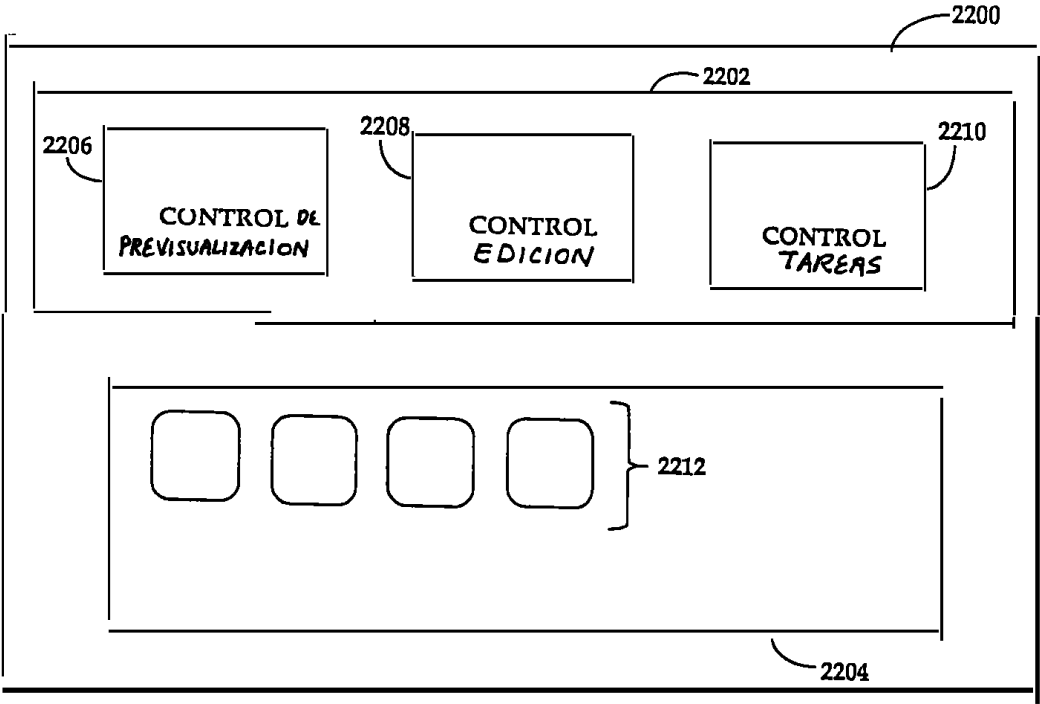


Fig.48B.



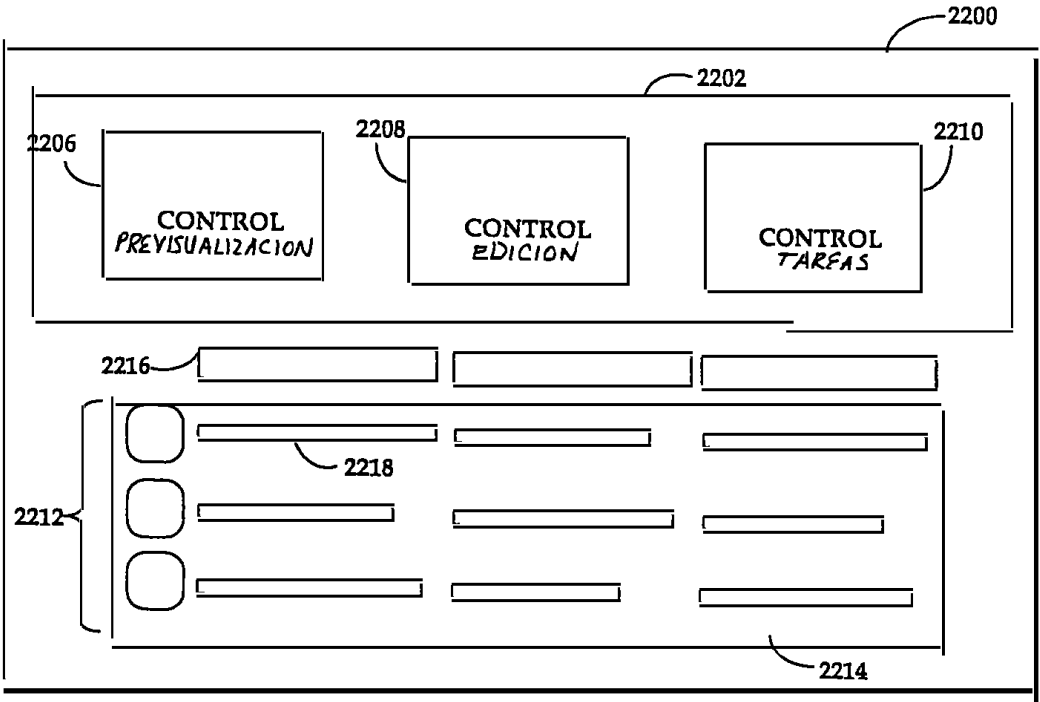
50/68





51/68

Fig.51A.



52/68

Fig.51B.

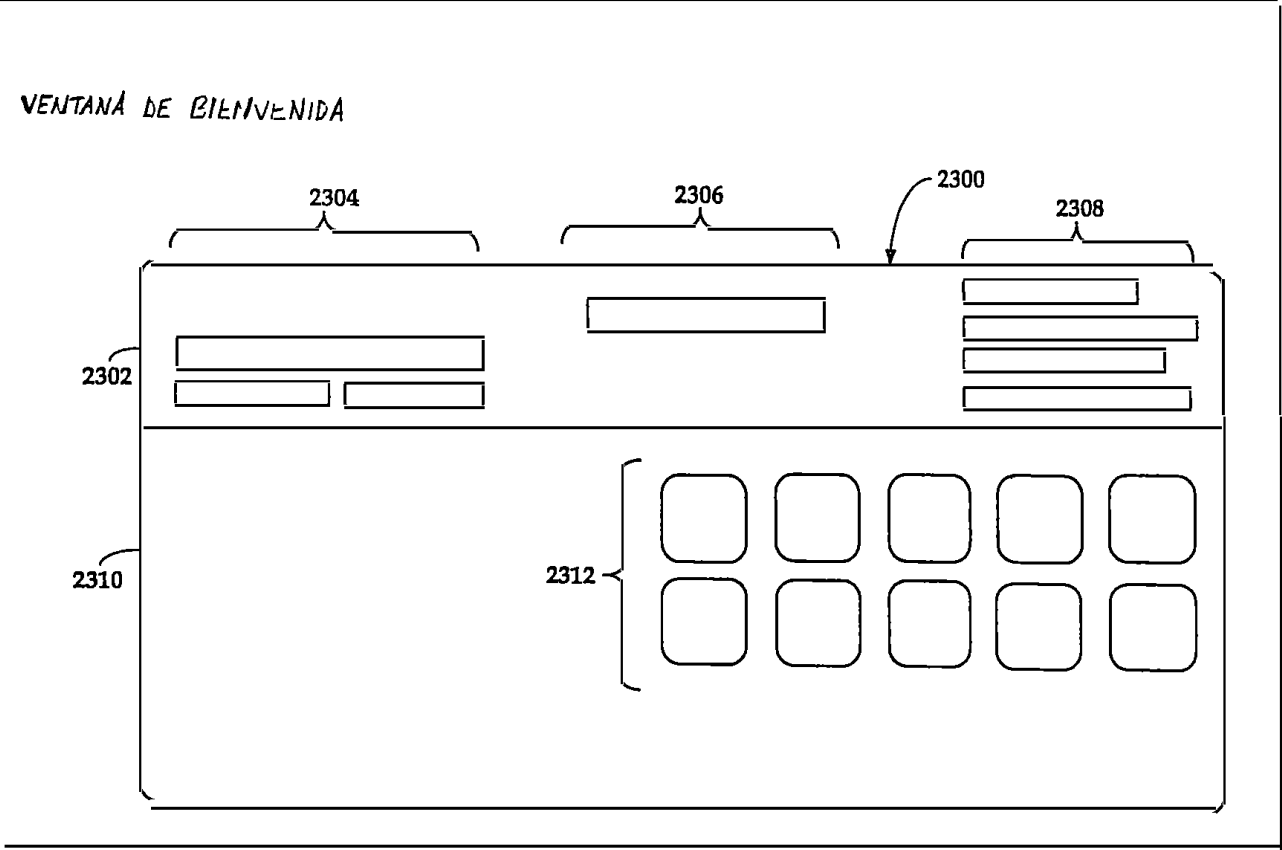


Fig.52.

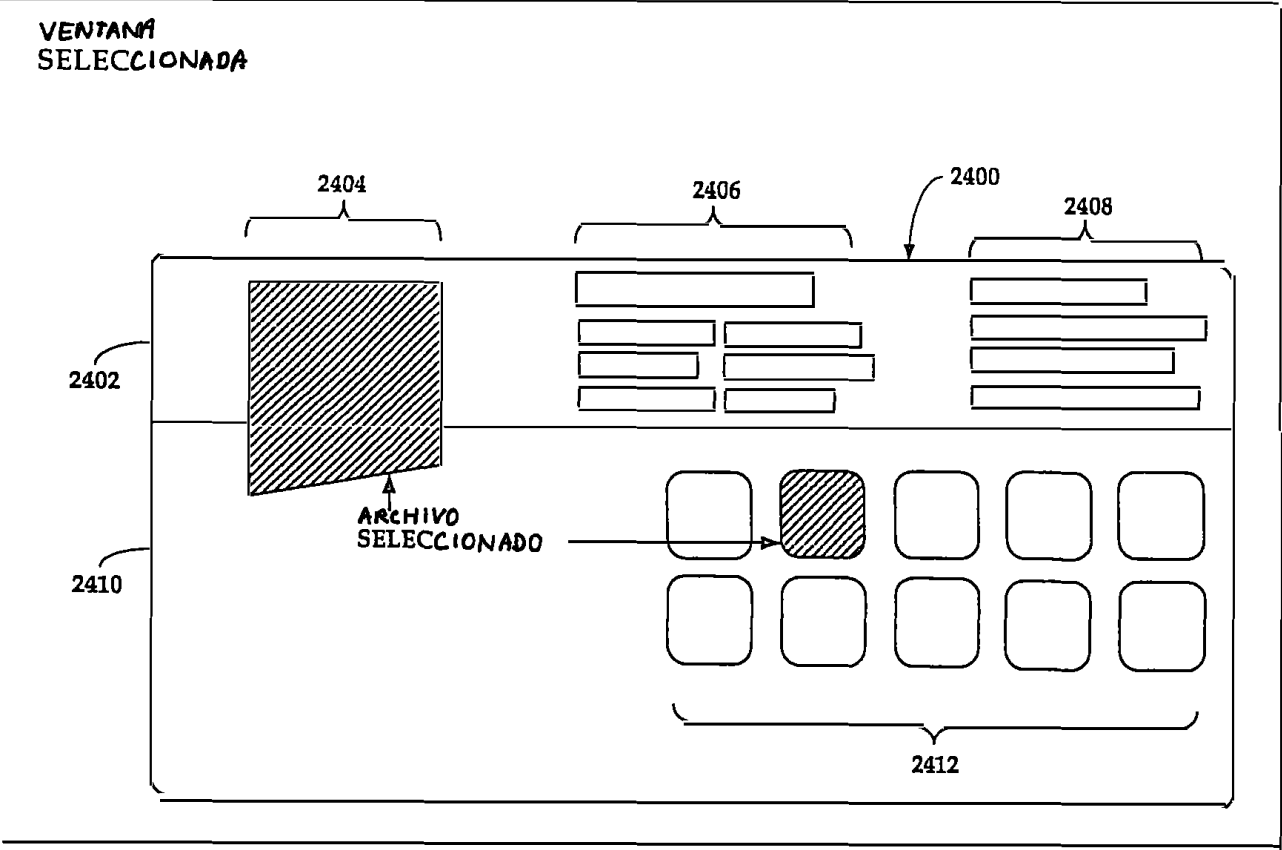
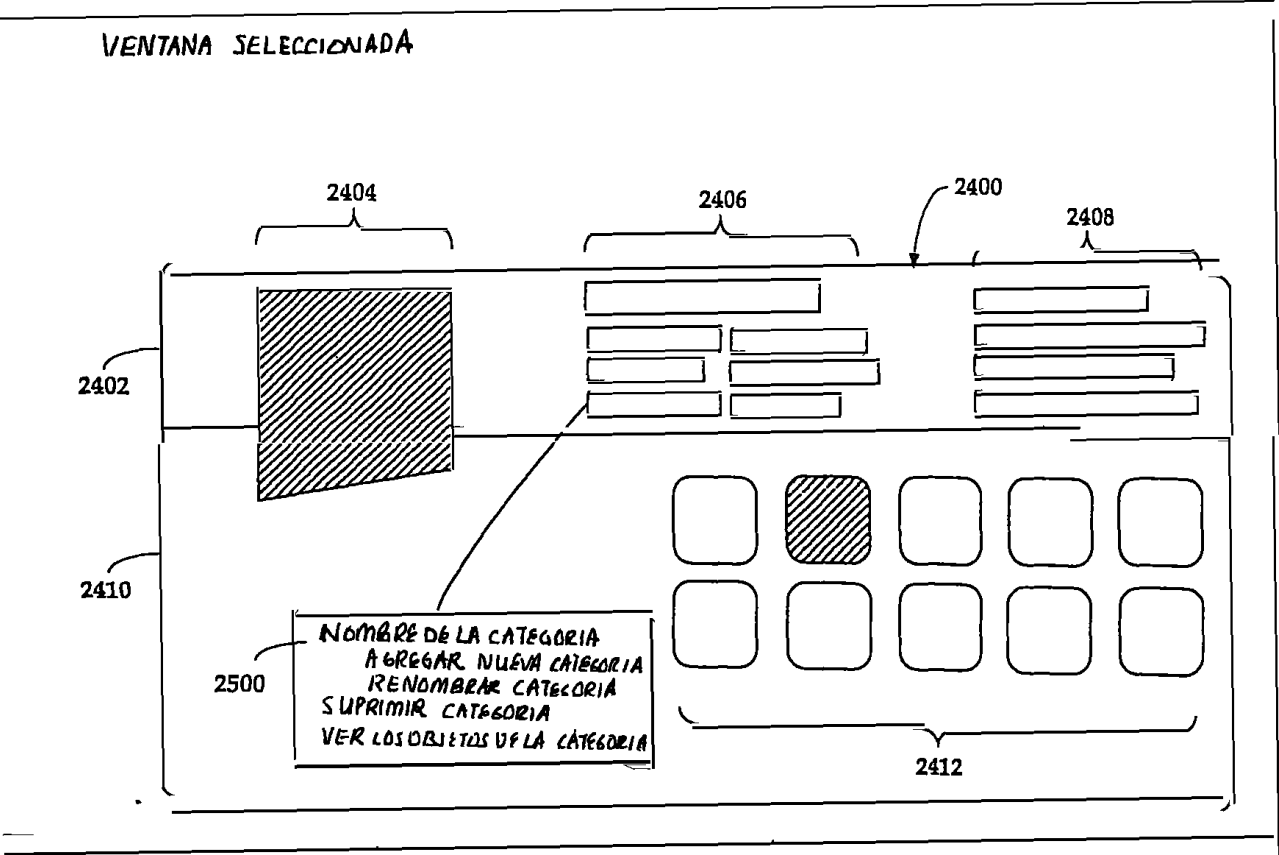


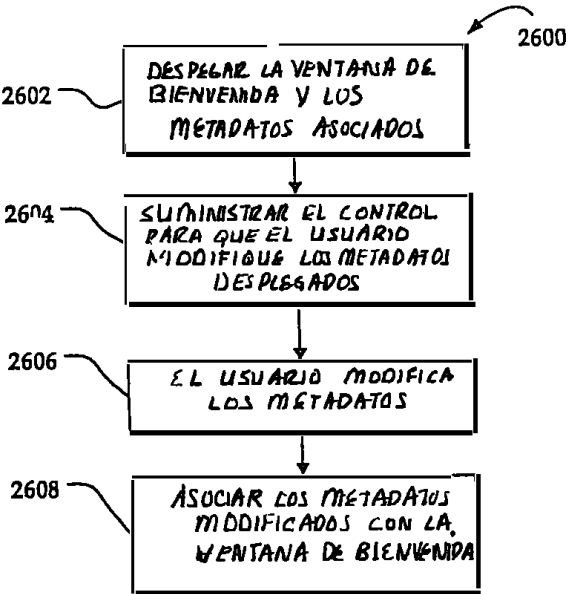
Fig.53.

230



55/68

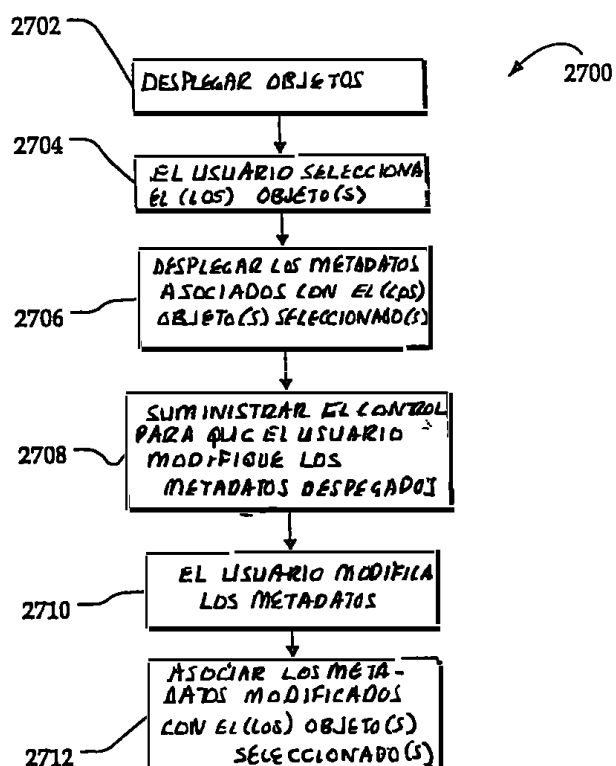
Fig.54.



56/68

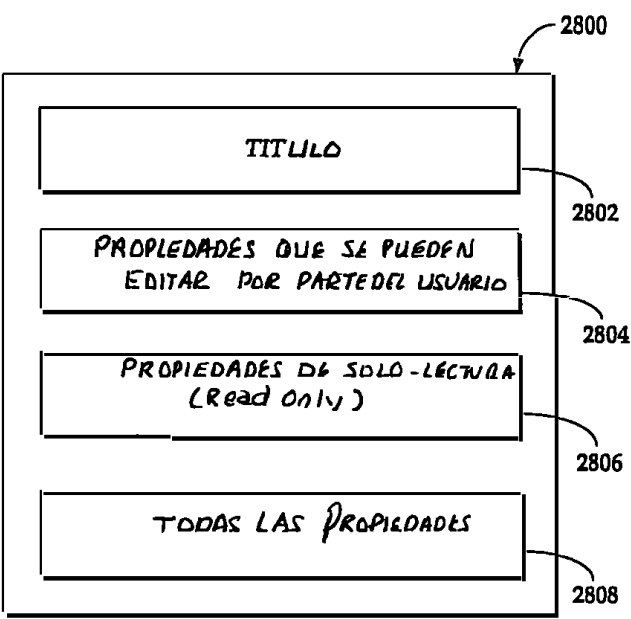
Fig.55.

288



57/68

Fig.56.



58/68

FIG.57.

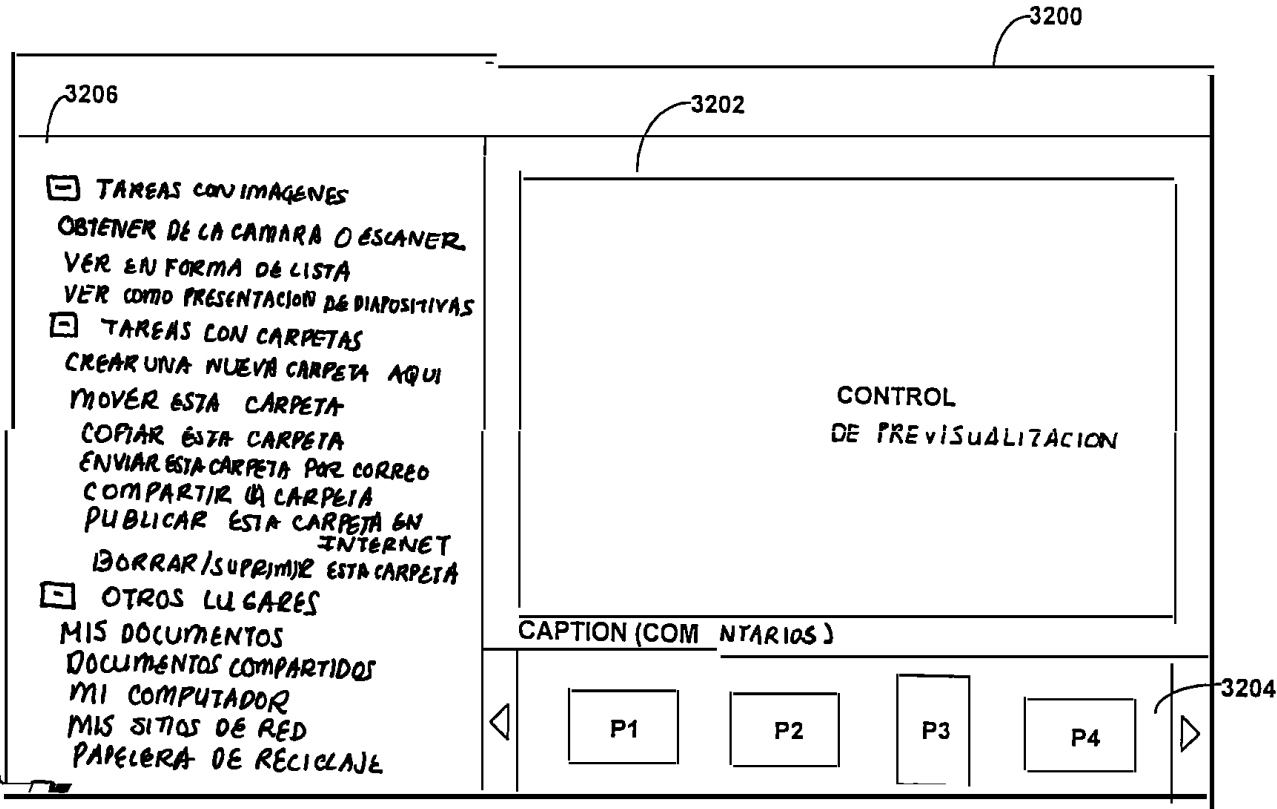
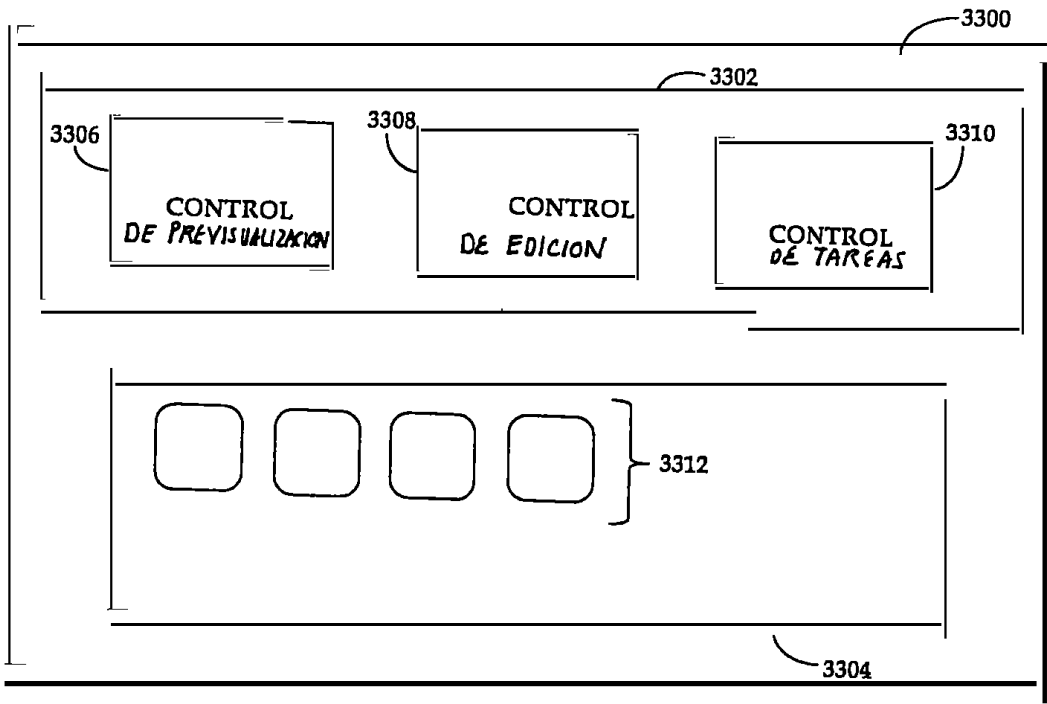


Fig.58.

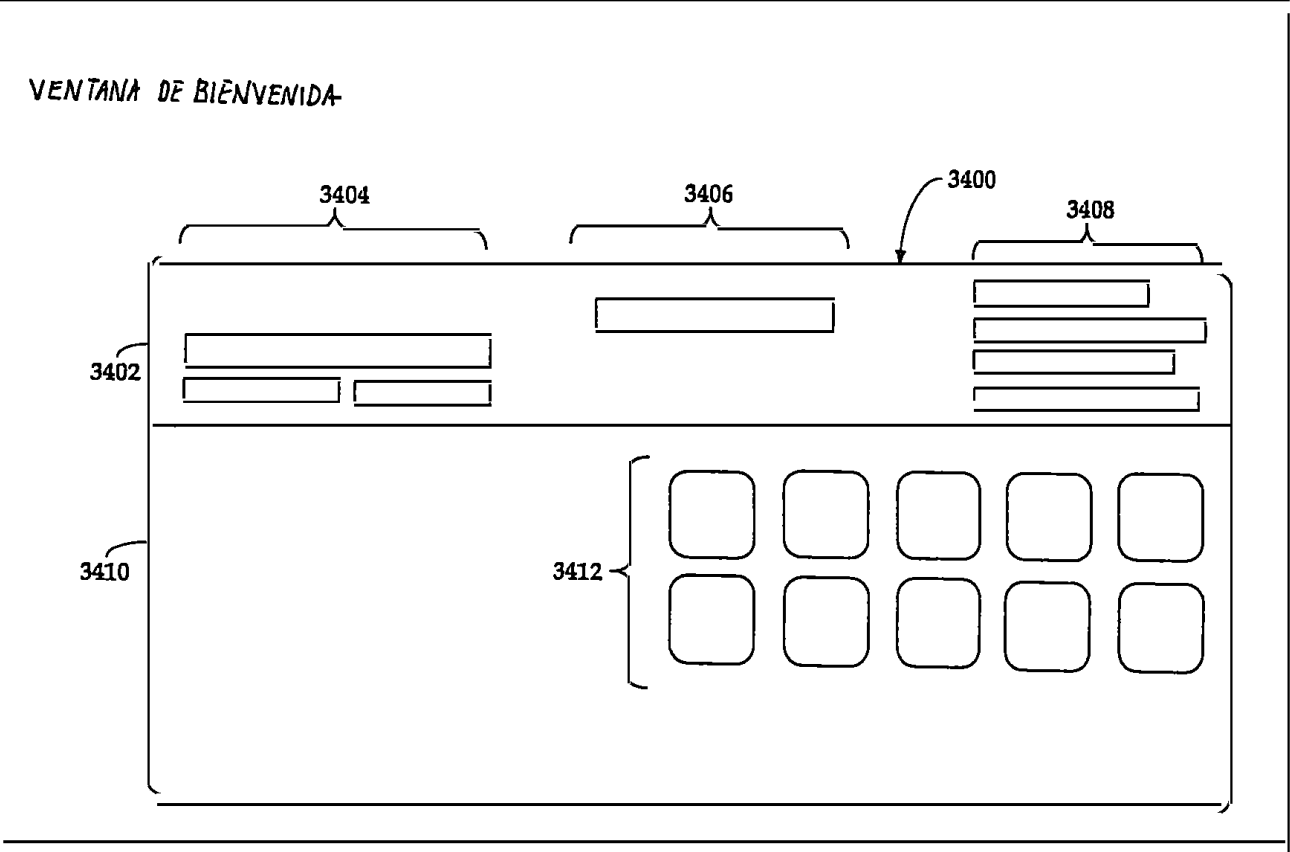
235



60/68

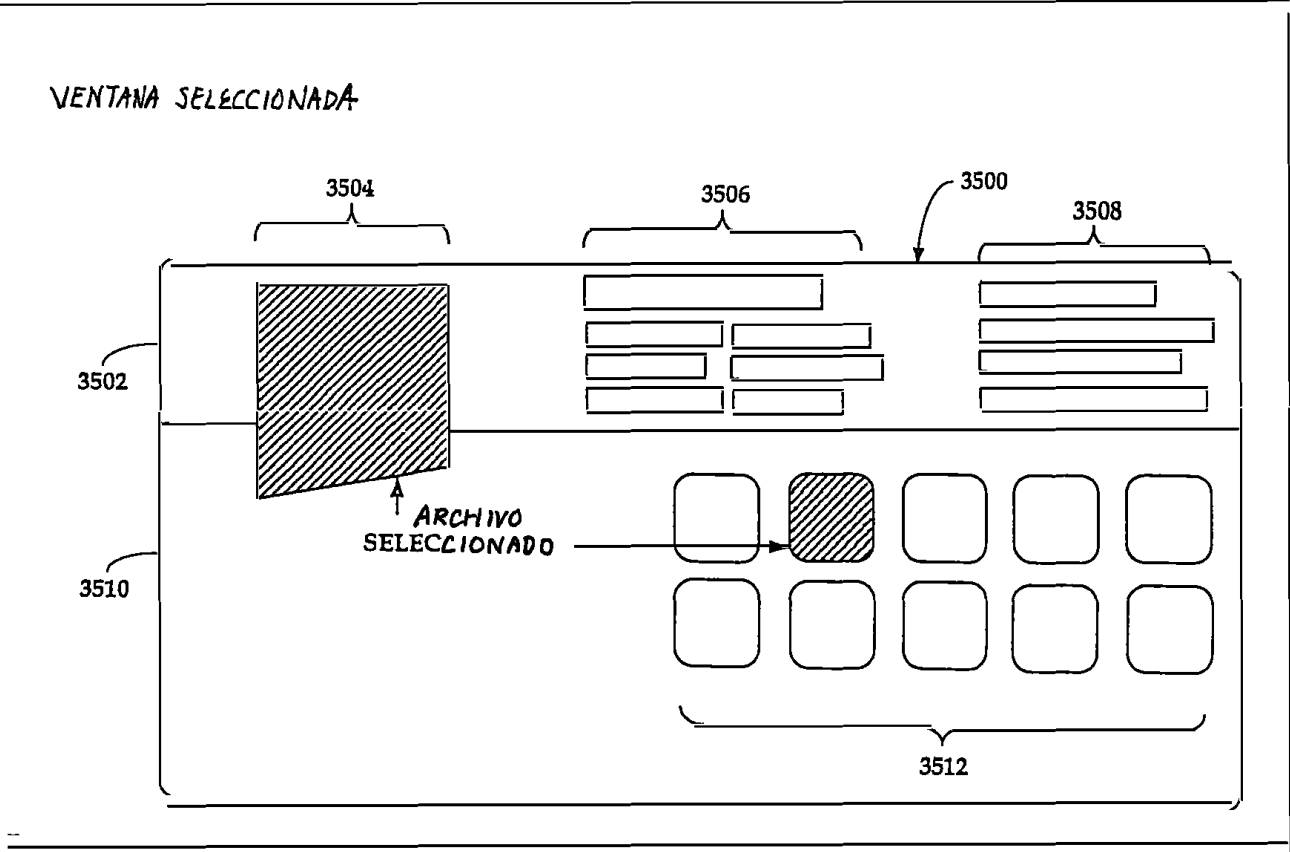
Fig.59.

286



61/68

Fig.60.

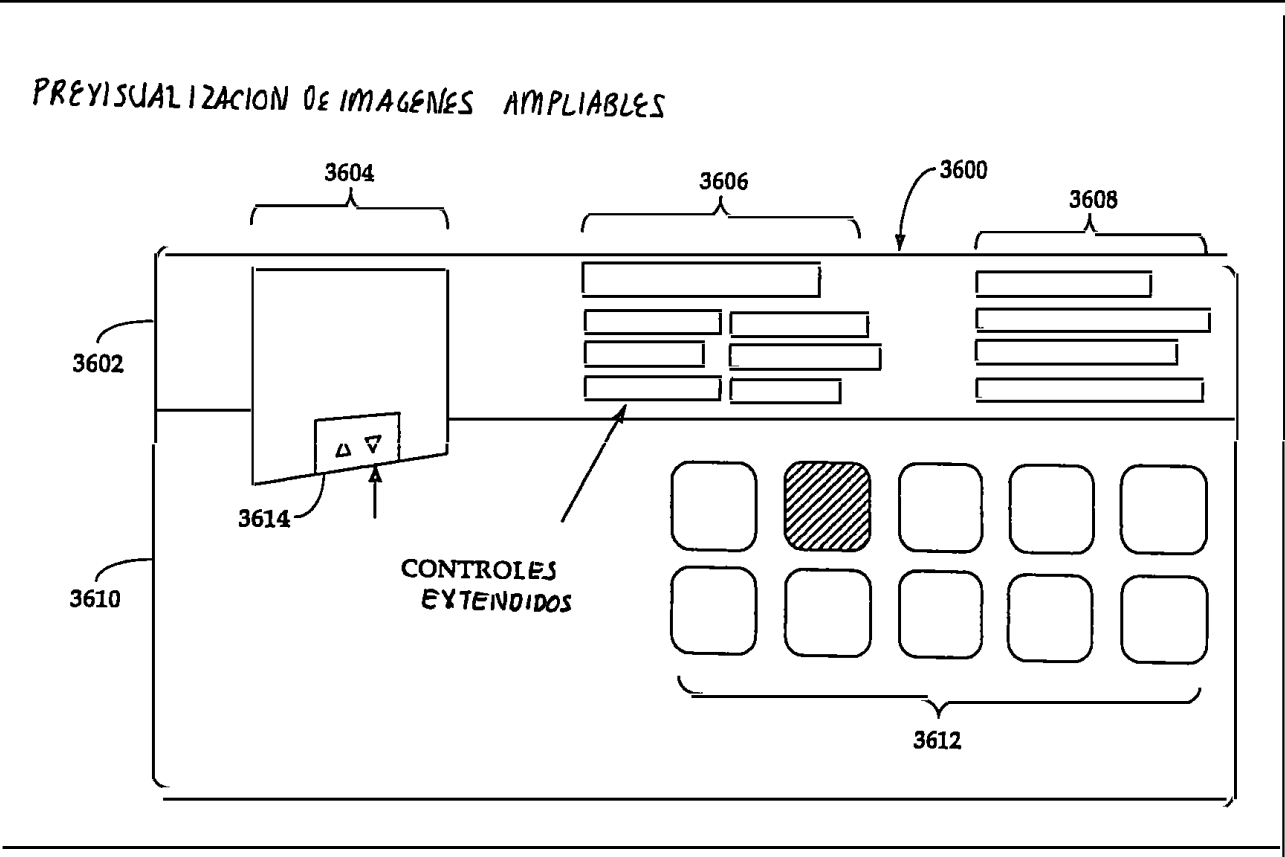


62/68

Fig.61.

240

238



63/68

Fig.62.

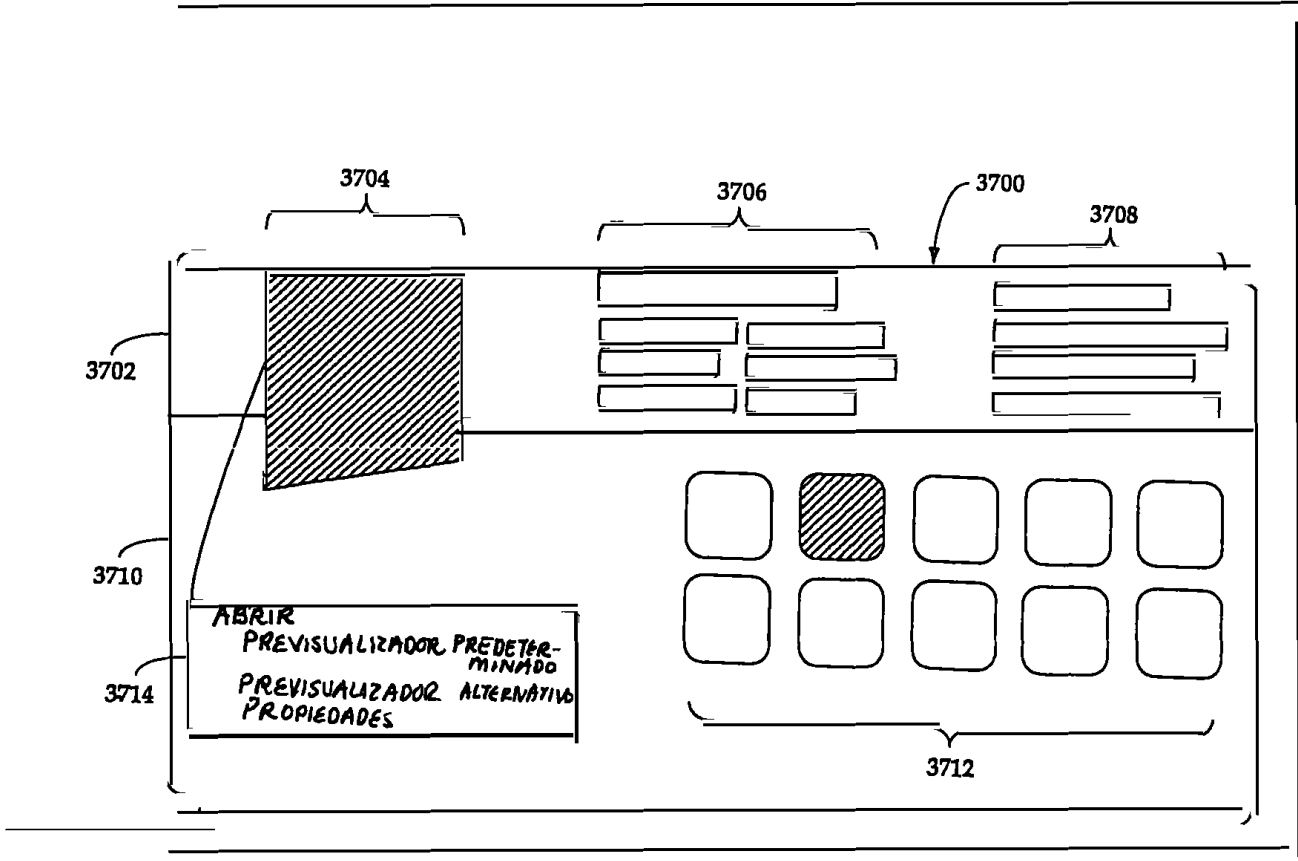
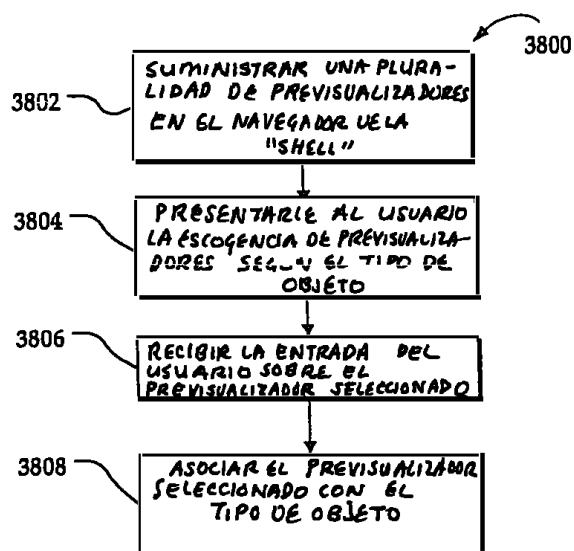


Fig.63.

240

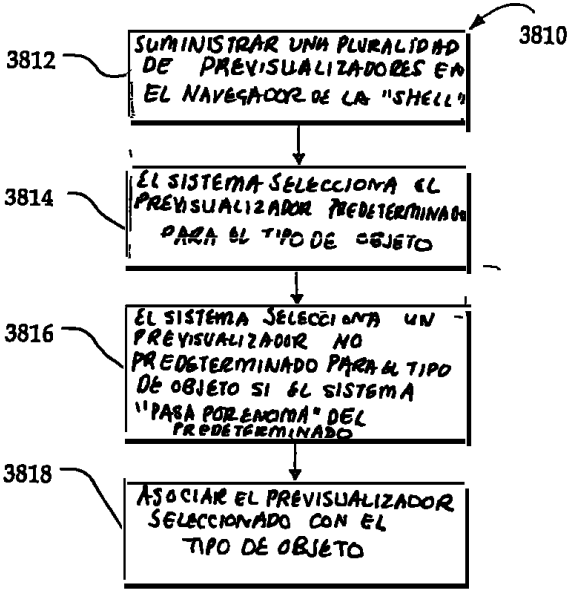


65/68

Fig.64A.

1

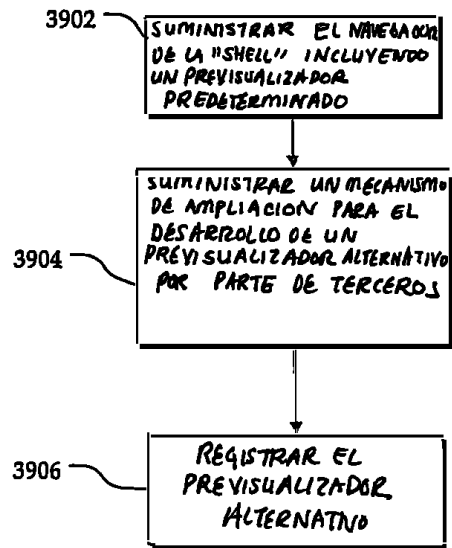
241



66/68

Fig.64B.

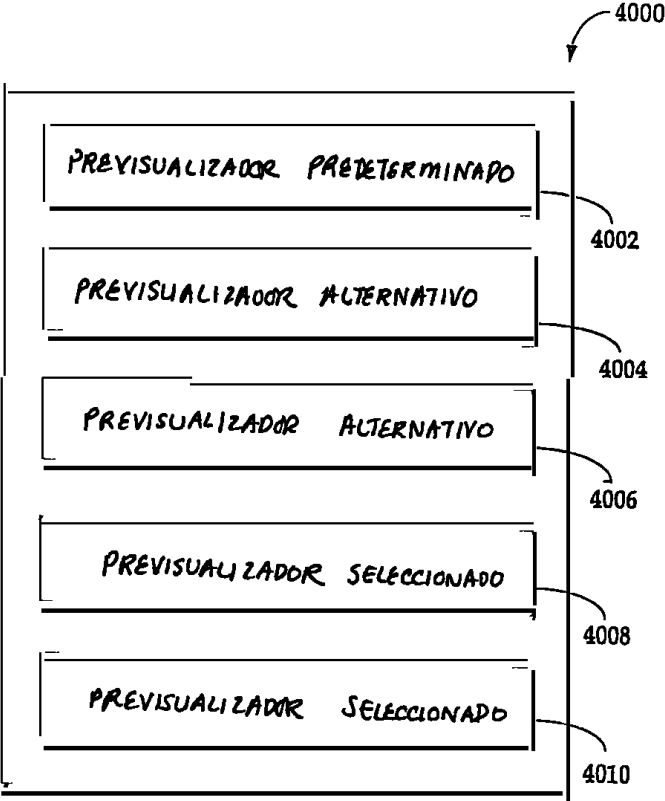
3900



67/68

Fig.65.

1 - - -



68/68

Fig.66.

7/176

FILE SYSTEM SHELL

FIELD OF THE INVENTION

5 The present invention relates to file systems, and more particularly, to a file system shell.

BACKGROUND OF THE INVENTION

Present computer file systems have a number of undesirable limitations. One limitation is that users are generally unable to control the structure that they are shown. In other words, when folders are organized, a user must choose a structure, and that structure is then difficult to change. As a specific example, for a "music" folder, a user may choose to organize the music files in an artist/album format, wherein all of the album folders for each artist are grouped into that particular artist's folder, and all of the songs on a particular album are grouped into that album's folder. The artist/album format is not conducive to playing a type of music (e.g., playing two jazz songs from two different artists), or for playing a selection of albums from different artists.

As another issue, a user may have a large number of files which are difficult to organize. Some users implement a rigid sense of placement for the files, and thus create strict hierarchies for them. The management of such files become increasingly complex and difficult as the number of available documents grows, making search and retrieval also difficult. This problem is further exacerbated when additional files are utilized from other locations, such as shared files, etc.

Users also have to deal with files being in different locations, such as on different devices, on other PCs, or online. For example, users can select to listen to their music on the computer (as may be accessible to a music program) or can go online and listen to music from Web sites, however there is a strict division between these two sources. Music coming from different locations is organized differently, and not kept in the same fashion or place. As another example, files stored on a corporate network may inherently be separated from files a user has on a current machine.

Users also have to keep track not only of what file data is stored, but where it is stored. For example, for music files, users are forced to keep copies on various systems and to try to track which music files are located where. This can make files difficult to locate, even when they are locally stored.

It is also sometimes difficult to find and return to files that a user has. A user may find it difficult to recall where and how they stored certain files. Given a set of folders and even a group of similar files, users often find it difficult to quickly find the one that they are looking for. For files stored in a difficult place to find, it is that much more complex to locate. In addition, once users have enough files in a folder, it becomes more difficult to parse the folder quickly, especially if the contents are similar.

It is also sometimes difficult for users to find or return to files on a network. Sharing and publishing files is often hard to do, and it may often be even more difficult to retrieve such a file from someone who makes it available. Users typically have to memorize or map the various sites and names that they need for finding files on a network.

Name spaces may vary, which can cause confusion to the user as to what is "correct." This is particularly true on a network where there are different naming conventions, limitations, and so on. For example, certain operating systems may require short names with no spaces in order for them to be visible.

Programs also often save files to their own directory or other name spaces, which can make it difficult for users to find their way back to the files. Programs often have default directories and places they save documents. A user often has to search through their hard disk and make guesses about where a file is stored.

Related items are also often stored in separate places. Related files that a user has may be stored on different parts of the hard disk, etc. This problem becomes more common with the developments of digital media services that have multiple content types (e.g., pictures, music, video).

Another issue with file systems is related to the address bar. As users navigate within a file system on a computer, a conventional graphical interface control, referred to as an address bar, shows the users where they are in the file system hierarchy. The conventional address bar shows the current location in terms of the file system's hierarchical structure of folders, subfolders, and files. Altering the user's location displayed in the conventional address bar is typically performed in one of two manners. The first is to manually edit the address in the address bar. Manually editing the address in the address bar permits a user to relocate to any number of locations in the file system hierarchy, but requires the user to have specific information regarding the organization of the file system on the computer, i.e., a specific file system location. The second method

involves using external navigation tools which, when manipulated, update the address bar to reflect the new address or location. While bypassing the manual edit of the address in the address bar, manipulating external navigation tools still requires the user to have specific information concerning the organization of the file system and traverse the hierarchical structure. However, conventional address bars cannot reference files or data stored among multiple file system locations, such as folders or drives, due to a one-to-one relationship between the address in the address bar and a specific location in the file system hierarchy.

The prior art lacks an address bar that allows users to specify addresses that display files stored among multiple file system locations. The prior art further lacks an address bar that also permits users to easily modify the address of the address bar without manually editing the address, or requiring specific knowledge concerning the organization of the underlying file system. Also lacking in the prior art is an address bar that presents alternative selections of files to the user from which the user may select to navigate to those selections of files. Such an address bar could also selectively present a conventional address bar interface to the user enabling the user to interact with the address bar according to previous experience according to user preferences.

Another issue with file systems is related to the identification of items stored on a computer. The need to readily identify items stored in a computing environment such as a personal computer (PC) is dramatically increasing as more individuals utilize computers in their daily routines and as the type of stored information varies between pictures, music, documents, etc. Documents and media are typically stored on computers in a hierarchical fashion and are organized with files of information or media stored within folders. File system browsers enable users to navigate through the file system and locate and open files and folders. For example, Microsoft Corporation's WINDOWS® EXPLORER™ is an operating system utility which enables users to browse the file system.

Many users find it difficult to correctly identify a file based on the information currently available in conventional file system browsers. Of course the contents of a file can be verified by opening it with an application program, but this method of browsing files is extremely inefficient. The ability to view metadata about a file within a file system browser can greatly assist a user in identifying a particular file without having to open it. In Microsoft Corporation's WINDOWS® 9X operating systems, for example, a

user can view object metadata by accessing the property sheet for a particular object. A property sheet presents the user with a list of the attributes or settings of an object in the form of a tabbed, index-card-like selection of property pages, each of which features standard dialog-style controls for customizing parameters. However, using the property sheet to locate an item can be slow and cumbersome, and some users find it difficult to locate the relevant metadata in a property sheet. Similarly, the use of infotips to locate an item can be slow and cumbersome because a user must hover the mouse over each file in order to view the limited metadata displayed in an infotip.

Conventional file system browsers do not allow users to enter and edit metadata relating to files and folders, which would significantly enhance a user's ability to later locate a file. To date, the ability of users to enter and edit metadata has been limited to special purpose software programs. For example, media players for electronic music files present users with the ability to edit metadata associated with music albums and artists. Another example of such programs includes application programs for electronic picture files. However, the utility of media players and other such programs is limited to the particular type of file supported by the program, as opposed to a general purpose file system browser which supports multiple file types.

Microsoft Corporation's WINDOWS® XP operating system includes an image browser for use in the My Pictures folder. The My Pictures folder is endowed with special features which enable users to view pictures as photos, not just as document icons. My Picture's image browsing features include the ability to view thumbnail-size and large versions of photos, rotate photos that are sideways, and create a slide show. A user can also view a photo's details, such as its dimensions, the date and time it was taken, and the name of the camera that took it. The preview control area in the My Picture's folder contains an enlarged preview image of a user-selected image, iterator buttons to assist a user in iterating through a series of pictures and controls for rotating pictures in a clockwise or counterclockwise direction. While the image browsing features in WINDOWS® XP have advanced the state of the art by alleviating the need to invoke an application program to view and manipulate pictures, users still cannot enter and edit metadata associated with the pictures.

Accordingly, there is a need for an improved user experience within a shell or file system browser which enables users to readily locate an item based on the metadata associated with that item. There is also a need for a system and method which allow

users to enter and edit metadata associated with items of various types within a shell browser without the need to invoke an application program. There is also a need for a file system or shell browser which offers users improved file content recognition features so that users can readily locate their files. A need also exists for an improved graphical user interface for a shell browser which allows for the selection of a previewer for a particular file type from a plurality of available previewers. There is also a need for an extensible shell browser which would allow software developers to provide additional information and functionality to users on a file type basis. There is also a need to provide a similar UI experience across different collections of items.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, a system and method utilizing virtual folders is provided. The virtual folders expose regular files and folders (also known as directories) to users in different views based on their metadata instead of the actual physical underlying file system structure on the disk. Thus, the system is able to take a property that is stored in the database and represent it as a container that is like a folder. Since users are already familiar with working with folders, by presenting the virtual folders in a similar manner, users can adapt to the new system more quickly.

In accordance with another aspect of the invention, the virtual folders are provided according to a method that is utilized in a computer system having a display and a memory for storing the items. In accordance with the method, a metadata property is selected. The system then searches for items that have the selected metadata property, and a virtual folder display object is provided that represents the collection of items that have the metadata property.

In accordance with another aspect of the invention, the system includes a folder processor that obtains queries from a user and a relational database for storing information about the items. The folder processor first obtains a query from a user and passes the query to the relational database. The relational database provides results back to the folder processor, and based on the results from the relational database, the folder processor provides the results to the user as virtual folders. In one embodiment, the results that are provided back to the folder processor include database rows and columns. The database rows and columns are converted by the folder processor into an enumerator structure, which is then used to populate the display with the resulting virtual folders.

In accordance with another aspect of the invention, users are able to work with the virtual folders through direct manipulation. In other words, the mechanisms that are provided for manipulating the virtual folders are similar to those that are currently used for manipulating conventional physical folders (e.g., clicking and dragging, copying, pasting, etc.).

In accordance with another aspect of the invention, the method for performing the direct manipulation of the virtual folders is provided in a computer system having a display and a memory for storing the items. In accordance with the method, groups of items are represented as virtual folders. Defined actions are provided that can be performed for direct manipulation of the virtual folders, wherein when a defined action is performed, the virtual folder is manipulated as directed by the defined action. An example of a defined action would be clicking and dragging a virtual folder. In one embodiment, the action of clicking and dragging a first virtual folder to a second virtual folder performs the function of copying the items from the first virtual folder to the second virtual folder. The copying of items to a virtual folder may involve adding or otherwise altering selected metadata properties that are associated with the items.

In accordance with another aspect of the invention, filters are provided for manipulating the virtual folders. The filters are essentially tools for narrowing down a set of items. In one embodiment, the filters are dynamically generated based on the properties of the separate items. For example, for a set of items, the filter mechanism may review the properties, and if the items generally have "authors" as a property, the filter can provide a list of the authors. Then by clicking on a particular author, the items that don't have the author disappear. This allows the user to narrow the contents.

In accordance with another aspect of the invention, a method for filtering items is provided in a computer system having a display and a memory for storing items with metadata properties. Display objects are provided on the display that each represent one or more items. The metadata properties of the items that are represented by the display objects are evaluated. A filter term is provided on the display that corresponds to a metadata property that is shared by a plurality of the items, wherein the selection of the filter term causes the items that are represented on the display to be reduced to those items that share the specified metadata property.

In accordance with another aspect of the invention, a plurality of items are represented on the display, and a filter term is dynamically generated based on the

metadata properties of the items. When the filter term is selected, it reduces the items that are represented on the display to those that have the metadata property that corresponds to the filter term.

5 In accordance with another aspect of the invention, a plurality of items are represented on the display, and a filter area is provided in which a user can enter a filter term. When a filter term is entered by the user, the items that are represented on the display are reduced to those that contain the filter term. As the user types the filter term, additional items may be filtered as each new character is added to the filter term.

10 In accordance with another aspect of the invention, a back button is provided which may be used to back through a filtering process. For example, after a user has entered a filter term, the user may wish to return to the set of items that were represented on the display before the filter term was applied. The back button allows the user to back up to the desired point in the filter navigation.

15 In accordance with another aspect of the invention, quick links are provided. In one embodiment, quick links are a set of predefined links (e.g., located on the left side of the display) that can be clicked on to generate useful views of the sets of items. These can be predefined by the program, or set by a user. For example, clicking on "all authors" could return a view stacked by authors. "All documents" may return a flat view of all the documents across all of the storage areas. Users can also create their own quick links.
20 For example, a user might filter down to all of the documents that they modified in January 2003, and then could save that as a quick link.

In accordance with another aspect of the invention, a method for providing quick links is implemented in a computer system having a display and a memory for storing items. In accordance with the method, a user first navigates to a view of a desired
25 collection of items. A quick link that corresponds to the desired collection of items is saved and is provided with a name. The name of the quick link is presented on the display such that by clicking on the quick link a user can return to the view of the desired collection of items.

In accordance with another aspect of the invention, libraries are provided.
30 Libraries consist of large groups of usable types of files that can be associated together. For example, photos may be one library, music may be another, and documents may be another. The libraries provide tools and activities that are related to the particular types

of items. For example, in the photo library, there are tools and filters that relate to manipulating photos, such as for creating slide shows or sharing pictures.

In accordance with another aspect of the invention, a method for creating libraries is provided in a computer system with a display and a memory for storing items. The method begins by creating a library to contain items with one or more specified metadata properties. Then, items with the one or more specified metadata properties are automatically grouped into the library. Tools are also provided for manipulating the items of the library.

In accordance with another aspect of the invention, a wide scope of files or items may be available. In other words, the system is able to represent files/items from multiple physical locations (e.g., different hard drives, different computers, different network locations, etc.) so that to a user all the items appear to be from one location. For example, a user can be presented with all of their music files on a single screen, and manipulate the files all from one view, even though the files may be physically stored on different hard drives, different computers, or different network locations.

In accordance with another aspect of the invention, a scope is utilized in a method for displaying items in a computer system having a display. The method involves defining a scope of the physical memory locations from which items are to be drawn, the scope comprising the present computer memory and at least one other physical location. Once a query is received, in response to the query items are drawn from the physical locations as defined in the scope, and the items that are drawn from the query are then presented in a view on the display. In one embodiment, the at least one other physical location may be another computer, a location on a network, or an external storage device. In one embodiment, the view on the display can be switched to a physical folder view which indicates the physical locations where the items are physically stored.

In accordance with another aspect of the invention, non-file items may be represented in the virtual folders. In other words, files that are stored in memory are located in a physical store. The virtual folders can be made to include items that are not currently represented in the physical store. Examples of non-file items are e-mails, and contacts.

In accordance with another aspect of the invention, a method for presenting non-file items is implemented in a computer system with a display and a memory for storing items. The method includes providing a database that allows both non-file items and file

items to be searched by a query. Once a query is received, both non-file items and file items that match the query are drawn, and the items that match the query are then presented on the display. In one embodiment, a relational database is provided that includes selected information about file items, and which may hold certain non-file items in their entireties.

5 In accordance with another aspect of the invention, a virtual address bar for selecting content stored on a computer file system is provided. A virtual address bar comprises a plurality of segments. Each segment corresponds to a filter for selecting content stored on the computer file system. Collectively, the corresponding filters of
10 each segment in the virtual address bar represent a virtual address for selecting content stored on a computer file system.

Each segment is an interactive segment that can respond to user interactions to modify the virtual address of the virtual address bar. Selecting a segment in the virtual address bar causes those segments subsequent to the selected segment to be removed
15 from the virtual address bar. Alternatively selecting a segment in the virtual address bar causes a list of selectable peer filters to be displayed to the user. The peer filters are peers to the alternatively selected segment's corresponding filter. Selecting one of the peer filters causes the alternatively selected segment to replace its corresponding filter with the selected peer filter. Additionally, those segments subsequent to the alternatively selected
20 filter segment are removed from the virtual address bar.

Segments may be added to the virtual address bar according to external user actions. Segments are added at the end of the segments in the virtual address bar. Any segments that conflict with an added segment are removed. An existing segment in the virtual address bar conflicts with the added segment when the existing segment is
25 mutually exclusive to the added segment. An existing segment in the virtual address bar also conflicts with the added segment when the existing segment is broader or narrower in scope than the added segment.

In accordance with another aspect of the invention, a shell browser is provided which includes a window and an edit control. The window displays a group of items and
30 also displays metadata values associated with one or more of the displayed items. The edit control permits user modification of at least a portion of the metadata values displayed in the window.

In accordance with another aspect of the invention, a graphical user interface is embodied on a computer-readable medium and is executable on a computer. The graphical user interface includes a first screen area which displays a set of items in a shell browser and a second screen area which displays metadata associated with one or more of the displayed items. The graphical user interface also presents the user with means within the shell browser for modifying the displayed metadata.

In accordance with a further aspect of the invention, computer-implemented methods are provided for enabling a user to modify metadata within a shell browser. One such method includes displaying a plurality of items, receiving a first input from the user representing a selection of at least one displayed item, displaying metadata associated with the selected item(s) and providing an edit control for user modification of the displayed metadata. Another such method includes displaying a welcome pane and metadata associated with the welcome pane and providing an edit control for user modification of the displayed metadata.

In accordance with another aspect of the invention, a data structure containing metadata associated with one or more items is displayed in a shell browser. The data structure, which is stored on one or more computer-readable media, includes a field containing user modifiable metadata associated with the one or more displayed items, and the user modifiable metadata contained in the data structure is also displayed in the shell browser.

In accordance with another aspect of the invention, a shell browser is provided which includes a default previewer and an extensibility mechanism. The default previewer provides a standard level of functionality for multiple item types. The extensibility mechanism enables functionality beyond the standard level provided by the default previewer for one or more of the item types.

In accordance with another aspect of the invention, a shell browser is provided which includes a first previewer and a second previewer. The first previewer provides a standard level of functionality for multiple item types, and the second previewer provides an alternative or extended level of functionality for one or more of the multiple item types. The shell browser is configured to selectively deploy either the first previewer or the second previewer for the one or more item types.

In accordance with another aspect of the present invention, a graphical user interface for a shell browser which supports multiple item types is provided. The

graphical user interface includes a first screen area for displaying a set of items in the shell browser and means for selecting a previewer for the displayed items from a plurality of available previewers.

5 In accordance with another aspect of the invention, a computer-implemented method is provided for selecting a previewer in a shell browser which supports multiple item types. The method includes providing a plurality of previewers in the shell browser for a particular item type and selecting one of the previewers for the particular item type. The method then associates the selected previewer with the particular item type.

10 In accordance with another aspect of the invention, a computer-implemented method is provided for enabling the use of third party previewers in a shell browser which supports multiple item types. The method includes providing a shell browser having a default previewer for the multiple item types and providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

15 In accordance with another aspect of the invention, a data structure is provided which contains information indicative of a plurality of previewers in a shell browser. The data structure, which is stored on one or more computer-readable media, includes a first field containing information indicative of a default previewer which supports multiple item types. A second field contains information indicative of an alternative previewer for
20 a first item type, and a third field contains information indicative of whether to invoke the default previewer or the alternative previewer when items of the first item type are displayed in the shell browser.

In accordance with another aspect of the invention, different types of items are grouped into libraries for which a similar set of basic UI features are provided. In other
25 words, a similar set of basic UI features is provided for different types of libraries, such as a document library, a photo library, and a music library. This set of basic UI features may include features such as filtering, creating new categories, editing the metadata of the items, altering the pivots, etc. The similar set of basic UI features for the libraries allows a user to process and organize different types of items using attributes and features
30 they are already familiar with.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a block diagram of a general purpose computer system suitable for implementing the present invention;

FIGURE 2 is a block diagram of a virtual folder system in accordance with the present invention;

FIGURE 3 is a flow diagram illustrative of a routine by which a user provides a query that draws back selected files and folders;

FIGURE 4 is a flow diagram illustrative of a routine by which virtual folders are constructed and displayed on the screen in accordance with either a default query or a query from the user;

FIGURE 5 is a tree diagram of a folder structure in accordance with a physical folder arrangement on a hard drive;

FIGURE 6 is a tree diagram of a virtual folder structure;

FIGURE 7 is a tree diagram of the virtual folder structure of FIGURE 6, wherein the clients stack is further filtered by contracts and year;

FIGURE 8 is a tree diagram of the virtual folder structure of FIGURE 7, wherein the contracts of the clients stack are further filtered by year;

FIGURE 9 is a tree diagram of the virtual folder structure of FIGURE 6, wherein the contracts stack is further filtered by clients and year, of which the clients are still further filtered by year;

FIGURE 10 is a diagram illustrative of a screen display showing the stacks of a document library;

FIGURE 11 is a diagram illustrative of a screen display showing the documents in the ABC Corp. stack of FIGURE 10;

FIGURE 12 is a diagram illustrative of a screen display in which a stacking function is selected for the documents of FIGURE 11;

FIGURE 13 is a diagram illustrative of a screen display in which a "stack by author" parameter is selected for the stacking function of FIGURE 12;

FIGURE 14 is a diagram illustrative of a screen display in which the files of FIGURE 13 have been stacked by author;

FIGURE 15 is a diagram illustrative of a screen display in which a stacking function is selected and a "stack by category" option is further selected for restacking the
5 files of FIGURE 14;

FIGURE 16 is a diagram illustrative of a screen display in which the files of FIGURE 14 have been restacked by category;

FIGURE 17 is a diagram illustrative of a screen display in which a quick link for showing physical folders is selected;

10 FIGURE 18 is a diagram illustrative of a screen display in which the physical folders are shown which contain the files of the virtual folder stacks of FIGURE 17;

FIGURE 19 is a flow diagram illustrative of a routine by which a user can directly manipulate virtual folders;

FIGURE 20 is a diagram illustrative of a screen display in which a new "West
15 Coast" stack has been added to the stacks of FIGURE 10;

FIGURE 21 is a diagram illustrative of a screen display in which direct manipulation is used for copying the files from the "ABC Corp." stack to the "West Coast" stack of FIGURE 20;

FIGURE 22 is a flow diagram illustrative of a routine for the system dynamically
20 generating new filter terms;

FIGURE 23 is a flow diagram illustrative of a routine for the system filtering items based on the selection of a filter term;

FIGURE 24 is a diagram illustrative of a screen display in which the stacks of FIGURE 10 have been filtered by the term "AB";

25 FIGURE 25 is a diagram illustrative of a screen display in which the stacks of FIGURE 10 have been filtered by the term "ABC";

FIGURE 26 is a diagram illustrative of a screen display in which the filter term "year 2002" is selected for the stacks of FIGURE 10;

FIGURE 27 is a diagram illustrative of a screen display in which the stacks of
30 FIGURE 10 have been filtered by the "year 2002" and the further selection of the filter term "month";

FIGURE 28 is a diagram illustrative of a screen display in which a list is presented for selecting a month for filtering;

FIGURE 29 is a diagram illustrative of a screen display wherein the stacks of FIGURE 10 have been further filtered by the month of January, and further showing a filter term of "day";

5 FIGURE 30 is a flow diagram illustrative of a routine for creating a new quick link;

FIGURE 31 is a diagram illustrative of a screen display for creating a new quick link called "January Work" based on the filtering of FIGURE 29;

FIGURE 32 is a diagram illustrative of a screen display in which a quick link of "All Authors" is selected;

10 FIGURE 33 is a diagram illustrative of a screen display in which a list of all of the authors of FIGURE 32 is presented;

FIGURE 34 is a diagram illustrative of a screen display in which "Author 1" has been selected from the list of FIGURE 33 and all of the Author 1's documents are shown;

FIGURE 35 is a flow diagram illustrative of a routine for creating a new library;

15 FIGURE 36 is a diagram illustrative of a screen display in which a collection of various available libraries are shown;

FIGURE 37 is a flow diagram illustrative of a routine for defining the scope of a virtual folder collection;

20 FIGURE 38 is a block diagram illustrative of the various sources which may form the scope of a virtual folder collection;

FIGURE 39 is a flow diagram illustrative of a routine for including non-file items in a virtual folder collection;

FIGURE 40 is a diagram illustrative of a screen display showing various non-file items included in a virtual folder;

25 FIGURE 41 is a pictorial diagram of an exemplary networked computer environment suitable for implementing the present invention;

FIGURE 42 is a pictorial diagram illustrating an exemplary file viewer having a conventional address bar associated with displaying files in a computer file system, as found in the prior art;

30 FIGURE 43 is a pictorial diagram illustrating an exemplary file viewer for displaying files in a computer file system in accordance with a virtual address in a virtual address bar formed in accordance with the present invention;

FIGURE 44A is a pictorial diagram of the exemplary file viewer of FIGURE 5 illustrating selecting a segment of the virtual address in the virtual address bar to navigate in the file system;

FIGURE 44B is a pictorial diagram of the exemplary file viewer of FIGURE 45A illustrating the results of selecting a segment of the virtual address in the virtual address bar;

FIGURES 45A-45D are pictorial diagrams illustrating selecting a peer filter associated with a segment of a virtual address in a virtual address bar;

FIGURES 46A-46D are pictorial diagrams illustrating adding additional filters to a virtual address in a virtual address bar;

FIGURES 47A and 47B are pictorial diagrams illustrating an exemplary virtual address bar displaying a virtual address where the virtual address exceeds the virtual address bar's display capacity;

FIGURE 48A is a pictorial diagram illustrating an exemplary virtual address bar having a virtual address with filters referencing both virtual and actual locations in a file system;

FIGURE 48B is a pictorial diagram illustrating the exemplary virtual address bar of FIGURE 48A as configured to display a conventional address bar;

FIGURE 49 is a flow diagram illustrative of an alternate filter selection routine for selecting alternate filters in a virtual address bar;

FIGURE 50 is a flow diagram illustrating an exemplary add filter routine for adding a filter to a virtual address in a virtual address bar;

FIGURE 51A is a block diagram of an exemplary graphical user interface for a shell browser having an edit control in accordance with an embodiment of the present invention;

FIGURE 51B is a block diagram of an exemplary graphical user interface for a shell browser having one or more edit controls in accordance with an embodiment of the present invention;

FIGURE 52 is a schematic diagram of a welcome pane in a shell browser;

FIGURE 53 is a schematic diagram of a selected pane in a shell browser;

FIGURE 54 is a schematic diagram of the selected pane of FIGURE 53 including a context menu enabling a user to modify metadata in a shell browser in accordance with an embodiment of the present invention;

FIGURE 55 is a flow diagram illustrating a method for enabling a user to modify metadata displayed in a welcome pane within a shell browser in accordance with an embodiment of the present invention;

FIGURE 56 is a flow diagram illustrating a method for enabling a user to modify metadata displayed in a selected pane within a shell browser in accordance with an embodiment of the present invention;

FIGURE 57 is a block diagram of a data structure containing user modifiable metadata associated with an item displayed in a shell browser;

FIGURE 58 is a schematic diagram of a prior art graphical user interface for browsing pictures stored in a folder within a shell browser environment which is used for viewing other non-pictorial files and folders;

FIGURE 59 is a block diagram of an exemplary graphical user interface for a shell browser;

FIGURE 60 is a schematic diagram of a welcome pane in a shell browser;

FIGURE 61 is a schematic diagram of a selected pane in a shell browser;

FIGURE 62 is a schematic diagram of a selected pane in a shell browser with extended controls in accordance with an embodiment of the present invention;

FIGURE 63 is a schematic diagram of a selected pane similar to FIGURE 61 but including a context menu enabling a user to select a previewer in a shell browser in accordance with an embodiment of the present invention;

FIGURE 64A is a flow diagram illustrating a method for enabling a user to select a previewer in a shell browser in accordance with an embodiment of the present invention;

FIGURE 64B is a flow diagram illustrating a method for enabling the system to select a previewer in a shell browser in accordance with an embodiment of the present invention;

FIGURE 65 is a flow diagram illustrating a method for enabling the use of third party previewers in a shell browser in accordance with an embodiment of the present invention; and

FIGURE 66 is a block diagram of a data structure containing information indicative of multiple previewers in a shell browser.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention is directed to a file system shell which incorporates a number of desirable features. In essence, the shell provides users with the ability to view and manipulate files and other items that are stored on a computer. The following
5 description first provides a summary of the features that are shown in the FIGURES 1-66, and then provides a detailed discussion.

In summary, FIGURES 1-9 are generally directed to an overall system for virtual folders. Virtual folders provide a method for allowing a conventional user interface to expose regular files and folders (also known as directories) to users in different views
10 based on their metadata instead of the actual physical underlying file system structure on the disk. FIGURES 10-18 are generally directed to stacks, which are related to the ability of the virtual folders to take any property that is stored in the database and represent it as a container that is like a folder. FIGURES 19-21 are generally directed to direct manipulation of virtual folders, which relates to providing mechanisms for manipulating
15 virtual folders that are similar to the mechanisms currently used for manipulating standard folders (e.g., copying, pasting, clicking and dragging, etc.). FIGURES 22-29 are generally directed to filters, which provide a set of tools for narrowing down a set of files/items. FIGURES 30-34 are generally directed to quick links, which are a set of predefined links that can be clicked on to generate useful views of sets of files/items.
20 FIGURES 35-36 are generally directed to libraries, which are related to the concept that groups of usable types of files can be associated together, and that tools and activities that are related to the particular types of items can be provided. FIGURES 37-38 are generally directed to scope which is related to the concept of being able to acquire files/items from multiple physical locations (e.g., different hard drives, different
25 computers, from a computer in a network location, etc.) so that to the user all the files/items are presented with the same convenience as if they were being provided from one location. FIGURES 39-40 are generally directed to non-file items, which can be included in the database along with files, and which can include items such as emails and contacts. FIGURES 41-50 are generally directed to a virtual address bar which comprises
30 a plurality of segments, each segment corresponding to a filter for selecting content. FIGURES 51-57 are generally directed to a shell browser, with which users can readily identify an item based on the metadata associated with that item. FIGURES 58-66 are generally directed to extending the functionality of an object previewer in a shell browser

configured to display a plurality of items representing multiple item types. The following description provides a detailed discussion of each of these aspects of the invention.

As noted above, FIGURES 1-9 are generally directed to a system for implementing virtual folders. Virtual folders utilize the same or similar user interfaces that are currently used for file systems. The virtual folders expose regular files and folders (also known as directories) to users in different views based on their metadata instead of the actual physical underlying file system structure on the disk. Location-independent views are created which allow users to manipulate their files and folders utilizing similar controls as those presently used for managing file systems. In general, this means that users can organize and rearrange their files based on inherent properties in the files themselves, instead of the managing and organization being done as a separate part of the system. The virtual folders may represent files or items from different physical locations, such as from multiple disk drives within the same computer, between multiple computers, or different network locations, such that one view of files or items can expose files or items sitting at different physical locations. In one embodiment, the different items or files need only be connected via an IP network in order to be included.

The virtual folder modeling is also able to be used for traditionally non-file entities. An application of this is to have a set of user interfaces similar to files and folders (that is, objects and containers) to show traditionally non-file entities. One example of such non-file entities would be e-mails, while another would be contact information from a contact database. In this manner, virtual folders provide for a location-independent, metadata-based view system that works regardless of whether the data being shown is from files or non-file entities. In general, these aspects allow more flexibility in terms of letting users manipulate their files and data, using both common user interface techniques (drag and drop, double-click, etc.) as well as leveraging the rich integration of various data types.

FIGURE 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the virtual folders, and any of the other aspects of the present invention that are described herein, may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, characters, components, data structures, etc., that perform particular tasks or implement particular

abstract data types. As those skilled in the art will appreciate, the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may
5 also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to FIGURE 1, an exemplary system for implementing the
10 invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, system memory 22, and a system bus 23 that couples various system components including the system memory 22 to the processing unit 21. The system bus 23 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
15 of a variety of bus architectures. The system memory includes read-only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that helps to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer 20 further includes a hard disk drive 27 for reading from or
20 writing to a hard disk 39, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31, such as a CD-ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical
25 drive interface 34, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer-readable instructions, data structures, program modules, and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk 39, a removable magnetic disk 29, and a removable optical disk 31, it should be appreciated by those skilled in the art that other
30 types of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read-only memories (ROMs), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk 39, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37 and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. 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 21 through a serial port interface 46 that is coupled to the system bus 23, but may also be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A display in the form of a monitor 47 is also connected to the system bus 23 via an interface, such as a video card or adapter 48. One or more speakers 57 may also be connected to the system bus 23 via an interface, such as an audio adapter 56. In addition to the display and speakers, personal computers typically include other peripheral output devices (not shown), such as printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more personal computers, such as a remote computer 49. The remote computer 49 may be another 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 personal computer 20. The logical connections depicted in FIGURE 1 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the local area network 51 through a network interface or adapter 53. When used in a WAN networking environment, the personal computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20 or portions thereof may be stored in the remote memory storage device. 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.

As implemented on a system of the type illustrated in FIGURE 1, the present invention utilizes virtual folders which make it easier for users to perform basic tasks around file manipulation and folder navigation (browsing) and to provide higher level storage capabilities which can be leveraged in new features. The virtual folders expose
5 files and items to users in different views based on their metadata instead of the actual physical underlying file system structure on the disk.

FIGURE 2 is a block diagram of a virtual folder system 200 in accordance with the present invention. As will be described in more detail below, the virtual folders allow a user to change the "pivot" which controls the way the data is viewed. As an example, a
10 user could view their music as a flat list of all the songs, which can be grouped by album. Alternatively, the user could switch the view to show only the genres or artists or years, etc. The user can tailor the view to see only the objects suited to the task at hand. This allows an improved browsing experience that negates the need for further navigation through folders (both down and back up). The same lessons and capabilities apply to
15 modeling other data-types not stored as files. Contacts, for example, can be exposed to the user in this way, giving them familiar interface capabilities, as well as richer infrastructure for manipulating them than is provided by a flat address book.

As illustrated in FIGURE 2, the virtual folder system 200 includes a folder processor 210, a relational database 230, a virtual folder descriptions database 232, an
20 other shell folders component 234, a folder handler's component 236, and a shell browser and view component 240. The folder processor 210 includes a native handling code component 212, a handler factory component 214, a property writer component 216, a rowset parser component 218, a query builder component 220, an enumerator component 222, and a property factory component 224.

25 The relational database 230 stores properties about all files in the system. It also stores some items, like contacts (i.e., non-file items), entirely. In general, it stores metadata about the types of files and items that it contains. The relational database 230 receives SQL queries from the query builder 220. The relational database 230 also sends SQL rowsets to the rowset parser component 218, with one row per item column,
30 columns being the item properties.

The virtual folder descriptions database 232 includes the virtual folder descriptions. The virtual folder descriptions database 232 sends data to the query builder

component 220, including a list of types to display in the folder, the initial filter, and the physical locations to show results from (the scopes).

With regard to the other shell folders component 234, the folder processor 210 delegates to existing shell folders from many types of items, including all files, for handlers or properties. The other shell folders component 234 sends properties from
5 other folders to the property factory 224. The other shell folders component also sends handlers to the handler factory 214.

The folder handlers component 236 provides code behavior for the items that exist only in the database, like contacts. This is what allows non-file items to behave akin to
10 files. The folder handlers component 236 sends handlers to the handler factory 214.

For the native handling code component 212, the folder processor 210 directly implements certain handlers based on the properties of the items. The native handling code component 212 sends handlers to the handler factory 214. For the native handling code component 212 and the folder handlers component 236, like all namespaces, virtual
15 folders have to provide a set of handlers (context menu, icon, thumbnail, infotip, . . .) for their items. For most of these (infotip, data object, drag-drop handler, background context menu . . .) the virtual folder provides a common (native) handler for all the types it holds. However there are others which the author of the type has to provide (context menu on the item itself, writable property store, . . .). The default handler can also be
20 overridden. Virtual folders reuse this for files and allow non-file items do the same.

The handler factory 214 takes ID lists and produces code behaviors that provide context menus, icons, etc. In general, the folder processor 210 may use native handlers, external handlers, or delegate to other shell folders to get handlers, as described above with respect to the native handling code component 212, the other shell folders
25 component 234, and the folder handlers component 236. The handler factory component 214 sends handlers to the shell browser in view 240, as requested by the view. The handler factory component 214 sends a property handler to the property writer 216.

The property writer 216 converts user intentions such as cut, copy, and paste into property rights to the file or item. A shell browser and view component 240 sends data to
30 the property writer 216, including direct manipulation (cut/copy/paste) or editing of metadata. In general, since virtual folders present an organization based on the properties of an item, operations such as move and copy (drag-drop) become an edit on those properties. For example, moving a document, in a view stacked by author, from Author 1

to Author 2, means changing the author. The property writer component 216 implements this function.

The rowset parser 218 takes database rowsets and stores all item properties into a shell ID list structure. A rowset takes the piecewise definition of the virtual folder and
5 builds a SQL string which can then be issued to the database. The rowset parser component 218 sends ID lists to the enumerator component 222. As described above, the rowset parser component 218 also receives data from the relational database 230, including SQL rowsets, with one row per item, the columns being item properties.

The query builder component 220 builds SQL queries. The query builder
10 component 220 receives data from the enumerator component 222, including new filters from the navigation. The query builder component 220 also receives data from the virtual folder descriptions database 232, including a list of the types to display in the folder, the initial filter, and the physical location to show results from (the scopes). The query builder component 220 sends the SQL queries to the relational database 230.

15 In general, the query builder component 220 includes a set of rows (in other words a table). This is what running the query yields. The rowset parser component 218 takes each row and using the column names transforms the row into an ID list. An ID list is a well-known shell structure which is used to reference items in a namespace. Doing this allows virtual folders to be just like any other namespace to the rest of the
20 shell. Also caching this data helps keep database access, which can be costly, to a minimum.

The enumerator component 222 operates in response to a navigation to a virtual folder. As described above, the enumerator component 222 receives ID lists from the rowset parser component 218, and sends new filters from the navigation to the query
25 builder component 220. The enumerator 222 also sends data to the shell browser and view component 240, including ID lists that are returned to be inserted into the view after a navigation.

The property factory component 224 takes ID lists and property identifiers and returns values for those properties. The property factory component 224 receives data
30 from the handler factory component 214 including the property handler. As described above, the property factory component 224 also receives data from the other shell folders component 234, including properties from other folders. The property factory

component 224 also sends data to the shell browser and view component 240, including item properties, as requested by the view.

The shell browser and view component 240 displays the contents of a folder in a window, and handles all the user interaction with the displayed files or items, such as clicking, dragging, and navigating. Thus, the shell browser and view component 240 receives the user actions. The shell browser and view component 240 also gets the data regarding the code behaviors that it needs from the folder, in this case the folder processor 210.

As described above, the virtual folders expose regular files and folders (also known as directories) to users in different views based on their metadata instead of the actual physical underlying file system structure on the disk. Thus, the system is able to take a property that is stored in the database and represent it as a container that is like a folder. Since users are already familiar with working with folders, by presenting the virtual folders in a similar manner, users can adapt to the new system more quickly.

FIGURE 3 is a flow diagram illustrative of a routine 300 by which a user provides a query that draws back selected items. At a block 302, the folder processor gets a query from the user. In a block 304, the folder processor passes the query to the relational database. At a block 306, the relational database provides the results back to the folder processor. At block 308, the folder processor provides the results to the user in the form of virtual folders and items.

FIGURE 4 is a flow diagram illustrative of a routine 320 by which virtual folders are constructed and displayed on the screen in accordance with either a default query or a query from the user. At a block 322, when a user first opens the virtual folder, a default query is used. This default query is taken from the registry. For example, the default query for a music library could be to show all the songs grouped by album. At a block 324, the folder processor constructs a query object for this query, and then passes this query to the relational database. At a block 326, the relational database generates the results of the query and passes these back to the folder processor as database rows and columns.

At a block 328, the folder processor takes these results and converts them from the rows and columns of data into an enumerator structure, which is used by the folder view to populate the screen with the resulting virtual folders and items for the user to interact upon. At a decision block 330, a user decides whether to change the view (by

issuing a different query or "pivot"). For example, a user could issue a "show all artists" pivot. If the user does want to change the view, then the routine returns to block 324 where the folder processor passes this new query to the relational database, and receives back new rows and columns of results, and constructs a new enumerator structure. The process then continues as described above, as the folder view clears and updates, using the enumerator to draw the "artist" objects to the screen.

In one example, album objects are provided that represent containers that users can navigate into. For example, double-clicking the "Beatles" albums will navigate the view to see all of the Beatles' songs. The folder processor issues the "show all Beatles' songs" query to the relational database, which hands back the rows and columns of data for those songs. The folder processor creates an enumerator of all these songs, which then get drawn to the screen.

The user can also choose the view at any point while browsing virtual folders. From the above example, after narrowing down to just show Beatles songs, a user can change the view to only show the songs as albums. The process of changing the view of items into another representation is called "stacking". This is because the items are conceptually arranged into "stacks" based on that representation. In this case, the songs are rearranged into stacks for each of the various albums. Users can then navigate into one of these stacks, only seeing the songs from that particular album. Again, the user can rearrange the view of these remaining songs into stacks based on a property (e.g., a rating, for example). If the rating property were selected, the songs from that Beatles album would be shown in stacks for a one-, two-, or a three-star rating.

The results of each query depend on which physical locations are included in the scope. For example, the scope may be made to include only the folders in the user's "my documents" folder. Alternatively, the scope could include all folders on the computer, or even all folders on multiple network connected computers. The user is able to view and change the scope through a scope property sheet. In one example, the scope property sheet could be exposed by right-clicking on the virtual folder and choosing "properties." The user could add new folders to the scope, or remove folders that were previously added.

One group of users for which virtual folders will provide particular utility is knowledge workers. Virtual folders allow knowledge workers to easily switch between viewing documents by file type, project, case number, author, etc. Since knowledge

workers each tend to have a different method for organizing documents, virtual folders can be used to accommodate these different preferences.

FIGURE 5 is a tree diagram of a folder structure in accordance with a physical folder arrangement on a hard drive. This physical folder arrangement is based on the traditional implementation of folders, which may be based on NTFS or other existing file systems. Such folders are referred to as physical folders because their structuring is based on the actual physical underlying file system structure on the disk. As will be described in more detail below, this is in contrast to virtual folders, which create location-independent views that allow users to manipulate files and folders in ways that are similar to those currently used for manipulating physical folders.

As illustrated in FIGURE 5, a folder 400 is a "my documents" folder. At a first level, the folder 400 includes folders 410, 420, and 430, corresponding to Clients 1, 2, and 3, respectively. At a second level, each of the folders 410, 420, and 430 contain a folder 411, 421, and 431, respectively, which each correspond to the contracts for the selected client. At a third level, each of the folders 411, 421, and 431 contains a folder 412, 422, and 432, respectively, each corresponding to the year 2001. At the third level, each of the folders 411, 421, and 431 also contains a folder 413, 423, and 433, respectively, each corresponding to the year 2002.

It will be appreciated that a number of obstacles are presented to a user who wishes to navigate a physical folder file structure such as that illustrated in FIGURE 5. For example, if the user wishes to work with all of the contracts that the user has produced, the user will first need to navigate to the folder 411 to work with the contracts for Client 1, and then will have to renavigate to the folder 421 to reach the contracts for Client 2, and will again have to renavigate to the folder 431 for the contracts for Client 3. This arrangement makes it difficult for the user to access all of the contracts, and in general prevents simultaneous viewing and manipulation of all of the contracts. Similarly, if the user wishes to view all of the contracts produced in the year 2001, the user will have to navigate and renavigate to the folders 412, 422, and 432, respectively. As will be described in more detail below, the virtual folders of the present invention provide an improved file system structure.

FIGURE 6 is a tree diagram of a virtual folder structure. As will be described in more detail below, virtual folders create location-independent views that allow users to manipulate their files and folders in convenient ways. As shown in FIGURE 6, the

virtual folders are represented as stacks. A virtual folder 500 is an "all items" folder. At a first level, the virtual folder 500 contains virtual folders 510, 520, and 530, corresponding to clients, contracts, and year, respectively. As will be described in more detail below, this structure allows a user to access files according to a desired parameter.

5 FIGURE 7 is a tree diagram of the virtual folder structure of FIGURE 6, wherein at a second level, the virtual folder 510 further includes virtual folders 511 and 512, which correspond to contracts and year, respectively. In other words, the clients stack of virtual folder 510 is further filtered by contracts and year. The process for determining which files and items are contained in each of the virtual folders will be described in
10 more detail below.

FIGURE 8 is a tree diagram of the virtual folder structure of FIGURE 7, wherein at a third level, the virtual folder 511 contains a virtual folder 513, which corresponds to a year. In other words, the contracts stack of virtual folder 511 is further filtered by year. While the virtual folder structure for the virtual folders 510, 511, and 513 have been
15 structured according to clients, contracts, and year, it will be appreciated that the virtual folders allow for other structuring sequences to occur, as will be described in more detail below with reference to FIGURE 9.

FIGURE 9 is a tree diagram of the virtual folder structure of FIGURE 6, wherein at a second level, the virtual folder 520 has been further filtered into virtual folders 521 and 522, corresponding to clients and year. At a third level, the virtual folder 521 has
20 further been filtered to a virtual folder 523, corresponding to a year. The contrast between the organizational structures of FIGURES 8 and 9 helps illustrate the flexibility of the virtual folder system. In other words, in a virtual folder system, a user is able to navigate the virtual folders according to desired parameters, as opposed to being
25 dependent on the location-dependent views of a physical file structure such as that illustrated in FIGURE 5.

FIGURE 10 is a diagram illustrative of a screen display 600 showing the stacks of a document library. As noted above, stacks can be used to represent a type of virtual folder. As will be described in more detail below, the screen display 600 includes quick
30 link elements 610-613, filter elements 620-626, activity elements 630-633, information and control elements 640-645, and virtual folder stacks 651-655.

The quick link elements include an "all categories" quick link 610, on "all authors" quick link 611, a "January work" quick link 612, and a selection for displaying

additional quick links 613. As will be described in more detail below, quick links can be selected by a user to perform desired navigations of the virtual folders. Quick links may be provided by the system, and some quick links may be created and saved by a user.

5 The filter elements include a "filter by" indicator 620, an entry blank 621, a "by date" indicator 622, a "year" selector 623, a "pick an author" selector 624, a "pick a category" selector 625, and a "more filters" selector 626. The "filter by" indicator 620 directs a user to the fact that the items below can be used to filter the virtual folders or items. The entry blank 621 provides an area in which a user can type a desired new filter term. The "by date" indicator 622 directs a user to the fact that by selecting a date from
10 the "year" selector 623, the virtual folders or items can be filtered by the selected year. The "pick an author" selector 624 allows a user to filter according to a specific author. The "pick a category" selector 625 allows a user to filter according to a selected category. The "more filters" selector 626 allows a user to pull up additional filters on the display.

The activity selectors include a "create a new category" selector 630, "activity" selectors 631 and 632, and a "more activities" selector 633. As will be described in more
15 detail below, the activities that are presented may be for generally desirable functions, or may more specifically be directed to activities useful for the type of virtual folders that are currently being displayed. For example, the "create a new category" selector 630 can be selected by the user to create a new category which will be represented by a new stack.

20 As noted above, the activity selectors 631 and 632 may be more specifically directed to the type of folders or items that are being displayed. For example, the present display is of a document library, for which the "activity" selectors 631 and 632 may be directed to activities specifically tailored for documents, such as editing or creating attachments. If the present library had been a photo library, the "activity" selector 631
25 and 632 could be for activities specifically directed to photos, such as forming photo albums or sharing photos with other users.

The information and control elements include information lines 640 and 641, a control line 642, a backspace control 643, and information lines 644 and 645. The information lines 640 and 641 provide information as to the current navigation of the
30 virtual folders or items. In the present example, the information line 640 indicates that the current navigation is to a document library, while the information line 641 indicates the more complete navigation, showing that the document library is within the storage area. The control line 642 provides a number of standard controls, and the backspace

button 643 allows a user to back up through a navigation. The information line 644 provides numerical information about the contents of the present navigation. In the present example, the information line 644 indicates that there are 41 items which take up 100 MB in the stacks of the document library. The information line 645 is available to provide additional information, such as additional information about a file that is selected.

The stacks of the document library include an "ABC Corp." stack 651, a "backups stack" 652, a "business plans" stack 653, an "XYZ Corp." stack 654, and a "marketing reports" stack 655. The numbers on top of each of the stacks indicate how many items are in each stack. For example, the "ABC Corp." stack 651 is shown to include 8 items. The total number of items of the stacks adds up to the number of items indicated in the information line 644, which as described above is 41 in the present example. A selection box SB is provided which can be utilized by a user to select a desired item. The selection of the "ABC Corp." stack 651 yields a view of the items of that stack, as will be described below with respect to FIGURE 11.

FIGURE 11 is a diagram illustrative of a screen display showing the items in the "ABC Corp." stack 651 of FIGURE 10. It should be noted that the information lines 640 and 641 now indicate that the present navigation is showing the "ABC Corp." stack. The "ABC Corp." stack 651 is shown to include 8 documents 751-758, corresponding to documents 1-8, respectively. The information line 644 correspondingly indicates that there are 8 items which take up 20 MB of memory. Documents of FIGURE 11 may be further arranged into stacks within the ABC Corp. stack. In other words, within the virtual folder represented by the ABC Corp. stack 651, additional virtual folders may be organized to hold the documents, as will be described below with respect to FIGURES 12-16.

FIGURE 12 is a diagram illustrative of a screen display in which a stacking function is selected for the documents of FIGURE 11. As shown in FIGURE 12, the user is able to pull up a function box 760. The function box 760 includes a "view" selection 761, an "arrange icons by" selection 762, a "stacks" selection 763, a "refresh" selection 764, an "open containing folders" selection 765, a "cut" selection 766, a "copy" selection 767, an "undo" selection 768, a "new" selection 769, and a "properties" selection 770. The selection box SB is shown to be around the "stacks" selection 763.

FIGURE 13 is a diagram illustrative of a screen display in which a "stack by author" parameter is selected for the stacking function of FIGURE 12. As shown in

FIGURE 13, a box 780 is displayed which presents various stacking options. The stacking options include an "unstack" option 781, a "stack by category" option 782, a "stack by author" option 783, and a "stack by a user" option 784. The selection box SB is shown to be around the "stack by author" option 783.

5 FIGURE 14 is a diagram illustrative of a screen display in which the files of FIGURE 13 have been stacked by author. As shown in FIGURE 14, stacks 791 and 792 correspond to authors Bob and Lisa, respectively. As indicated by the numbers on top of each of the stacks, the Bob stack 791 includes two items, while the Lisa stack 792 includes five items. The item 758 (corresponding to document 8) did not have an author, and so is not included in an "author" stack. The stacks 791 and 792 illustrate that stacks
10 may be organized at multiple levels, such as within the "ABC Corp." stack 651. Thus, the virtual folders may be formed at multiple levels, such as the "Lisa" stack 792 being within the "ABC Corp." stack 651 which is within the document library.

FIGURE 15 is a diagram illustrative of a screen display in which a "stack by
15 category" option is further selected for restacking the files of FIGURE 14. As shown in FIGURE 15, the selection box SB is around the "stack by category" option 782. Since some of the items are already stacked in the stacks 791 and 792, the selection of the "stack by category" option 782 will restack the items, as will be described in more detail below with reference to FIGURE 16.

20 FIGURE 16 is a diagram illustrative of a screen display in which the files of FIGURE 14 are restacked by category. As shown in FIGURE 16, the stacks 793 and 794 correspond to the "XYZ Corp." and "marketing reports" categories, respectively. The items 751 and 752, corresponding to documents 1 and 2, were not designated for any additional categories, and thus did not fall into any of the other category stacks.

25 FIGURE 17 is a diagram illustrative of a screen display in which a quick link for physical folders is selected. The selection box SB is shown to be around the "all folders" quick link 616. As will be described in more detail below with respect to FIGURE 18, the "all folders" quick link 616 provides for switching to a view of physical folders.

FIGURE 18 is a diagram illustrative of a screen display showing physical folders.
30 The physical folders that are shown contain the files of the virtual folder stacks of FIGURE 17. In other words, the items contained within the stacks 651-655 of FIGURE 17 are also contained in certain physical folders in the system. These are shown in FIGURE 18 as a "My Documents" folder 851 that is located on the present computer, a

"Desktop" folder 852 that is located on the present computer, a "Foo" folder 853 that is located on the hard drive C:, a "My Files" folder 854 that is located on a server, an "External Drive" folder 855 that is located on an external drive, a "My Documents" folder 856 that is located on another computer, and a "Desktop" folder 857 that is located on another computer.

As shown in FIGURE 18, a user is able to switch from the virtual files representation of FIGURE 17 to the physical file representation of FIGURE 18. This allows a user to toggle between virtual file representations and physical file representations, depending on which is desired for a current task. The different locations of the physical folders 851-857 also illustrate that the scope of the virtual file system may be relatively broad, as will be described in more detail below.

FIGURE 19 is a flow diagram illustrative of a routine 880 by which a user can directly manipulate virtual folders. As will be described in more detail below, the mechanisms that are provided for manipulating the virtual folders are similar to those that are currently used for manipulating regular folders (e.g., clicking and dragging, copying, pasting, etc.). As shown in FIGURE 19, at a block 882, the system provides defined actions that the user can perform for direct manipulation of the virtual folders that are represented as display objects. At a block 884, the user performs a defined action. As noted above, one example of this might be a user clicking and dragging a virtual folder to copy its contents to another virtual folder. At a block 886, the virtual folder and/or contents are manipulated as directed by the action performed by the user.

FIGURE 20 is a diagram illustrative of a screen display in which a new West Coast stack 656 has been added to the stacks of FIGURE 10. The West Coast stack 656 was formed by a user creating a new category of "West Coast." Upon its initial creation, the new West Coast stack 656 would be empty and have zero items. In the embodiment of FIGURE 20, two items have been added to the West Coast stack 656. One method for adding items to a stack is to select a particular item, and either modify or add additional categories to the category metadata for the item, such as adding the category "West Coast" to two items as was done in the embodiment of FIGURE 20. This process illustrates that the category data is a metadata property for an item that is a type of ad-hoc property. In other words, a property of this type does not have any implicit meaning, and can be assigned an arbitrary value by the user. For example, the category "property" can have any value whereas the "author" property should be the name of a person. As will be

described in more detail below with reference to FIGURE 21, items may also be clicked and dragged to be copied from other stacks to the West Coast stack 656 (in which case the categories of the items are automatically updated to include "West Coast"). In this regard, FIGURE 20 shows that the selection box SB is around the ABC Corp. stack 651,
5 in preparation for its contents being copied.

FIGURE 21 is a diagram illustrative of a screen display in which direct manipulation is used for copying the files from the ABC Corp. stack 651 to the West Coast stack 656. In other words, as shown in FIGURE 20, the user selected the ABC Corp. stack 651, and then as shown in FIGURE 21 the user has clicked and dragged the
10 stack to be copied to the West Coast stack 656. Thus, the West Coast stack 656 which had two items in FIGURE 20, is now shown to include a total of ten items, including the additional eight items from the ABC Corp. stack 651. When the items from the ABC Corp. stack 651 were copied to the West Coast stack 656, this was accomplished by modifying the category descriptions of the eight items to also include the "West Coast"
15 category in addition to including the original "ABC Corp." category. This illustrates one type of direct manipulation that may be performed.

Another example of direct manipulation is right clicking an item and selecting delete. In one embodiment, when a deleting function is selected by a user, the user is queried whether the item should be deleted all together, or simply removed from the
20 present virtual folder. If the item is just to be removed from a present virtual folder category stack as noted above, this can be accomplished by removing the desired category from the metadata for the item. In other words, if one of the items that had been copied from the ABC Corp. stack 651 to the West Coast stack 656 was then to be removed from the West Coast stack 656, this could be accomplished by modifying the
25 category data for the particular file to no longer include the "West Coast" category.

FIGURE 22 is a flow diagram illustrative of a routine 900 for the system dynamically generating new filter terms. Filter terms are utilized for manipulating the virtual folders. The filtering terms are essentially utilized as a set of tools for narrowing down a set of items. In one embodiment, filters consist of metadata categories and their
30 values (presented to the user in the user interface as clickable links or drop-down menus). The user clicks on a filter term in order to filter down the current results set of items on the display.

FIGURE 22 illustrates how filters may be dynamically generated. As shown in FIGURE 22, at a block 902, the properties (from the metadata) of the items in a collection on the present display are reviewed. In a block 904, proposed filter terms are dynamically generated based on common properties of the items. At a block 906, the proposed filter terms are presented to the user for possible selection for filtering items. As an example of this process, the system may review the properties of a set of items, and if the items generally have "Authors" as a property, the filter can provide a list of the authors to filter by. Then, by clicking on a particular Author, the items that don't have that Author are removed from the set on the display. This filtering process provides the user with a mechanism for narrowing the set of items on the display.

FIGURE 23 is a flow diagram illustrative of a routine 920 for the system filtering items based on the selection of a filter term. At a block 922, the user either enters a new filter term or else selects one of the filter terms that have been presented by the system. As noted above, the filter terms may be dynamically generated by the system, or they may be preset. At a block 924, the items from the collection on the display are evaluated with regard to whether their selected properties match the filter term. For example, if the filter term is for items that were authored by "Bob," then the items are evaluated in accordance with whether their author property includes "Bob". At block 926, the items for which the selected properties do not match the filter term are removed from the collection on the display.

FIGURES 24-29 generally illustrate how the filtering process appears on the screen display. As will be described below with reference to FIGURES 24-29, in one embodiment, the filtering may generally operate according to the following process. After the user clicks on a filter value, the items outside the filter range are animated off the screen. The animation is generally designed to make it obvious that items are being removed and that no new items are being added. The back button 643 may be selected by a user so as to undo the filter operations. In one embodiment, a navigation stack is created which contains the sequential filter actions, which is utilized to undo each of the filter actions when the back button 643 is selected. Each time a filter value is selected, the information areas 640 and 641 are updated to indicate the current filter value. In one embodiment, after a filter value is selected, a user is provided an option for saving a new quick link to the current filter navigation, as will be described in more detail below with

respect to FIGURE 30. As filter values are selected, the filter controls may be updated to be appropriate for the items remaining in the view.

FIGURE 24 is a diagram illustrative of a screen display in which the stacks of FIGURE 10 have been filtered by the term "AB". As shown, in the filter area 621, the term "AB" has been typed by a user. The information lines 640 and 641 indicate that the items in the display are now those that have been filtered by the term "AB". As shown, the ABC Corp. stack 651 still contains eight items, while the Backups stack 652 now contains three items, and the XYZ Corp. stack 654 also contains three items. The information line 644 thus indicates that there are a total of 14 items, taking up a total of 35 MB of memory.

FIGURE 25 is a diagram illustrative of a screen display in which the stacks of FIGURE 10 have been filtered by the term "ABC". With regard to the filter term "AB" of FIGURE 24, the user has simply typed the additional letter "C" to make the total filter term "ABC". As shown in FIGURE 25, the information lines 640 and 641 now indicate that the items on the display are those that contain the term "ABC". The ABC Corp. stack 651 is still shown to contain eight items, while the Backups stack 652 now contains only two items. The XYZ Corp. stack 654 has disappeared because none of its contents matched the "ABC" filter. The information line 644 now indicates that there are a total of 10 items in the stacks on the display, which take up a total of 25 MB of memory. FIGURES 24 and 25 thus provide examples of how a user may enter new filter terms, and how those filter terms are then used to filter the items that are shown on the display.

The back button 643 may be utilized by a user to back through the filtering process. As described above with respect to FIGURE 10, the back button 643 allows a user to back up through a navigation. With regard to the examples of FIGURES 24 and 25, after filtering by the term "ABC" in FIGURE 25, a user could select the back button 643 so as to back up one step of the filtering process, which would return to the state of FIGURE 24. Alternatively, in another embodiment, the back button 643 may clear out the entire filter term, and may thus return to the state before that filtering occurred. In this case, by pressing the back button 643 in FIGURE 25, a user would return to the state of FIGURE 10.

In one embodiment, in addition to the back button, an additional means is provided for a user to back up in or otherwise modify the filtering navigation. This

additional means involves allowing the user to directly access and modify the information line 641, which correspondingly changes the filter navigation. In other words, by directly accessing and modifying the information line 641, the user can remove one or more of the applied filters, or modify the values for any of the applied filters. This feature is described in greater detail in U.S. Patent Application No. 10/420,040, filed April 17, 2003, which is commonly assigned and hereby incorporated by reference in its entirety.

A timer may also be utilized in conjunction with a user typing in filter terms such as those shown in FIGURE 24 and 25. The timer is used to monitor for a pause in the typing by the user. After a selected interval of no typing, the filter is applied. For example, in the state of FIGURE 24, a user has typed the filter term "AB", with no significant time lag between the "A" and the "B." After typing the term "AB", the user pauses, thus producing the state shown in FIGURE 24, where the filter term "AB" is applied. Sometime later, the user adds the letter "C" to complete the filter term "ABC", and then pauses again, at which point the filter term "ABC" is applied as illustrated in FIGURE 25.

In one embodiment, after a user has typed a filter term in the filter area 621, and then chooses another filter or navigation, the navigation state is updated, and the filter term in the filter area 621 is made to be empty again. In addition, as will be described in more detail below with reference to FIGURES 26-29, other filter controls may be updated based on the selection of certain filter terms.

FIGURE 26 is a diagram illustrative of a screen display in which the system provided filter term "year 2002" is selected. As noted above, under the by date indicator 622, the year selections 623 include the years 2000, 2001, or 2002. The selection box SB is shown to be around the year 2002, indicating that the user is selecting that as the desired filter term.

FIGURE 27 is a diagram illustrative of a screen display in which the filter term "2002" has been applied. Also shown is the further selection of the "pick a month" selector 623A. As shown in FIGURE 27, after applying the filter term "2002", the number of items in the stacks have been reduced. More specifically, the ABC Corp. stack 651 now contains six items, the Backups stack 652 now contains eight items, the Business Plans stack 653 now contains three items, and the XYZ Corp. stack 654 now contains five items. The information line 644 now indicates a total of 22 items, taking up

a total of 50 MB of memory. The information lines 640 and 641 now indicate that the items shown on the display are those that have been filtered to contain the filter term "2002".

FIGURE 28 is a diagram illustrative of a screen display in which a list is presented for selecting a month for filtering. A box 950 is provided which includes the list of the months. The box 950 has been provided on the display due to the user selecting the "pick a month" selector 623A. The selection box SB is shown to be around the month of January.

FIGURE 29 is a diagram illustrative of a screen display wherein the stacks of FIGURE 28 have been further filtered by the month of January, and further showing a filter term of "day". As shown in FIGURE 29, the information lines 640 and 641 now indicate that the items on the display are those that have been filtered by the term "January". The Backups stack 652 is now shown to contain two items, while the Business Plans stack 653 is also shown to contain two items. The information line 644 indicates that there are a total of four items on the display, which take up a total of 10 MB of memory. A "pick by day" selector 623B is provided, should the user wish to further filter the results to a specific day.

As described above with respect to FIGURES 24-29, filter terms may be presented by the system, or typed by a user. Once a filter term is selected, the remaining filter terms that are presented may be updated (e.g., after the year "2002" is selected in FIGURE 26, in FIGURE 27 the options for selecting a year are no longer presented and instead a "pick a month" option is provided). As noted above, the back button 643 may be selected by a user to back through the filtering process. For example, after the month of "January" has been selected in FIGURE 29, the user may select the back button 643 to back up the filtering process to the year "2002", as illustrated in FIGURE 27. The filter menu may also include a "stack by" function, which would work similarly to the stack by function described above with respect to FIGURES 15 and 16. For example, a "file type" filter could have choices for "Excel", "PowerPoint", "Word", and also "Stack by file type". Choosing the "stack by" function changes the view to show stacks for the various file types.

In general, the filters may be configured to apply to different properties of the files or items. In one embodiment, the filters may be classified according to different types, such as: alphabet index; discrete values; dates; and numerical ranges. Example

properties for the alphabet index may include file name, author, artist, contact friendly name, owner, document author, document title, document subject, and description. Example properties for the discrete values may include location, file type (application name), genre, track, decade (for music), rating (for music), bit rate, protected, document category, document page count, document comments, camera model, dimensions, product name, product version, image X, image Y, and document created time. Example properties for the dates may include last accessed, last modified, created on, taken on (for pictures). An example property for the numerical range may be file size.

It will be appreciated that the filters described above with respect to FIGURES 24-29 allow users to reduce a list of items to find a particular item that is of interest. As a specific example, according to the processes described above, a user could narrow a current list of documents to only show Microsoft Word files, authored by a particular person and edited in the last week. This functionality allows a user to find a particular item in a list of many, and helps the user avoid having to manually scan each item in the list.

FIGURE 30 is a flow diagram illustrative of a routine 940 for creating a new quick link. As will be described in more detail below, quick links are predefined links that can be clicked on by a user to create user selected views of the sets of items. In one embodiment, a quick link may be thought of as a type of pivot. Quick links provide a mechanism for retrieving a virtual folder. Clicking a quick link can take a user to a desired folder (in the same way that clicking a "favorites" may take a user to a Web site. The quick links can be predefined by the system, or can be set by a user. For example, clicking on "all authors" could return a view stacked by authors. Clicking on "all documents" may return a flat view for all of the documents for all of the storage areas. Users can also create their own quick links.

As shown in FIGURE 30, at a block 942, a user makes a selection on the display to indicate that a new quick link should be formed from the present filter term or navigation. At a block 944, the user provides a new name for the new quick link. At a block 946, the new quick link is saved and the new quick link name is provided in the quick link section on the display.

FIGURE 31 is a diagram illustrative of a screen display for creating a new quick link called "January Work" based on the filtering of FIGURE 29. As described above, in FIGURE 29, the stacks have been filtered by the month of January. In FIGURE 31, the

user has indicated that the filtering of FIGURE 29 should be saved as a new quick link, and has named the new quick link "January work". Thus, the new January work quick link 612 is shown in the quick links section of the display. With regard to forming new quick links, the user is generally provided with an option such as "save this collection as a quick link".

FIGURE 32 is a diagram illustrative of a screen display in which a quick link of "All Authors" is selected. As shown in FIGURE 32, the selection box SB is shown around the All Authors selection 611. Other examples of collections that might be accessible by quick links include "all authors", "recent documents", "all documents I've shared", "all documents I've authored", "all documents not authored by me", "desktop", and "all types".

FIGURE 33 is a diagram illustrative of a screen display in which a list of all of the authors of the items of FIGURE 32 is presented. As shown in FIGURE 33, an information line 950 is provided, which indicates columns for showing the name of an item, the author, the modified date, the type, the size, and the location of an item. A list of Authors 951-954 are shown, corresponding to Authors 1-4, respectively.

FIGURE 34 is a diagram illustrative of a screen display in which "Author 1" has been selected from the list of FIGURE 33. The Author 1's documents include documents 951A and 951B, corresponding to documents 1 and 2, respectively. The document 951A is shown to have been authored by Author 1, was modified on 11 July, 2001, is a Microsoft Excel file, takes up 282 Kb of memory, and was obtained from the location \\server1\folder2. The document 951B is shown to have been authored by Author 1, was modified on 22 December, 2002, is a Microsoft Word file, takes up 206 kilobytes of memory, and is physically stored in the location My Documents\folder1. The locations of the documents 951A and 951B also illustrate that the virtual folders of the present invention may contain items from different physical locations, as will be described in more detail below.

FIGURE 35 is a flow diagram illustrative of a routine 960 for creating a new library. One example of a library is the documents library described above with reference to FIGURE 10. In general, libraries consist of large groups of usable types of files that can be associated together. For example, photos may be one library, music may be another, and documents may be another. Libraries may provide tools and activities that are related to the particular types of items. For example, in the photo library, there may

be tools and filters that relate to manipulating photos, such as for creating slide shows or sharing pictures. As shown in FIGURE 35, at a block 962, a new library is created which is to include items with selected characteristics. At a block 964, the selected items are grouped into the library. At a block 966, the tools and/or activities related to the selected characteristics of the items or to other desired functions are provided.

FIGURE 36 is a diagram illustrative of a screen display in which a collection of available libraries are shown. As shown in FIGURE 36, the libraries include a documents library 971, a photos and video library 972, a music library 973, a messages library 974, a contacts library 975, and a TV and movies library 976, as well as an all items library 977. The all items library 977 is shown to include 275 items, which is the total number of items from all of the other libraries combined. The information line 644 indicates a total of 275 items, which take up a total of 700 MB of memory. It should be noted that the documents library 971 is the library that was described above with respect to FIGURE 10.

FIGURE 37 is a flow diagram illustrative of a routine 990 for defining the scope of a virtual folder collection. As will be described in more detail below, a virtual folder system is able to represent items from multiple physical locations (e.g., different hard drives, different computers, different networks locations, etc.) so that to a user, all of the items are readily accessible. For example, a user can be presented with music files from multiple physical locations on a single display, and manipulate the files all at once.

As shown in FIGURE 37, at a block 992, a scope is defined for the physical locations from which items are to be drawn. At a block 994, in response to a query, the items are drawn from the physical locations as defined in the scope. At a block 996, all of the items drawn by the query are presented on a single display.

FIGURE 38 is a block diagram illustrative of the various sources which may form the scope of a virtual folder collection. As shown in FIGURE 38, the system 1000 may include a present computer 1010, an additional computer 1020, external and removable storage 1030, and locations on a network 1040. The overall scope 1001 is described as including all of the physical locations from which a user's items are drawn to create collections. The scope may be set and modified by a user. As noted above, other figures have illustrated that items may come from different physical locations, such as FIGURE 34 showing different documents coming from a server and a My Documents

folder on a present computer, and in FIGURE 18 showing physical folders that are physically stored in multiple locations.

FIGURE 39 is a flow diagram illustrative of a routine 1080 for including non-file items in a virtual folder collection. Non-file items are contrasted with file items that are typically located in a physical file storage. Examples of non-file items would be things like e-mails, or contacts. As shown in FIGURE 39, at a block 1082 a database is utilized to include non-file items along with file items that may be searched by a query. At a block 1084, in response to a query, both non-file items and file items are drawn to match the query. At a block 1086, both the non-file items and the file items that matched the query are presented on the display.

FIGURE 40 is a diagram illustrative of a screen display showing various non-file items. As shown in FIGURE 40, the items have been filtered to those that include "John". The items are shown to include a contact item 1101, an e-mail item 1102, and document items 1103 and 1104. The contact item 1101 and e-mail item 1102 are non-file items. The present system allows such non-file items to be included with regular file items, such that they can be organized and manipulated as desired by a user. As was described above with respect to FIGURE 2, such non-file items may be contained entirely within the relational database 230, which otherwise includes information about the properties of files.

FIGURES 41-50 are diagrams related to a virtual address bar that corresponds to the information line 641 of FIGURE 10 and which is formed in accordance with the present invention. As will be described in more detail below, the virtual address bar comprises a plurality of segments, and each segment corresponds to a filter for selecting content. Collectively, the corresponding filters of each segment represent a virtual address for selecting content.

FIGURE 41 is a block diagram of an exemplary networked computing environment 1200 suitable for operating the virtual address bar, or any of the other aspects of the present invention described herein. The exemplary networked computing environment 1200 includes a computing device 1202, such as the personal computer described in regard to FIGURE 1, for interacting with a user, and upon which the user may view files stored either locally or remotely to the computing device. While the following discussion describes the present invention in relation to a personal computer, it should be understood that the computing device 1202 includes many types of physical

devices including, but not limited to mini- and mainframe computers, PDAs, tablet computers, and other devices capable of interacting with a user and displaying files and content stored on the computing device and elsewhere.

The exemplary networked computing environment 1200 may also include one or more remote servers, such as server 1204, that stores files accessible to the computing device 1202, and connected to the computing device via a communications network, such as the Internet 1206, as shown in FIGURE 41. In addition, the computing device 1202 may also be connected to other information sources storing files or other content, such as a remote database 1208. Those skilled in the art will recognize that files and information stored on both the remote server 1204 and the remote database 1208, as well as on local storage devices such as hard disk drive (FIGURE 1), may be accessible to, and displayable on, the computing device 1202 as part of an integrated file system on the computing device. Additionally, while a particular configuration of a remote server 1204 and remote database 1208 is presented in FIGURE 41, those skilled in the art will readily recognize that this particular configuration is for illustrative purposes only, and should not be construed as limiting upon the present invention.

FIGURE 42 illustrates an exemplary file viewer 1300 having a conventional address bar 1302 associated with displaying files in a computer file system, as found in the prior art. For purposes of the present discussion, a file viewer is a view or window on a display device, such as display device (FIGURE 1), for displaying files or other content to a user. A file viewer may be a window corresponding to an executable program specifically for displaying files to a user. Alternatively, a file viewer may be a view within an open or close dialog box on an executable program that must save or retrieve data from a storage device connected locally or remotely to the computer system. It should be noted that the above examples of a file viewer are illustrative, and should not be construed as limiting upon the present invention.

An address in the conventional address bar 1302 corresponds to a specific location in a file system. As previously described, in order to edit the address displayed in the conventional address bar 1302, a user must modify the address according to specific knowledge of the file system. Alternatively, a user may select an entry in a tree view 1304 to navigate to an alternative location. Those skilled in the art will recognize that other controls external to the address bar 1302 may also be available that are not shown in the exemplary file view 1300. While the address displayed in the conventional

address bar 1302 corresponds to a specific location in a file system, related files distributed among multiple folders in the file system cannot be displayed in conjunction with the conventional address bar 1302.

FIGURE 43 illustrates an exemplary file viewer 1400 having a virtual address bar 1402 associated with displaying files in a computer file system. The virtual address bar 1402 corresponds to the information line 641 of FIGURE 10. The virtual address bar 1402, having a virtual address 1404, is configured to display similar information to that displayed by the conventional address 1304 of the prior art file viewer 1300 of FIGURE 42. A virtual address, also referred to as a virtual path, references files stored in a computer file system according to selection criteria.

Similar to a conventional address, such as address 1304 of FIGURE 42, the virtual address's selection criteria may reference files stored in a specific location in the file system hierarchy. However, in contrast to a conventional address, the virtual address's selection criteria may also reference files irrespective of their specific file system location. Thus, a virtual address may reference files stored in multiple locations in a computer file system. As shown in FIGURE 43, the file viewer 1400, according to the virtual address 1404 in the virtual address bar 1402, is able to display additional files, such as files 1406 and 1408, not found in the file viewer 1300 of FIGURE 42. Additionally, the virtual address bar 1402 may also be utilized to display content other than files in a computer file system. For example, the virtual address bar 1402 may be used to reference content including system devices, system services, or Internet locations.

FIGURE 44A illustrates manipulating a segment of the virtual address 1404 in the virtual address bar 1402 in order to navigate in a computer file system. Each virtual address bar, such as virtual address bar 1402, is comprised of one or more interactive segments, such as segments 1502, 1504, 1506, and 1508. Each segment in a virtual address bar corresponds to a predetermined filter, or selection criteria, on all of the available content or files accessible to a computer file system. Collectively, the filters of all of the segments in a virtual address bar 1402 represent the virtual address bar's virtual address.

The first segment in a virtual address bar, such as segment 1502, is referred to as a root segment, or root filter. The root segment represents the broadest category of content available for selection by the virtual address bar 1402. For example, segment 1502 "Files" would likely represent a filter that references all files accessible the computer file

system. Alternatively, a root segment may represent a filter that references all system services available to the user on the computer system, or a filter that references all hardware devices installed in the computer system. Those skilled in the art will recognize that numerous other alternative root filters may be utilized by the present invention.

5 Thus, the above described examples are given for illustrative purposes, and should not be construed as limiting upon the present invention. Additionally, the labels displayed for each segment, such as "Files" on the root segment 1502, are illustrative and should not be construed as limiting upon the present invention. According to one embodiment, a label displayed on a segment is user configurable.

10 Each additional segment in a virtual address bar 1402, such as segments 1504, 1506, and 1508, represent additional filters to be applied when selecting and displaying files or content in a file viewer 1400. For example, root segment 1502 "Files" references all files available to the computer system. Segment 1504 "Document Library" filters the files selected by the root segment 1502, by selecting those files that

15 were generated as documents by the user, such as through a word processor, spreadsheet, or some other document generating application. Segment 1506 "Word Documents" filters the files selected by segment 1504 according to those documents that were generated using a word processor, such as Microsoft Corporation's Word application. Finally, segment 1508 "Author A" filters the word processing documents selected by

20 segment 1506 according to whether they were authored by "Author A." Thus, content selected according to the virtual address represented in the virtual address bar 1402 must satisfy the filters corresponding to all of the segments in the virtual address bar.

Segments in the virtual address bar 1402 are generally ordered from those filters that are most inclusive, to those filters that are least inclusive. For example, as previously

25 discussed, segment 1502 "Files" is the broadest and most inclusive. Segments 1506 "Word Documents" and segment 1508 "Author A" are less inclusive. The virtual address bar 1402 illustrates the ordering of segments from left to right, and, for purposes of the present discussion, segments 1504, 1506, and 1508 are subsequent to the root segment 1502. However, it should be understood that other orientations are possible,

30 such as a top-down arrangement, without departing from the scope of the invention. Thus, the orientation from left to right should be viewed as illustrative, and not construed as limiting on the present invention.

As previously mentioned, segments in a virtual address bar 1402, such as segments 1502, 1504, 1506, and 1508, do not necessarily correspond to specific locations in a computer file system, such as folders, drives, and directories. Thus, segment 1504 "Document Library" may reference files or content distributed on multiple servers, drives, or folders/directories. However, certain segments in a virtual address bar 1402 may reference specific locations with a computer file system hierarchy. A further discussion of virtual address segments referencing specific file system locations is given below in regard to FIGURES 48A and 48B.

In contrast to a conventional address bar, each segment in a virtual address bar 1402 represents an actionable, interactive user interface element. For example, a segment in a virtual address bar 1402 is responsive to user selection, monitors whether a cursor is located over the segment for a specific period of time, and may be removed from the virtual address bar by a dragging user interaction. Hence, as shown in FIGURE 44A, a user may place a cursor 1510 over a segment in the virtual address bar 1402, such as segment 1504 "Document Library," to select, or click, on that segment in order to navigate to that level, i.e., truncate the virtual address at that segment, as described in regard to FIGURE 44B.

FIGURE 44B illustrates the results of selecting a segment 1504 in the virtual address bar 1402. By clicking on the segment 1504 in the virtual address bar 1402, the user is indicating a desire to navigate to that level in the virtual address. In effect, the user is trimming off those filters subsequent to the selected segment. For example, by clicking on segment 1504 "Document Library" (FIGURE 44A), the resulting virtual address 1404 no longer contains segments 1506 "Word Documents" and 1508 "Author A" (FIGURE 44A). Additionally, because the user has navigated to a less restrictive set of filters, the resulting virtual address 1404 in the virtual address bar 1402 is more inclusive. This is indicated by the addition of documents in the file viewer 1400 of FIGURE 44B not previously found in the file viewer 1400 of FIGURE 44A, including document 1512, document 1514, and document 1516, and by the presence of a scroll button 1518 indicating that additional files may be viewed that cannot be displayed in the file viewer 1400 (FIGURE 44B) due to space limitations.

In addition to selecting segments in a virtual address bar to navigate to a less restrictive segment, a user may also wish to navigate to, or select, peer filters of current segments in a virtual address. A peer filter is an alternative filter that may be selected and

applied to a given segment in the virtual address bar. For example, with reference to FIGURE 44A, peer filters for segment 1506 "Word Documents" may include filters such as "Excel Documents," "Journals," and the like. Other types of filters, including specific file system locations, hardware devices, or computer services, may also be applied to a given segment in the virtual address bar. Peer filters may or may not be logically related to a given segment's current filter. Each segment in a virtual address bar may have peer filters. Selecting a peer filter of a segment in a virtual address bar is sometimes referred to as navigating laterally. Selecting peer filters of segments in a virtual address bar is described below in regard to FIGURES 45A-45D, and also in regard to FIGURE 49.

FIGURES 45A-45D are pictorial diagrams illustrating selecting a peer filter associated with a segment of virtual address in a virtual address bar 1600. As shown in FIGURE 45A, virtual address bar 1600 has a virtual address comprising multiple segments, segments 1602-1608. In order to select a peer filter for a given interactive segment in a virtual address bar 1600, a user must make an alternative selection, or alternative manipulation, of that interactive segment. One way to make an alternative selection is to right click on a given segment. Right clicking is known in the art and refers to using a secondary button on a mouse, or other input device, where the secondary button is typically on the right-hand side of the mouse. Alternatively, because an interactive segment can monitor when a cursor is located over it, an alternative selection may be made by locating the cursor over an interactive segment and leaving the cursor in place for predetermined amount of time, sometimes referred to as hovering. However, while the present discussion describes alternatives for causing peer filters to be displayed, they are for illustration, and should not be construed as limiting upon the present invention. Those skilled in the art will recognize that there are numerous alternatives for generating an alternative selection.

To illustrate alternatively selecting a segment, with reference to FIGURE 45A, a user first places the cursor 1610 over segment 1604 "Document Library" for a predetermined amount of time, i.e., hovers over the segment, to select that segment. FIGURE 45B demonstrates the results of alternatively selecting segment 1604 "Document Library" in the virtual address bar 1600. As shown in FIGURE 45B, after alternatively selecting segment 1604 "Document Library," a peer filter view 1612 is displayed including peer filters corresponding to the selected segment. It should be

understood that the peer filters presented in the peer filter view 1612 are for illustrative purposes only, and should not be construed as limiting upon the present invention.

In order to select an alternative peer filter, as shown in FIGURE 45C, the user positions the cursor 1610 over one of the filters presented in the peer filter view 1612, such as peer filter 1614, and selects the peer filter. As shown in FIGURE 45D, after selecting the alternative peer filter 1614, the previously selected segment 1604 (FIGURE 45A) is replaced with a new segment 1616 representing the selected alternative peer filter 1614. Additionally, those segments that followed the alternatively selected segment 1604 in the virtual address bar 1600 of FIGURE 45A, specifically segments 1606 "Journals" and 1608 "All Documents in 2002", are removed from the virtual address bar 1600 in FIGURE 45D. Although not shown, it follows that any files or content previously selected according to segments 1604 "Document Library", 1606 "Journals", and 1608 "All Documents In 2002" would no longer be displayed in a corresponding file viewer, and only those files or content selected according to segments 1602 "Files" and 1616 "Picture Library" would be displayed.

Segments may be added to a virtual address in a virtual address bar through various user interactions at the end of the existing segments. To add a filter to a virtual address in a virtual address bar, a user may manipulate an actionable control associated with a particular filter found on a window, or file viewer with the virtual address bar. For example, with reference to the file viewer 1400 of FIGURE 43, a user may click on the actionable control 1412 "2003" to add a corresponding filter to the virtual address 1404 in the virtual address bar 1402. Alternatively (not shown), a user may manually enter in a known filter at the end of the virtual address by typing the filter's name. Numerous other ways of adding a filter to a virtual address exist, all of which are contemplated as falling within the scope of the present invention. Thus, it should be understood that the above examples are for illustration purposes, and should not be construed as limiting upon the present invention.

When a filter is added to a virtual address in a virtual address bar, a process is undertaken to ensure that the newly added filter does not conflict with any filters currently existing as part of the virtual address. If the newly added filter conflicts with an existing filter, the existing filter is removed. A newly added filter conflicts with an existing filter in a virtual address if the newly added filter varies from the breadth of the existing filter, being either more or less broad than the existing filter. Additionally, a

newly added filter conflicts with an existing filter if the newly added filter is mutually exclusive to the existing filter. However, a newly added filter that is equivalent to an existing filter is not added because it has no effect. It should be understood that the above description of conflicts is given for illustration purposes, and should not be construed as limiting upon the present invention. Those skilled in the art will recognize that other conflicts between filters may exist that are contemplated as falling within the scope of the present invention.

FIGURES 46A-46D are pictorial diagrams illustrating adding filters to a virtual address 1702 in a virtual address bar 1700, and removing conflicting existing filters.

FIGURE 46A illustrates an exemplary virtual address 1702 displayed in a virtual address bar 1700. As shown in FIGURE 46B, a new filter, represented by segment 1706 "2002", is added to the virtual address 1702. As previously described, new filters are added to the end of the virtual address, as indicated by placing segment 1706 "2002" at the end of the segments in the virtual address bar 1700 of FIGURE 46B. Thereafter, the process undertaken for adding segment 1706 "2002" determines that the added filter does not conflict with any current filters in the virtual address 1702. Thus, no existing filters are removed from the virtual address 1702.

As shown in FIGURE 46C, another filter is added to the virtual address 1702, represented by segment 1708 "Author A." The process undertaken for adding this new filter determines that the new filter, "Author A," would conflict with the filter represented by segment 1704 "Author A-F" because the new filter, "Author A," is narrower than the existing filter. Accordingly, segment 1704 "Author A-F" is removed from the virtual address bar 1700, and segment 1708 "Author A" is added to the end of the segments in the virtual address bar.

FIGURE 46D illustrates the results of adding segment 1710 "2003" to the virtual address bar 1700 of FIGURE 46C. Filters in a virtual address 1702 are restrictive, not cumulative. Each filter further restricts the selected content. Thus, mutually exclusive filters would prevent the virtual address 1702 from selecting any files or content, and therefore, create a conflict. As illustrated in FIGURE 46D, segment 1706 "2002" (FIGURE 46C) is removed from the virtual address bar 1700 because of a conflict as it is mutually exclusive with the newly added segment 1710 "2003."

When a virtual address bar, such as virtual address bar 1800 (FIGURE 47A), cannot completely display the virtual address due to size limitations of the virtual address

bar, a portion of the virtual address is displayed according to the size of the virtual address bar. However, the undisplayed portions of the virtual address may still be accessed by the user. More specifically, the virtual address bar displays actionable visual indicators to scroll the virtual path within the virtual address bar. FIGURES 47A and 47B illustrate an exemplary virtual address bar 1800 displaying a virtual address where the virtual address exceeds the virtual address bar's display capacity. As shown in FIGURES 47A and 47B, scroll icons 1802 and 1804 indicate the direction the virtual address bar 1800 may scroll in order to display the previously undisplayed portions of the virtual address. However, while the illustrative diagrams demonstrate the use of scroll icons, it is for illustrative purposes only, and should not be construed as limiting on the present invention. Those skilled in the art will recognize that there are numerous other ways of scrolling the virtual address in a virtual address bar, all of which are contemplated as falling within the scope of the present invention.

FIGURE 48A is a block diagram illustrating a virtual address bar 1900 having segments referencing both virtual and actual locations in a file system. As previously discussed, a virtual address in a virtual address bar 1900 may contain segments referencing specific locations within a computer file system hierarchy, and also contain segments referencing virtual, or logical, locations within a computer file system. Files or content referenced by a virtual segment may be distributed among many physical locations. A virtual address bar 1900 may contain segments referencing physical locations and segments referencing virtual locations. For example, virtual address bar 1900 includes segment 1902 "Local Disk (C:)" referring to files or content contained in a specific area in the computer file system, in particular drive "C." Alternatively, segment 1904 "Case Files" of itself refers to files or content stored in multiple folders in the computer file system hierarchy associated with case files. However, in combination with segment 1902 "Local Disk (C:)", segment 1904 "Case Files" references only those case files found on local drive "C." Additionally, segment 1906 "Contains 'Fax'" further filters the files on local disk C: and associated with the case files according to whether they contain the word "Fax."

As shown in FIGURE 48B, a virtual address bar 1900 may be configured to function as a conventional address bar. For example, with reference to FIGURE 48A, by placing a cursor 1908 in the empty space of the virtual address bar 1900 and clicking there, the virtual address bar 1900 switches from displaying segments representing a

virtual address, to functioning as a conventional address bar displaying a conventional address 1910, as shown in FIGURE 48B. The conventional address 1910 in the virtual address bar 1900 of FIGURE 48B approximates the virtual address displayed in the virtual address bar 1900 of FIGURE 48A. However, those filters in the virtual address bar 1900 of FIGURE 48A that do not correspond to physical locations in a computer file system cannot be displayed and are removed from the conventional address 1910. Specifically, segment 1904 "Case Files" and segment 1906 "Contains 'Fax'" are not part of the conventional address 1910 (FIGURE 48B).

In order to reconfigure a virtual address bar 1900, functioning as a conventional address bar, to function normally as a virtual address bar, the user must so indicate in a manner other than clicking on the empty area of the bar. When configured to function as a conventional address bar, a virtual address bar must permit the user to click in the empty area for address editing purposes. Clicking in the empty area of a conventional address bar places an editing cursor at the end of the address/path for editing purposes. Accordingly, to reconfigure the virtual address to again function in its normal manner as described above, a user must press a predefined key or key sequence, such as the Esc or Tab key, or by place the focus on another area of a window or view by clicking on another area of the window or view. Those skilled in the art will recognize that other user actions may also be utilized to reconfigure the virtual address bar 1900 to again function in its normal mode as described above, all of which are contemplated as falling within the scope of the present invention.

FIGURE 49 is a flow diagram illustrative of a peer filter selection routine 2000 for selecting a peer filter for an identified segment in a virtual address bar. Beginning at block 2002, the routine 2000 detects a peer filter selection activation. Activating the peer filter selection process is described in above in regard to FIGURES 45A-45D. At block 2004, the segment for which the peer filter selection has been requested is identified. At block 2006, the peer filters for the identified segment are determined from a predetermined list of peer filters. At block 2008, the peer filters are displayed to the user. At block 2010, the user's peer filter selection from peer filters displayed is obtained. At block 2012, the virtual address is truncated by removing the identified segment from the virtual address bar, and any additional segments that follow the identified segment. At block 2014, a segment representing the selected peer filter is appended to the remaining segments in the virtual address bar. Thereafter, the routine 2000 terminates.

FIGURE 50 is a flow diagram illustrating an exemplary add filter routine 2100 for adding a filter to a virtual address in a virtual address bar. Beginning at block 2102, the exemplary routine 2100 obtains the filter to be added to the virtual address. For example, as previously discussed in regard to FIGURE 43, filters may be added to the virtual address according to user actions external to the virtual address bar, or alternatively, may be directly added to the virtual address bar by typing in the name of a predefined filter.

At block 2104, a determination is made whether the new filter conflicts with an existing filter already in the virtual address. As previously discussed in regard to FIGURES 46A-46D, a new filter may conflict with an existing filter by substantially narrowing or broadening the scope of the existing filter. Alternatively, a new filter may conflict with an existing filter because a new filter is mutually exclusive to an existing filter. If, at decision block 2104, the new filter conflicts with an existing filter, at block 2106, the existing filter is removed from the virtual address. Alternatively, at 2104, if the new filter does not conflict with an existing filter or, after removing the existing conflicting filter in block 2106, at block 2108, the new filter is added at the end of the virtual address. Thereafter, the exemplary routine 2100 terminates.

FIGURES 51-57 are diagrams related to a system and method in accordance with another aspect of the invention that provides an improved user experience within a shell browser. More specifically, a system and method are provided by which users can more readily identify an item based on the metadata associated with that item.

Turning to FIGURE 51A, a window 2200 represents a screen-size display area for a graphical user interface of a shell browser. The window 2200 contains a preview pane area 2202 and a view area 2204. The preview pane 2202 may include a preview control 2206, a user interface (UI) or edit control 2208, and a task control 2210. Typically, the preview control 2206 will provide the user with an image or other visual display of the item being previewed (e.g., a selected file). The preview control 2206 may also present the user with controls such as iterator buttons which allow the user to shift the focus from one item to the next by clicking a mouse button. Metadata corresponding to one or more items and/or metadata corresponding to the item container may be displayed in a variety of locations within the window 2200. For example, the edit control and metadata may be co-located within edit control area 2208 so that the edit control area not only includes a display of key properties of the previewed item but also presents the

user with the option of making edits to the metadata. The task control 2210 contains tasks relevant to the namespace and/or the selection.

For purposes of the present invention, the terms "metadata" and "user modifiable metadata" exclude the shell item name. The term "shell item name" refers to the property
5 which is used for purposes of sorting and displaying the item within the shell browser. As mentioned above, one unique aspect of the present invention is the ability of a user to edit metadata within a shell browser.

Those skilled in the art will appreciate that the present invention contemplates the presence of optional features within the window 2200. For example, the preview
10 control 2206 and the task control 2210 are not essential features for purposes of the present invention. Moreover, other non-essential features which are not shown in FIGURE 51A, such as a toolbar which includes iterator buttons or a show/hide button so the user can open/close the preview pane, are also within the scope of the present invention. Nevertheless, these and other optional features may assist the user in readily
15 locating a particular item in the shell browser.

The view area 2204 provides a listview of one or more items 2212, such as file system files or folders. The term "listview" refers to an enumeration or list of items within a container. The terms "item" and "shell item" are used interchangeably herein to refer to files, folders and other such containers, and other non-file objects which can be
20 represented in a listview. Examples of non-file objects may include, but would not be limited to, contacts, favorites and email messages. The terms "shell browser" and "file system browser" are used interchangeably herein to refer to a browser which allows a user to navigate through various namespaces including files and other non-file items.

Those skilled in the art will appreciate that the present invention contemplates
25 many possible designs and layouts for the window 2200. For example, the preview pane 2202 is shown above the view area 2204 in FIGURE 51A. However, other layouts, such as placing the preview pane 2202 and the view area 2204 side-by-side, are clearly within the scope of the present invention. The location of the edit control 2208 is also independent of the location of the displayed metadata and independent of the location of
30 any other controls. There are also many possible view types for the items depicted in listview area 2204, such as details, slide show, filmstrip, thumbnail, tiles, icons, etc.

FIGURE 51B is similar to FIGURE 51A, except that the view area 2204 is replaced by a view area 2214 which displays the items 2212 in details mode. As is

typical for shell items displayed in details mode, the items 2212 are aligned in a column at the left-hand side of view area 2214, and one or more column headings 2216 form the top row of a set of columns containing metadata 2218 relating to the corresponding item located in the same row. Importantly, the present invention contemplates the ability of a user to explicitly change a metadata value to another value through instantiation of one or more edit controls 2208 anywhere within the window 2200. For example, an edit control may be provided within the preview pane 2202 and/or within the view area 2214. For example, an edit control which is not initially visible to a user may be provided within the view area 2214. Such a control can be instantiated, for example, when the user hovers over the metadata 2218 and then clicks on it to enter an editing mode.

Referring next to FIGURE 52, a schematic illustration is provided of a welcome pane 2300 in a shell browser. A welcome pane is sometimes referred to as a "null select" pane because it represents a namespace or container as opposed to a selection. If the user has not yet made a selection, a preview pane 2302 displays metadata 2304 and key tasks relating to the folder or shell library. If desired, the tasks may be separated into premiered tasks 2306 and other relevant tasks 2308. The welcome pane 2300 also includes a view area 2310, in which multiple files or other items 2312 may be viewed. The welcome pane metadata 2304 may include information such as properties of the container (e.g., MyPictures), in which case the metadata display may be static. Alternatively, the welcome pane metadata 2304 may include information such as a sampling of metadata from each of the items within the container, in which case the metadata display may change frequently. For example, the metadata display may be limited to properties of one item at a time by cycling from one item to the next every 30 seconds.

FIGURE 53 is a schematic illustration of a selected pane 2400 in a shell browser. As opposed to a welcome pane, a selected pane represents a selection by the user. If the user selects a container or folder, the selected pane need not be identical to the welcome pane for that container or folder. In FIGURE 53, the selected pane 2400 includes a preview pane 2402 which contains a preview control 2404, a metadata display 2406 and a tasks display 2408. Like the welcome pane 2300 (in FIGURE 52), the selected pane 2400 also includes a view area 2410, in which multiple files or other items 2412 may be viewed. In FIGURE 53, however, the user has selected one of the files. Consequently, the preview control 2404 displays a preview image of the selected file, the

metadata display 2406 shows properties of the selected file, and the tasks display 2408 provides a menu of relevant tasks for operating on the selected file.

FIGURE 54 is a schematic representation of the selected pane of FIGURE 53 but which also includes a context menu 2500 to enable a user to modify metadata in a shell browser in accordance with an embodiment of the present invention. The context menu 2500 in FIGURE 54 presents the user with several options for changing the selected metadata. The generic text shown in the menu 2500 is of course merely one example of the type of options which may be presented to a user for editing the displayed metadata. A context menu can be provided in any window, including a welcome pane, to improve the user experience. As those skilled in the art will appreciate, any number and variety of context menus could be supported by the present invention. For purposes of the present invention, one means for enabling user modifications to displayed metadata within a shell browser is to provide a context menu such as editable metadata context menu 2500. A user may summon the context menu, for example, by clicking on the corresponding text or object in the preview pane.

Those skilled in the art will appreciate that the present invention contemplates means other than context menus for enabling user modifications to displayed metadata within a shell browser. Another such means for is for the user to click on the metadata to enter an editing mode. By contrast, a user could enter an editing mode by hovering over the relevant text or object in the preview pane. Numerous alternative means are available and within the scope of the present invention.

FIGURE 55 is a flow diagram illustrating a method 2600 for enabling a user to modify metadata displayed in a welcome pane within a shell browser in accordance with an embodiment of the present invention. The method 2600 includes displaying a welcome pane and metadata associated with the welcome pane at 2602. Then, at 2604, the method provides a control for user modification of the displayed metadata. When the user manipulates the control to modify the displayed metadata at 2606, the method then associates the modified metadata with the welcome pane at 2608 so that the modified metadata will be displayed the next time the welcome pane is displayed.

FIGURE 56 is a flow diagram illustrating a method 2700 for enabling a user to modify metadata displayed in a selected pane within a shell browser in accordance with an embodiment of the present invention. At 2702, the method 2700 first displays a number of items, such as items in a welcome pane or items in a selected container. When

the user selects one or more of the items at 2704, the method displays metadata associated with the selected item(s) at 2706. At 2708, the method provides a control for user modification of the displayed metadata. When the user manipulates the control to modify the displayed metadata at 2710, the method then associates the modified metadata with the selected item(s) at 2712 so that the modified metadata will be displayed the next time the selected item(s) is/are displayed.

In the event a user selects multiple items at 2704, the displayed metadata may include intersecting properties of the selected items, a union of properties, or perhaps a new property relevant to the selected items. Alternatively, the displayed metadata may include a rotating sample of metadata from each of the selected items (e.g., cycling from one selected item's metadata to the next selected item's metadata every 30 seconds). It is possible for the display of metadata which would result from a selection of all of the items to be identical to the display of metadata which would result from a null select.

FIGURE 57 is a block diagram of a data structure 2800 containing user modifiable metadata associated with an item displayed in a shell browser. The data structure 2800 includes a title field 2802 which indicates the name of the item. In the case of non-file items, the title field 2802 may contain the name of whatever property is used to alphabetize that item in a listview. The data structure 2800 includes a user editable properties field 2804 containing one or more properties associated with the displayed item, wherein the user editable properties are displayed in the shell browser with the displayed item. The data structure 2800 may optionally include a read-only properties field 2806 which contains any read-only properties associated with the displayed item and worthy of display in the shell browser. Given the size constraints of the metadata display in the shell browser, the number of properties in fields 2804 and 2806 may be limited. Consequently, the data structure 2800 may optionally include an all properties field 2808, which contains a link or pointer to a location (e.g., a property page) which contains all of the properties or metadata associated with the displayed item. Of course, the all properties field 2808 would not be necessary in the event that fields 2804 and 2806 contain all of the properties associated with the displayed item. The data structure 2800 is stored on one or more computer-readable media, such as in a file system or shell, to provide rich storage views, and thus an improved user experience, within the shell browser.

The present invention enables a number of scenarios which were not possible with conventional shell browsers. As a first example, a student can manage her projects using the preview pane. When she obtains new documents as part of a project she is working on, she can select those documents in her document library and enter the name of the document author and the name of the project into keyword fields using the edit control. Now the new documents will show up in her favorite view: "Documents Grouped by Keyword and Listed by Author." A second example of a new scenario enabled by the present invention involves an employee looking for materials for an upcoming ad campaign. As he browses through his employer's stock collection of photos using the shell browser, he selects a couple of pictures and, from the preview pane, adds a new keyword "Summer 2003 Campaign." Having updated the metadata for a multiple selection, the employee then pivots by keyword and can view all of the "Summer 2003 Campaign" files grouped together. Many other scenarios which take advantage of the present invention would be apparent to those skilled in the art.

FIGURES 58-66 are diagrams related to a system and method for extending the functionality of an object previewer in a shell browser configured to display a plurality of items representing multiple item types. As will be described in more detail below, a shell browser is provided which includes a default previewer and an extensibility mechanism. The default previewer provides a standard level of functionality for multiple item types. The extensibility mechanism enables functionality beyond the standard level provided by the default previewer for one or more of the item types.

FIGURE 58 is a schematic diagram of a prior art graphical user interface for browsing pictures stored in a folder within a shell browser environment which is used for viewing other non-pictorial files and folders. As stated above, the need to readily identify items that are stored in a computing environment such as a PC is dramatically increasing. With respect to digital pictures, users traditionally had to invoke a third party software program in order to view a specific file on the PC. FIGURE 58 illustrates a prior solution, a film strip view, which allows users to more readily view and identify the image associated with a given file within the graphical operating environment. The goal of film strip view was to alleviate the need for other software programs when browsing a folder of pictures by providing a quick iterative process that allows a user to preview a sizeable image of one or more picture files within the folder.

FIGURE 58 relates to a system for browsing pictures stored in a folder, wherein a series of folder pictures is presented as a single row of thumbnails within an environment that is utilized for viewing other non-pictorial files and folders (i.e., a shell browser). It further allows a user to selectively cursor through the thumbnails, as it displays an enlarged preview image of a user selected thumbnail. FIGURE 58 is a diagram of a representative window on a user's screen. As shown, the window 3200 is divided into several areas including a header region, a task option area 3206, a preview control area 3202, a caption or comment area and a filmstrip area 3204. The task option area 3206 contains a list of tasks that can be selected by a user in order to perform a wide variety of operations relating to the management of files and folders, as well as other system choices. Some of these operations are specific to the pictures in the filmstrip area 3204 and the preview control area 3202. The preview control area 3202 is a space in which an enlarged preview image of a user selected picture will be displayed. This space can also contain navigational icons to assist a user in iterating through a series of pictures. Immediately below the preview control area is a caption or comment area that can be utilized to display a variety of textual information. A film strip area 3204, provides a space to display a single row of thumbnail images P1, P2, P3, P4 of the picture files contained within a given folder. In addition, the film strip area 3204 also contains cursors to allow a user to scroll through a folder for the picture files. It should be noted that the filmstrip area 3204 can contain and display thumbnail images in mixed orientation. For instance, as shown in FIGURE 58, P1, P2 and P4 are in landscape while P3 is in portrait.

A user can select any one of the thumbnail images, which will cause a larger preview image of the user thumbnail selection image to be displayed within the preview control area. In addition, user selection of a thumbnail image will also allow the user to select and perform any one of the tasks listed in the task option area 3206, with respect to the selected image. A first control button allows a user to quickly and successively preview an enlarged image of each of the thumbnail images within a given folder, by iterating in one direction. In other words, a user would not have to specifically "click" on each and every successive thumbnail image in order to preview the picture. Instead the user will merely click on the first control button repeatedly to move through the folder. A second control button performs a similar iteration function but only in the opposite direction.

items depicted in view area 3304, such as details, slide show, filmstrip, thumbnail, tiles, icons, etc.

Referring next to FIGURE 60, a schematic illustration is provided of a welcome pane 3400 in a shell browser. A welcome pane is sometimes referred to as a "null select" pane because it represents a namespace or container as opposed to a selection. If the user has not yet made a selection, a preview pane 3402 displays metadata 3404 and key tasks relating to the folder or shell library. If desired, the tasks may be separated into premiered tasks 3406 and other relevant tasks 3408. The welcome pane 3400 also includes a view area 3410, in which multiple files or other items 3412 may be viewed.

10 The welcome pane metadata 3404 may include information such as properties of the container (e.g., MyPictures), in which case the metadata display may be static. Alternatively, the welcome pane metadata 3404 may include information such as a sampling of metadata from each of the items within the container, in which case the metadata display may change frequently. For example, the metadata display may be

15 limited to properties of one item at a time by cycling from one item to the next every 30 seconds.

FIGURE 61 is a schematic illustration of a selected pane 3500 in a shell browser. As opposed to a welcome pane, a selected pane represents a selection by the user. If the user selects a container or folder, the selected pane need not be identical to the welcome pane for that container or folder. In FIGURE 61, the selected pane 3500 includes a preview pane 3502 which contains a preview control 3504, a metadata display 3506 and a tasks display 3508. Like the welcome pane 3400 (in FIGURE 60), the selected pane 3500 also includes a view area 3510, in which multiple files or other items 3512 may be viewed. In FIGURE 61, however, the user has selected one of the files.

20 Consequently, the preview control 3504 displays a preview image of the selected file, the metadata display 3506 shows properties of the selected file, and the tasks display 3508 provides a menu of relevant tasks for operating on the selected file.

FIGURE 62 is a schematic diagram of a selected pane similar to the selected pane of 3500 of FIGURE 61 but with extended controls in accordance with an embodiment of the present invention. The selected pane 3600 includes a preview pane 3602 which contains a preview control 3604 having extended controls 3614, a metadata display 3606 and a tasks display 3608. The selected pane 3600 also includes a view area 3610, in which multiple files or other items 3612 may be viewed. The user has selected one of the

30

Turning to FIGURE 59, a window 3300 represents a screen-size display area for a graphical user interface of a general purpose shell browser. The window 3300 contains a preview pane area 3302 and a view area 3304. The preview pane 3302 may include a preview control 3306, an edit or metadata control 3308, and a task control 3310.

5 Typically, the preview control 3306 will provide the user with an image or other visual display of the item being previewed (e.g., a selected file). The preview control 3306 may also present the user with controls such as iterator buttons which allow the user to shift the focus from one item to the next by clicking a mouse button. The edit control 3308 not only includes a display of key properties of the previewed item, it also presents the user

10 with a control for making edits to the metadata. The task control 3310 contains tasks relevant to the namespace and/or the selection.

Those skilled in the art will appreciate that the present invention contemplates the presence of optional features within the window 3300. For example, the metadata control 3208 and the task control 3210 are not essential features for purposes of the

15 present invention. Moreover, other non-essential features which are not shown in FIGURE 59, such as a toolbar which includes iterator buttons or a show/hide button so the user can open/close the preview pane, are also within the scope of the present invention. Nevertheless, these and other optional features may assist the user in readily locating a particular item in the shell browser.

20 The view area 3304 provides a listview of one or more items 3312, such as file system files or folders. The term "listview" refers to an enumeration or list of items within a container. The terms "item" and "shell item" are used interchangeably herein to refer to files, folders and other such containers, and other non-file objects which can be represented in a listview. Similarly, "shell item" refers to an item in a shell library.

25 Examples of non-file objects may include, but would not be limited to, contacts, favorites and email messages. The terms "shell browser" and "file system browser" are used interchangeably herein to refer to a browser which allows a user to navigate through various namespaces including files and other non-file items.

Those skilled in the art will appreciate that the present invention contemplates

30 many possible designs and layouts for the window 3300. For example, the preview pane 3302 is shown above the view area 3304 in FIGURE 59. However, other layouts, such as placing the preview pane 3302 and the view area 3304 side-by-side, are clearly within the scope of the present invention. There are also many possible views for the

files 3612, so the preview control 3604 displays a preview image of the selected file, the metadata display 3606 shows properties of the selected file, and the tasks display 3608 provides a menu of relevant tasks for operating on the selected file.

5 The extended controls 3614 represent a level of functionality beyond what is typically available from a shell browser. For example, a default preview pane or preview control, such as those shown in FIGURES 58 and 61, may simply display a preview image of a selected item. If the item is a word processing document or slide presentation, the default preview image may be the first page of the document or slide deck. However, by extending the functionality of the preview image to make it more interactive, a user
10 can quite easily manipulate extended controls 3614 to page through the document or slide presentation. This enhanced level of functionality improves the user experience because it allows the user to more comprehensively browse the previewed item without opening it, which is particularly useful for files that are not readily identifiable based on the first page alone.

15 Extended controls 3614 can be made available to the user as part of an alternative previewer in a shell browser. The term "previewer" can refer to a preview control or to the a preview pane which includes a preview control. The present invention contemplates a shell browser which provides the user with a default previewer offering a standard level of functionality for multiple item types and one or more alternative previewers offering a
20 different level of functionality for particular item types to enhance the user experience. Opening up the development of alternative previewers to independent software vendors (ISVs) and other third party developers adds value to the file browsing experience by showing relevant aspects of the file in an easily recognizable way. The present invention contemplates custom previewers for numerous file types and non-file item types
25 including, but not limited to, image files, video files, contacts, games, scanners, video cameras, document files, spreadsheet files, slide presentation files, drawing files and tablet ink files.

The present invention enables a number of scenarios which were not possible with conventional shell browsers, some of which have been described above. Third parties are
30 allowed to describe and demonstrate their file types by providing code that can look inside the file type and provide a meaningful image that a user will understand. For example, Apple could implement a QuickTime™ preview control, which would be displayed when the user selects a QuickTime™ file in the shell browser. This preview

control could provide an alternative or extended level of functionality beyond the default previewer in the shell of an operating system, including functionality such as showing the first five seconds of a QuickTime™ movie and/or offering buttons and controls for the user to launch the QuickTime™ player. An alternative previewer for a music file could provide similar extended functionality. As those skilled in the art will appreciate, the possibilities for extended functionality in an alternative previewer are unlimited.

FIGURE 63 is a schematic representation of a selected pane similar to FIGURE 61 but which also includes a context menu 3714 to enable a user to modify metadata in a shell browser in accordance with an embodiment of the present invention. The selected pane 3700 includes a preview pane 3702 which contains a preview control 3704, a metadata display 3706 and a task control 3708. The selected pane 3700 also includes a view area 3710, in which multiple files or other items 3712 may be viewed. Those skilled in the art will appreciate that, for purposes of the present invention, the metadata control 3706 and the task control 3708 are not essential features. The present invention contemplates the presence of these and/or other optional features which may assist the user in readily locating a particular item in the shell browser or otherwise enhance the user experience.

The context menu 3714 in FIGURE 63 presents the user with several options, including the choice of selecting either the default previewer or an alternative previewer for the selected item. The generic text shown in the menu 3714 is of course merely one example of the type of options which may be presented to a user for selecting a previewer. A context menu can be provided in any window, including a welcome pane, to improve the user experience. As those skilled in the art will appreciate, any number and variety of context menus could be supported by the present invention. For purposes of the present invention, one means for enabling user selection of a previewer within a shell browser is to provide a context menu such as context menu 3714. A user may summon the context menu, for example, by clicking on the corresponding text or object in the preview pane.

Those skilled in the art will appreciate that the present invention contemplates means other than context menus for selecting a previewer for the displayed items from a plurality of available previewers within a shell browser. Another such means is for the user to click on the preview control to enter a selection mode. Similarly, the user may be prompted to select a previewer by right-clicking within the preview pane. By contrast, a

user could enter a selection mode by hovering over relevant text or over a relevant object in the preview pane. Numerous alternative means are available and within the scope of the present invention.

FIGURE 64A is a flow diagram illustrating a method 3800 for enabling a user to
5 select a previewer in a shell browser which supports multiple item types in accordance with an embodiment of the present invention. The method 3800 provides a plurality of previewers in the shell browser at 3802. The plurality of previewers may include a default previewer for multiple item types and one or more alternative previewers for particular item types. These alternative previewers may include installed applications
10 developed by a third party. At 3804, the method 3800 presents the user with a choice of two or more previewers for a particular item type. The prompt to select a previewer may be initiated by the shell browser (e.g., upon displaying a new item type) and/or by the user (e.g., by clicking on an object to display a context menu). Upon receiving an input from the user at 3806 indicating a selection of one of the previewers for the particular
15 item type, the method 3800 then associates the selected previewer with the particular item type at 3808. The selected previewer will remain in use until the user selects a different one. However, if the selected previewer is an installed application, uninstalling the application will also terminate the use of the selected previewer.

FIGURE 64B is a flow diagram illustrating a method 3810 for automatically
20 selecting a previewer in a shell browser which supports multiple item types in accordance with an embodiment of the present invention. The method 3810 provides a plurality of previewers in the shell browser at 3812. The plurality of previewers may include a default previewer for multiple item types and one or more alternative previewers for particular item types. These alternative previewers may include installed applications
25 developed by a third party.

At 3814, the system (as opposed to the user) automatically and transparently selects a default previewer from two or more available previewers for a particular item type. The system may select a previewer in response to an event such as display of a new item type or the presence of an alternative previewer. The system is configured to select
30 a default previewer based on logical rules. Under exceptional circumstances, the system may decide at 3816 to override the rules and select a previewer that would not have been selected under the applicable rules. For example, if the rule is to select a newly available previewer over the current default previewer, an installed application may generally have

the authority to change the default previewer to the previewer now available from the installed application. However, the shell browser, for example, may reserve the right to override the change proposed by the newly installed application. For instance, an override may be appropriate when the newly installed application cannot be authenticated as a proper owner of the item type in question.

In any event, the method 3810 then associates the selected previewer with the particular item type at 3818. The selected previewer will remain in use until a different one is selected. However, if the selected previewer is an installed application, uninstalling the application will also terminate the use of the selected previewer.

Referring next to FIGURE 65, a flow diagram illustrates a method 3900 for enabling the use of third party previewers in a shell browser which supports multiple item types in accordance with an embodiment of the present invention. The method 3900 includes providing a shell browser having a default previewer for the multiple item types at 3902. The method 3900 further includes providing an extensibility mechanism for third party development of an alternative previewer for at least one of the multiple item types at 3904. The alternative previewer may be registered in the shell browser at 3906. In the case of an installed application, registration may occur substantially at the time of installation. For example, if the application is installed by an OEM, the alternative previewer may be registered before the user has acquired the computer. Alternatively, the user may install the application locally or remotely.

There are many possible approaches for the extensibility mechanism referenced above in 3904. One such approach involves exposing a set of application program interfaces (APIs) so that independent software vendors (ISVs) and other third party developers may develop alternative previewers. With the API approach, a registration mechanism exists which allows an ISV to associate their preview control with an item type owned by the ISV. When an item or file of that type is selected in the shell browser, the ISV's preview control is instantiated via this registration mechanism and the extensibility API. The API provides data to the preview control: data representing the selected item(s) in the view and data representing the parent container of the items in the view. The preview control operates on this data and provides a user interface through the API which is presented in the shell browser. The user may provide input with keystrokes and mouse events which are passed by the shell browser to the preview control which can operate on those user input events.

Those skilled in the art will appreciate that many approaches are possible in the context of the extensibility mechanism of the present invention. In addition to the API approach, similar functionality may be achieved via user configuration, a pointer to HTML or hosting a flash. Moreover, the extensibility model may require that only one application that owns the item type selected may provide only one alternative previewer. In other words, the number of available previewers may be limited to a default previewer and one alternative previewer to avoid a poor user experience in which multiple registered, extended previewers are in competition with one another. However, another model would be to allow any application that can handle the selected item type to provide one additional previewer. An alternative model would allow any running code to provide one additional previewer for any item type. It may also be desirable under certain circumstances to allow replacement or removal of the default previewer. Many other models are possible and are contemplated by the present invention.

FIGURE 66 is a block diagram of a data structure 4000 which is stored on one or more computer-readable media and which contains information indicative of a plurality of previewers in a shell browser. The data structure 4000 includes a default previewer field 4002 containing information indicative of a default previewer which supports multiple item types. An alternative previewer field 4004 contains information indicative of an alternative previewer for a first item type. Another alternative previewer field 4006 may contain information indicative of a second alternative previewer for the first item type, or it may contain information indicative of an alternative previewer for a second item type. Those skilled in the art will appreciate that in some cases there may only be one alternative previewer field, and in other cases there may be two or more alternative previewer fields. The selected previewer field 4008 contains information indicative of whether to invoke the default previewer or an alternative previewer when items of a particular item type are displayed in the shell browser. In the event that field 4006 contains information indicative of an alternative previewer for a second item type, a selected previewer field 4010 may contain information indicative of whether to invoke the default previewer or the alternative previewer when one or more items of the second item type are displayed in the shell browser. The information contained in fields 4002, 4004 and/or 4006 may comprise the previewer code which is configured to run when a user selects an object of that type.

While the preferred embodiment of the invention has been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the invention. For example, it will be appreciated that the locations of the various UI features that are shown herein are illustrative and may
5 be altered, and that different placements of the various UI features will still fall within the spirit and scope of the invention. Furthermore, the different aspects of the invention described herein may be formed in various combinations, also without departing from the spirit and scope of the invention.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. In a computer system with a display and a memory for storing items, a method for presenting collections of the items, comprising:

creating a library to contain items with one or more specified metadata properties;
automatically grouping items with the one or more specified metadata properties into the library; and

providing one or more tools for manipulating items in the library.

2. The method of Claim 1, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

displaying a plurality of items;
receiving a first input from the user representing a selection of at least one displayed item;

displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

3. The method of Claim 1, further comprising providing a shell browser, the shell browser comprising:

a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and

an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

4. The method of Claim 1, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

providing a shell browser having a default previewer for the multiple item types;
and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

5. The method of Claim 1, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

6. The method of Claim 1, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

7. The method of Claim 1, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

8. The method of Claim 1, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

9. The method of Claim 1, further comprising a method for displaying items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;

receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

10. The method of Claim 1, further comprising a method for representing the items to a user, the method for representing the items to the user comprising:

providing a database that allows both non-file items and file items to be searched by a query;

receiving a query, and in response to the query drawing both non-file items and file items that match the query; and

presenting both the non-file items and file items that match the query on the display.

11. A system for exposing items that are stored in a memory to a user, comprising:

a folder processor that obtains queries from a user;

a relational database for storing information about items; and

wherein:

the folder processor obtains a query from a user and passes the query to the relational database;

the relational database provides results back to the folder processor; and

based on the results from the relational database, the folder processor provides the results to the user as virtual folders.

12. The system of Claim 11, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

displaying a plurality of items;

receiving a first input from the user representing a selection of at least one displayed item;

displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

13. The system of Claim 11, further comprising a shell browser, the shell browser comprising:

a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and

an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

14. The system of Claim 11, further comprising a previewer for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer comprising:

a shell browser having a default previewer for the multiple item types; and

an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

15. The system of Claim 11, further comprising a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

16. The system of Claim 11, further comprising a virtual address bar for providing a virtual path to items stored in a computer file system, the virtual address bar comprising:

a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

one or more additional interactive segments of the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

17. The system of Claim 11, further comprising a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

18. The system of Claim 11, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and
modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

19. The system of Claim 11, further comprising a method for presenting collections of the items, the method of presenting collections of the items comprising:

creating a library to contain items with one or more specified metadata properties;
automatically grouping items with the one or more specified metadata properties into the library; and
providing one or more tools for manipulating items in the library.

20. The system of Claim 11, further comprising a method for displaying items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;

receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

21. In a computer system having a display and a memory for storing items, the items having associated metadata properties, a method of exposing the items to a user, the method comprising:

selecting a first metadata property;

searching for items that have the selected first metadata property; and

providing a first virtual folder display object that represents the collection of items that have the first metadata property:

22. The method of Claim 21, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

- displaying a plurality of items;
- receiving a first input from the user representing a selection of at least one displayed item;
- displaying metadata associated with the at least one selected item; and
- providing a control for user modification of the displayed metadata.

23. The method of Claim 21, further comprising providing a shell browser, the shell browser comprising:

- a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and
- an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

24. The method of Claim 21, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

- providing a shell browser having a default previewer for the multiple item types;
- and
- providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

25. The method of Claim 21, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

- a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

26. The method of Claim 21, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

27. The method of Claim 21, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

28. The method of Claim 21, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

29. The method of Claim 21, further comprising a method for presenting collections of the items, the method of presenting collections of the items comprising:

- creating a library to contain items with one or more specified metadata properties;
- automatically grouping items with the one or more specified metadata properties into the library; and
- providing one or more tools for manipulating items in the library.

30. The method of Claim 21, further comprising a method for displaying items, the method for displaying items comprising:

- defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;
- receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and
- presenting the items drawn from the query in a view on the display.

31. In a computer system having a display and a memory for storing items, a method for manipulating the items, comprising:

- representing groups of items as virtual folders; and
- providing at least a first defined action that can be performed for direct manipulation of a virtual folder, wherein when the first defined action is performed, the virtual folder is manipulated as directed by the performed action.

32. The method of Claim 31, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

- displaying a plurality of items;
- receiving a first input from the user representing a selection of at least one displayed item;
- displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

33. The method of Claim 31, further comprising providing a shell browser, the shell browser comprising:

a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and

an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

34. The method of Claim 31, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

providing a shell browser having a default previewer for the multiple item types; and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

35. The method of Claim 31, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

36. The method of Claim 31, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

37. The method of Claim 31, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

38. The method of Claim 31, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

39. The method of Claim 31, further comprising a method for presenting collections of the items, the method of presenting collections of the items comprising:

creating a library to contain items with one or more specified metadata properties;

automatically grouping items with the one or more specified metadata properties into the library; and

providing one or more tools for manipulating items in the library.

40. The method of Claim 31, further comprising a method for displaying items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;

receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

41. In a computer system having a display and a memory for storing items with metadata properties, a method of filtering items, the method comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

42. The method of Claim 41, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

displaying a plurality of items;

receiving a first input from the user representing a selection of at least one displayed item;

displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

43. The method of Claim 41, further comprising providing a shell browser, the shell browser comprising:

a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and

an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

44. The method of Claim 41, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

providing a shell browser having a default previewer for the multiple item types;
and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

45. The method of Claim 41, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

46. The method of Claim 41, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

47. The method of Claim 41, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

48. The method of Claim 41, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

49. The method of Claim 41, further comprising a method for presenting collections of the items, the method of presenting collections of the items comprising:

creating a library to contain items with one or more specified metadata properties;

automatically grouping items with the one or more specified metadata properties into the library; and

providing one or more tools for manipulating items in the library.

50. The method of Claim 41, further comprising a method for displaying items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;

receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

51. In a computer system having a display and a memory for storing items, a method of viewing selected items, the method comprising:

- storing a link to a desired collection of items that the user has navigated to;
- presenting the link on the display such that by clicking on the link a user can return to the view of the desired collection of items.

52. The method of Claim 51, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

- displaying a plurality of items;
- receiving a first input from the user representing a selection of at least one displayed item;
- displaying metadata associated with the at least one selected item; and
- providing a control for user modification of the displayed metadata.

53. The method of Claim 51, further comprising providing a shell browser, the shell browser comprising:

- a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and
- an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

54. The method of Claim 51, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

- providing a shell browser having a default previewer for the multiple item types;
- and
- providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

55. The method of Claim 51, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

56. The method of Claim 51, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

57. The method of Claim 51, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

58. The method of Claim 51, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and
modifying the display objects that are provided on the display so as to represent
the items that remain after the filtering of the items.

59. The method of Claim 51, further comprising a method for presenting
collections of the items, the method of presenting collections of the items comprising:

creating a library to contain items with one or more specified metadata properties;
automatically grouping items with the one or more specified metadata properties
into the library; and
providing one or more tools for manipulating items in the library.

60. The method of Claim 51, further comprising a method for displaying
items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be
drawn, the scope comprising a present computer memory and at least one other physical
location;

receiving a query, and in response to the query, drawing items from the physical
locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

61. In a computer system with a display and a memory for storing items, a
method for representing the items to a user, comprising:

providing a database that allows both non-file items and file items to be searched
by a query;

receiving a query, and in response to the query drawing both non-file items and
file items that match the query; and

presenting both the non-file items and file items that match the query on the
display.

62. The method of Claim 61, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

- displaying a plurality of items;
- receiving a first input from the user representing a selection of at least one displayed item;
- displaying metadata associated with the at least one selected item; and
- providing a control for user modification of the displayed metadata.

63. The method of Claim 61, further comprising providing a shell browser, the shell browser comprising:

- a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and
- an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

64. The method of Claim 61, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

- providing a shell browser having a default previewer for the multiple item types;
- and
- providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

65. The method of Claim 61, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

- a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

66. The method of Claim 61, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

67. The method of Claim 61, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

68. The method of Claim 61, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

69. The method of Claim 61, further comprising a method for presenting collections of the items, the method of presenting collections of the items comprising:

- creating a library to contain items with one or more specified metadata properties;
- automatically grouping items with the one or more specified metadata properties into the library; and
- providing one or more tools for manipulating items in the library.

70. The method of Claim 61, further comprising a method for displaying items, the method for displaying items comprising:

- defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;
- receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and
- presenting the items drawn from the query in a view on the display.

71. A method for providing a virtual path to items stored in a computer file system using a virtual address bar, the method comprising:

- displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and
- adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;
- wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

72. The method of Claim 71, wherein a filter is provided on the display that can be selected by a user to filter items based on the metadata property specified by the filter.

73. The method of Claim 71, wherein a quick link is provided on the display that can be selected by a user to view a specified collection of items.

74. The method of Claim 71, wherein a library is provided that includes a collection of items and a set of tools for manipulating the items of the library.

75. The method of Claim 71, wherein the items in a virtual folder may be physically stored in memories at different physical locations.

76. The method of Claim 75, wherein the different physical locations comprise a present computer and at least one of a different computer, a location on a network, and an external storage device.

77. The method of Claim 71, wherein the items comprise both file items and non-file items.

78. The method of Claim 77, wherein the non-file items comprise at least one of contacts or e-mails.

79. The method of Claim 71, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

displaying a plurality of items;

receiving a first input from the user representing a selection of at least one displayed item;

displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

80. The method of Claim 71, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

providing a shell browser having a default previewer for the multiple item types;

and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

81. A computer-implemented method for enabling a user to modify metadata within a shell browser, the method comprising:

displaying a plurality of items;

receiving a first input from the user representing a selection of at least one displayed item;

displaying metadata associated with the at least one selected item; and

providing a control for user modification of the displayed metadata.

82. The method of Claim 81, wherein a filter is provided on the display that can be selected by a user to filter items based on the metadata property specified by the filter.

83. The method of Claim 81, wherein a quick link is provided on the display that can be selected by a user to view a specified collection of items.

84. The method of Claim 81, wherein a library is provided that includes a collection of items and a set of tools for manipulating the items of the library.

85. The method of Claim 81, wherein the items in a virtual folder may be physically stored in memories at different physical locations.

86. The method of Claim 85, wherein the different physical locations comprise a present computer and at least one of a different computer, a location on a network, and an external storage device.

87. The method of Claim 81, wherein the items comprise both file items and non-file items.

88. The method of Claim 87, wherein the non-file items comprise at least one of contacts or e-mails.

89. The method of Claim 81, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

90. The method of Claim 81, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

91. A computer-implemented method for enabling the use of third party previewers in a shell browser which supports multiple item types, the method comprising:

providing a shell browser having a default previewer for the multiple item types; and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

92. The method of Claim 91, wherein a filter is provided on the display that can be selected by a user to filter items based on the metadata property specified by the filter.

93. The method of Claim 91, wherein a quick link is provided on the display that can be selected by a user to view a specified collection of items.

94. The method of Claim 91, wherein a library is provided that includes a collection of items and a set of tools for manipulating the items of the library.

95. The method of Claim 91, wherein the items in a virtual folder may be physically stored in memories at different physical locations.

96. The method of Claim 95, wherein the different physical locations comprise a present computer and at least one of a different computer, a location on a network, and an external storage device.

97. The method of Claim 91, wherein the items comprise both file items and non-file items.

98. The method of Claim 97, wherein the non-file items comprise at least one of contacts or e-mails.

99. The method of Claim 91, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

100. The method of Claim 91, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

101. In a computer system with a display and a memory for storing items, the items being of a plurality of different types, a method for presenting libraries of the items, comprising:

providing libraries for containing items of similar types; and
providing a similar set of UI features for each of the libraries.

102. The method of Claim 101, wherein the different types comprise at least two of documents, images and music.

103. The method of Claim 101, wherein the similar set of UI features includes a feature for editing.

104. The method of Claim 101, wherein the similar set of UI features includes a feature for creating new categories.

105. The method of Claim 101, wherein the similar set of UI features includes a feature for editing the metadata of items.

106. The method of Claim 101, wherein the similar set of UI features includes a feature for altering the pivots.

107. The method of Claim 101, wherein in addition to containing items of similar types, the libraries may also contain items of different types.

108. The method of Claim 101, further comprising a shell browser method for enabling a user to modify metadata within a shell browser, the shell browser method comprising:

displaying a plurality of items;
receiving a first input from the user representing a selection of at least one displayed item;
displaying metadata associated with the at least one selected item; and
providing a control for user modification of the displayed metadata.

109. The method of Claim 101, further comprising providing a shell browser, the shell browser comprising:

a window which displays a group of items, wherein the window also displays metadata values associated with one or more of the displayed items; and

an edit control which permits user modification of at least a portion of the metadata values displayed in the window.

110. The method of Claim 101, further comprising a previewer method for enabling the use of third party previewers in a shell browser which supports multiple item types, the previewer method comprising:

providing a shell browser having a default previewer for the multiple item types; and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

111. The method of Claim 101, further comprising providing a shell browser configured to display a plurality of items representing multiple item types, the shell browser comprising:

a default previewer which provides a standard level of functionality for multiple item types; and

an extensibility mechanism which enables functionality beyond the standard level provided by the default previewer for one or more of the multiple item types.

112. The method of Claim 101, further comprising a virtual address bar method for providing a virtual path to items stored in a computer file system using a virtual address bar, the virtual address bar method comprising:

displaying a virtual address bar operable to include a first interactive segment, wherein the first interactive segment references items on the computer file system according to a corresponding filter for selecting items; and

adding one or more additional interactive segment to the virtual address bar, wherein each additional interactive segment further restricts the items referenced by the preceding interactive segments;

wherein the first interactive segment and the one or more additional interactive segments comprise a virtual path.

113. The method of Claim 101, further comprising providing a virtual address bar user interface control for selecting items accessible to a computer system for display, the virtual address bar comprising:

a plurality of interactive segments, wherein each interactive segment corresponds to a filter for selecting items for display.

114. The method of Claim 101, further comprising a method of filtering items, the method of filtering items comprising:

providing a plurality of display objects on the display that each represent one or more items;

filtering the items represented by the display objects based on a filter term; and

modifying the display objects that are provided on the display so as to represent the items that remain after the filtering of the items.

115. The method of Claim 101, further comprising a method for displaying items, the method for displaying items comprising:

defining a scope of the physical memory locations from which items are to be drawn, the scope comprising a present computer memory and at least one other physical location;

receiving a query, and in response to the query, drawing items from the physical locations as defined in the scope; and

presenting the items drawn from the query in a view on the display.

116. The method of Claim 101, further comprising a method for representing the items to a user, the method for representing the items to the user comprising:

providing a database that allows both non-file items and file items to be searched by a query;

receiving a query, and in response to the query drawing both non-file items and file items that match the query; and

presenting both the non-file items and file items that match the query on the display.

117. A computer-implemented method for enabling the use of third party previewers in a shell browser which supports multiple item types, the method comprising:

providing a shell browser having a default previewer for the multiple item types; and

providing an extensibility mechanism which enables a third party to develop an alternative previewer for at least one of the multiple item types.

118. The method of Claim 101, wherein a filter is provided on the display that can be selected by a user to filter items based on the metadata property specified by the filter.

119. The method of Claim 101, wherein a quick link is provided on the display that can be selected by a user to view a specified collection of items.

120. The method of Claim 101, wherein a library is provided that includes a collection of items and a set of tools for manipulating the items of the library.

121. The method of Claim 101, wherein the items in a virtual folder may be physically stored in memories at different physical locations.

122. The method of Claim 121, wherein the different physical locations comprise a present computer and at least one of a different computer, a location on a network, and an external storage device.

123. The method of Claim 101, wherein the items comprise both file items and non-file items.

124. The method of Claim 123, wherein the non-file items comprise at least one of contacts or e-mails.

ABSTRACT OF THE DISCLOSURE

A file system shell is provided. One aspect of the shell provides virtual folders which expose regular files and folders to users in different views based on their metadata instead of the actual physical underlying file system structure on the disk. Users are able to work with the virtual folders through direct manipulation (e.g., clicking and dragging, copying, pasting, etc.). Filters are provided for narrowing down sets of items. Quick links are provided which can be clicked on to generate useful views of the sets of items. Libraries are provided which consist of large groups of usable types of items that can be associated together, along with functions and tools related to the items. A virtual address bar is provided which comprises a plurality of segments, each segment corresponding to a filter for selecting content. A shell browser is provided with which users can readily identify an item based on the metadata associated with that item. An object previewer in a shell browser is provided which is configured to display a plurality of items representing multiple item types.

15

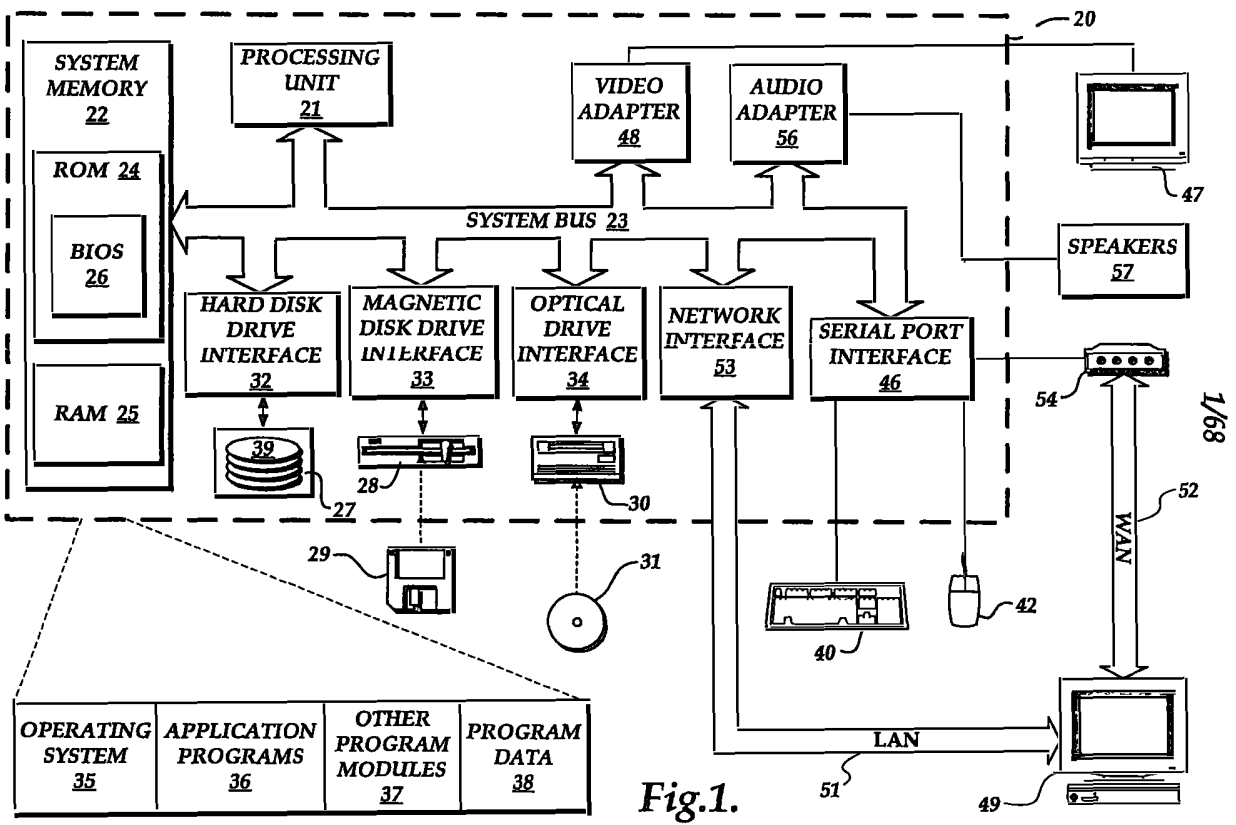


Fig.1.

339

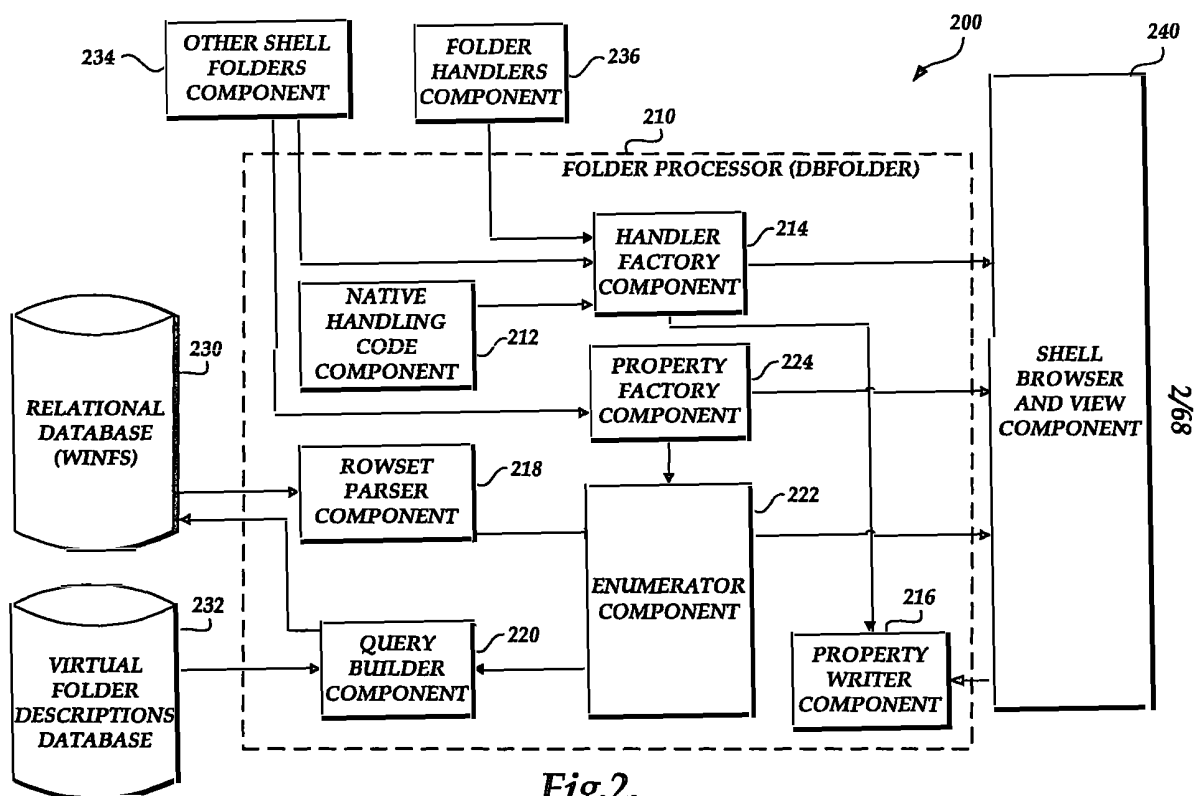


Fig.2.

340

3/68

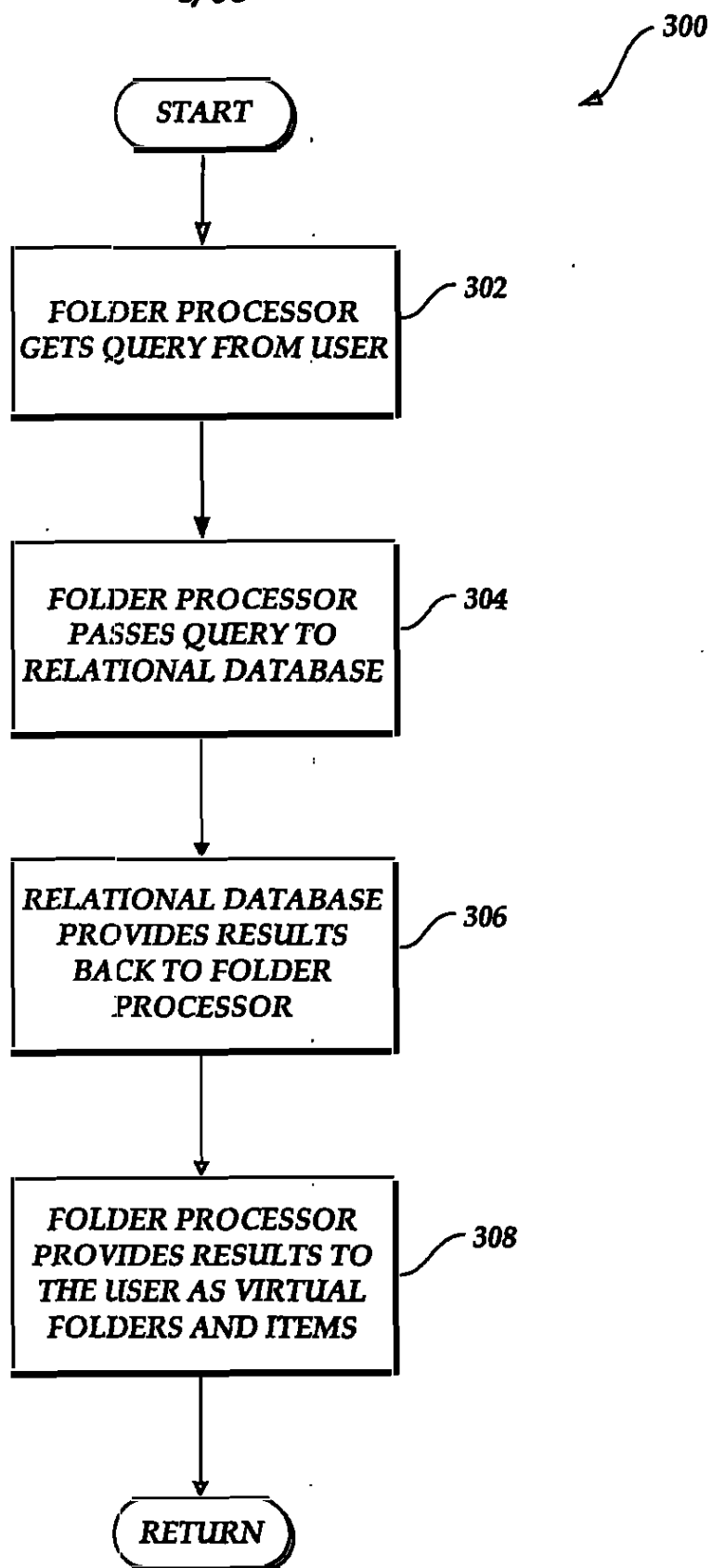


Fig.3.

4/68

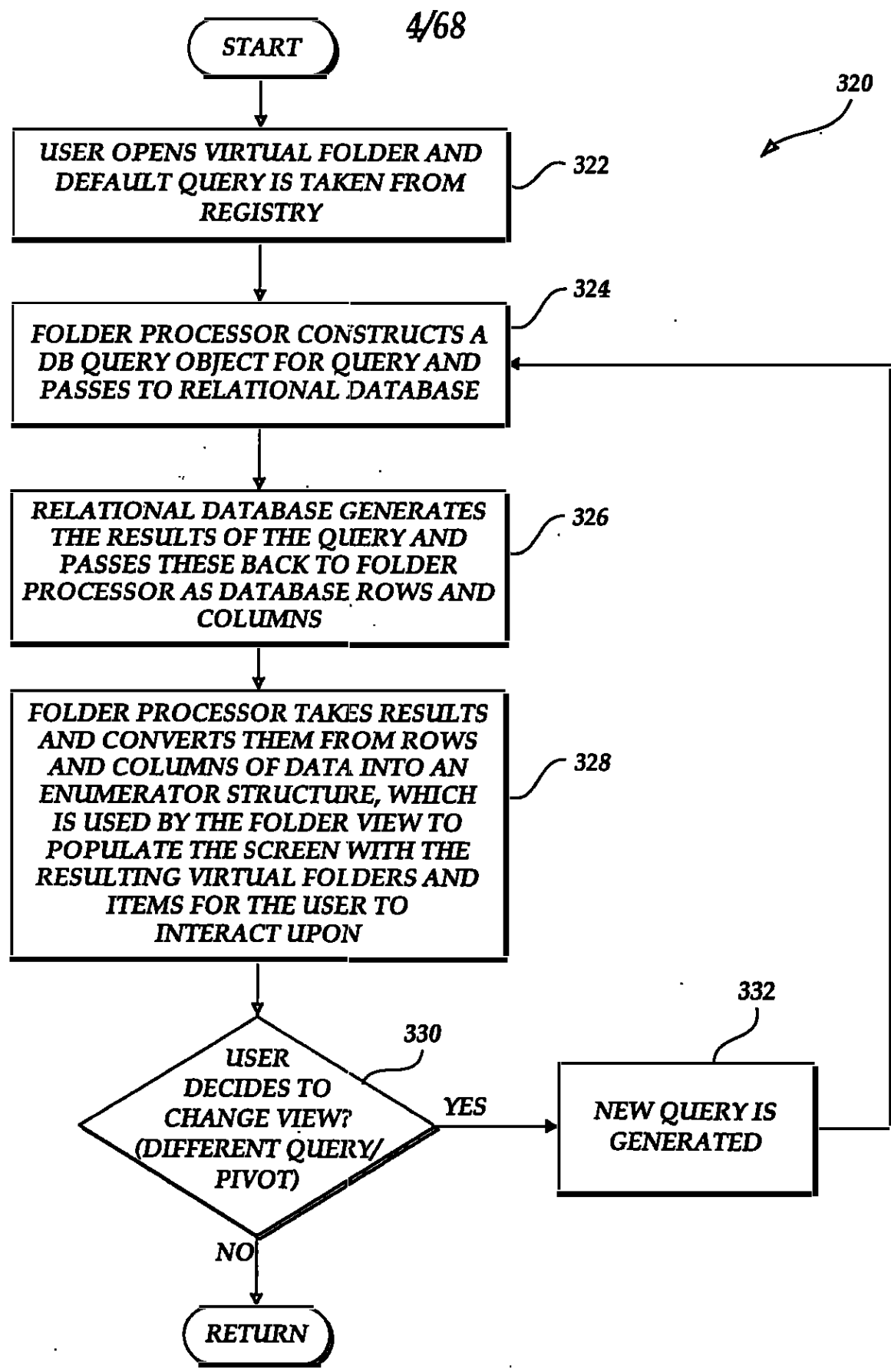


Fig.4.

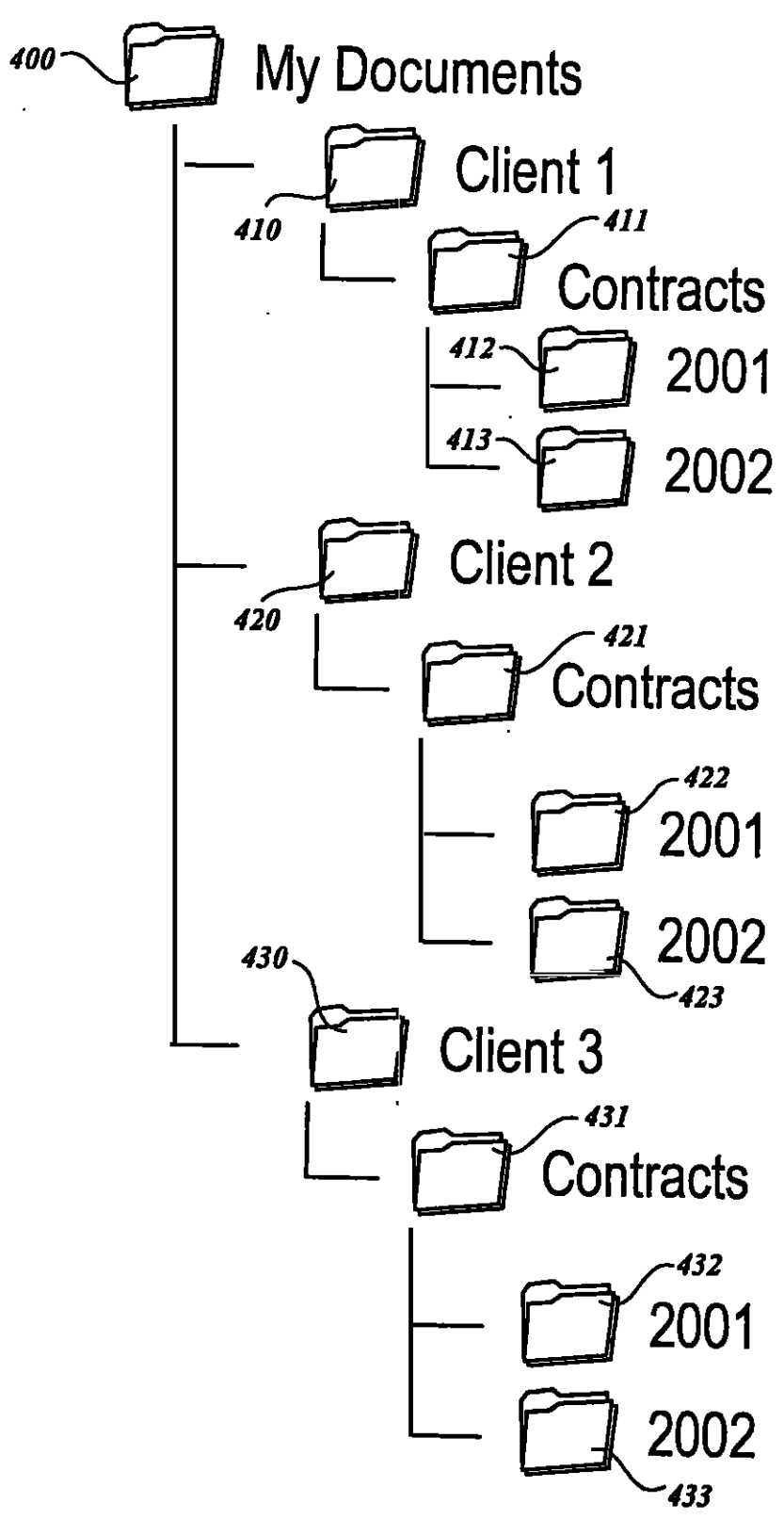
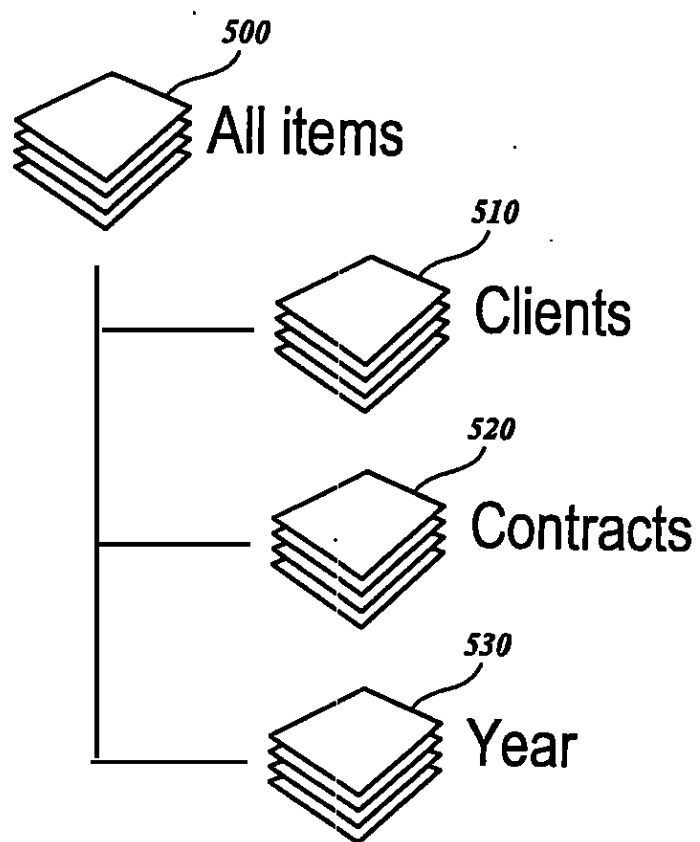
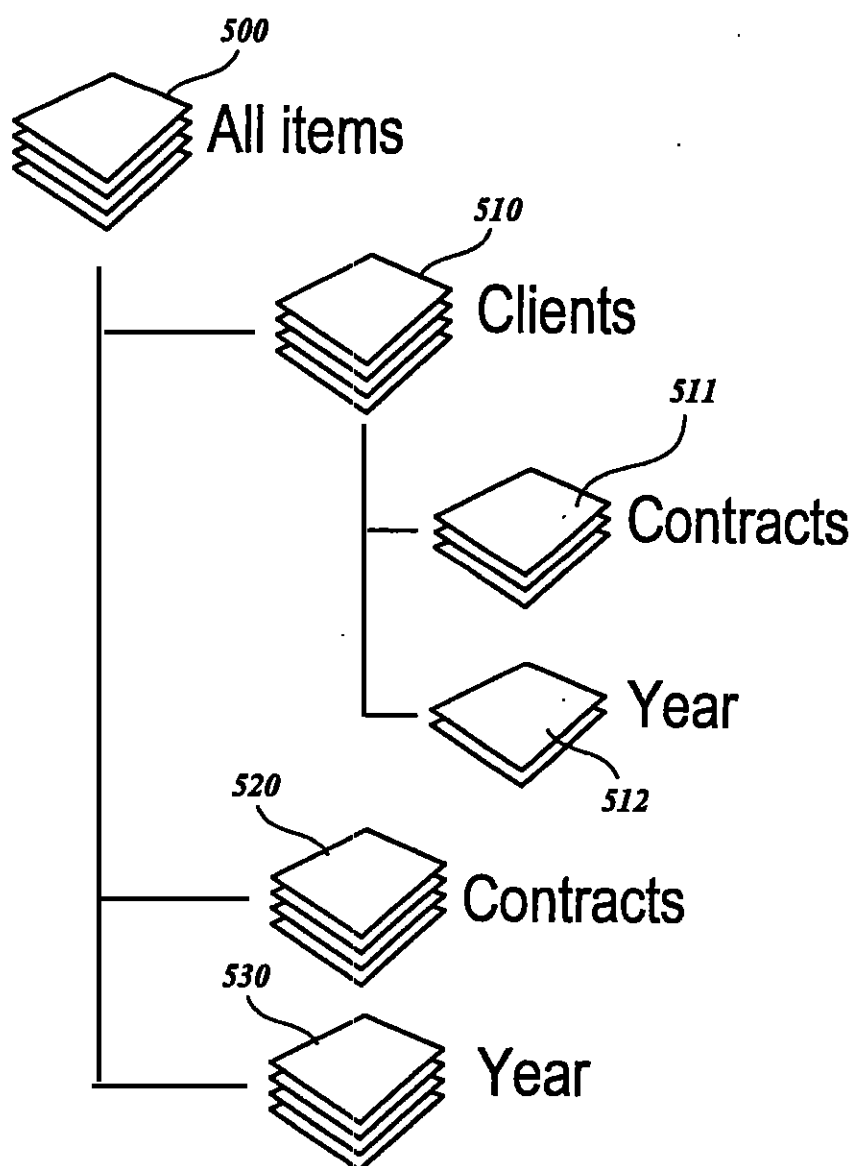


Fig.5.

6/68

*Fig.6.*

7/68

*Fig.7.*

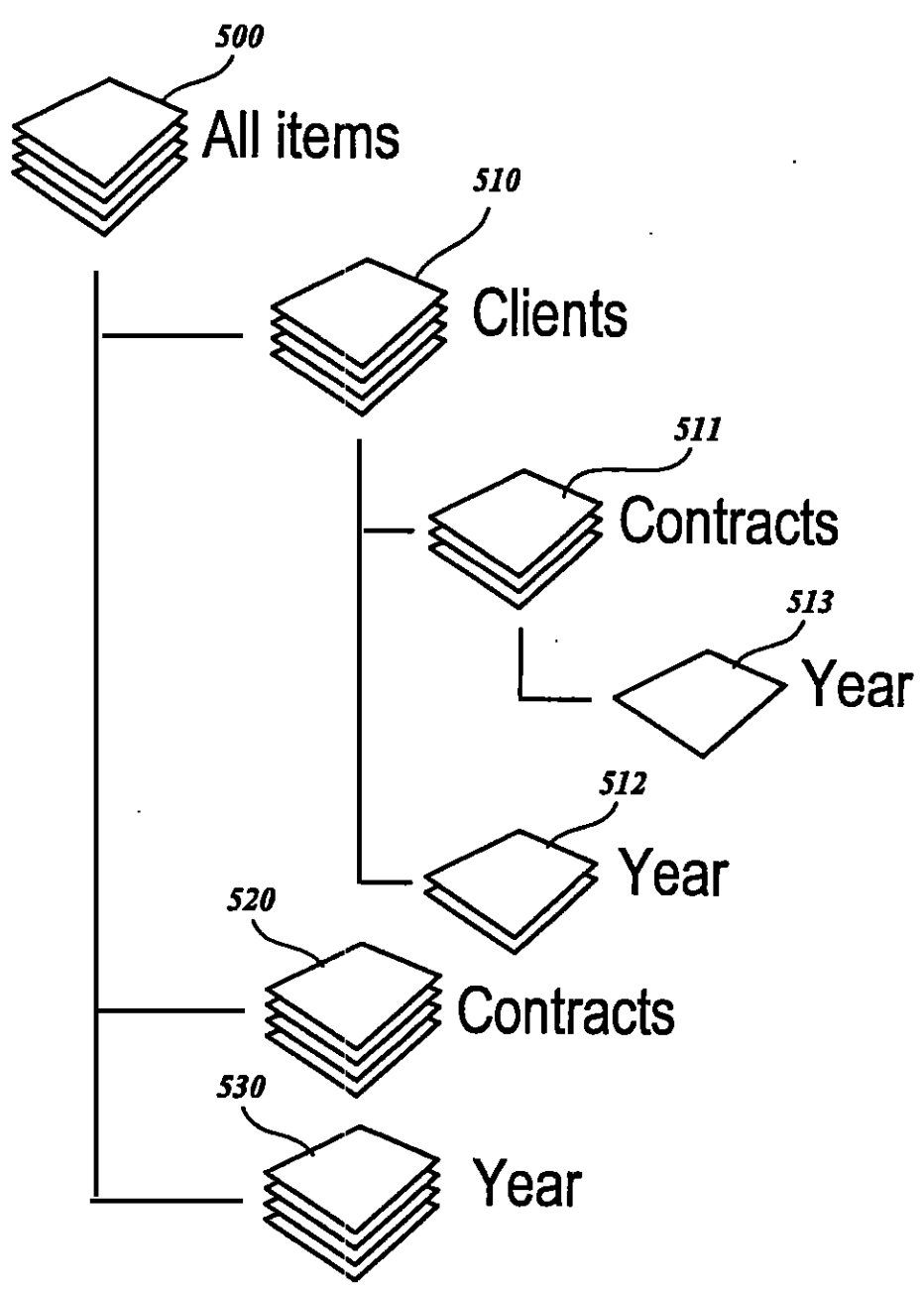


Fig.8.

9/68

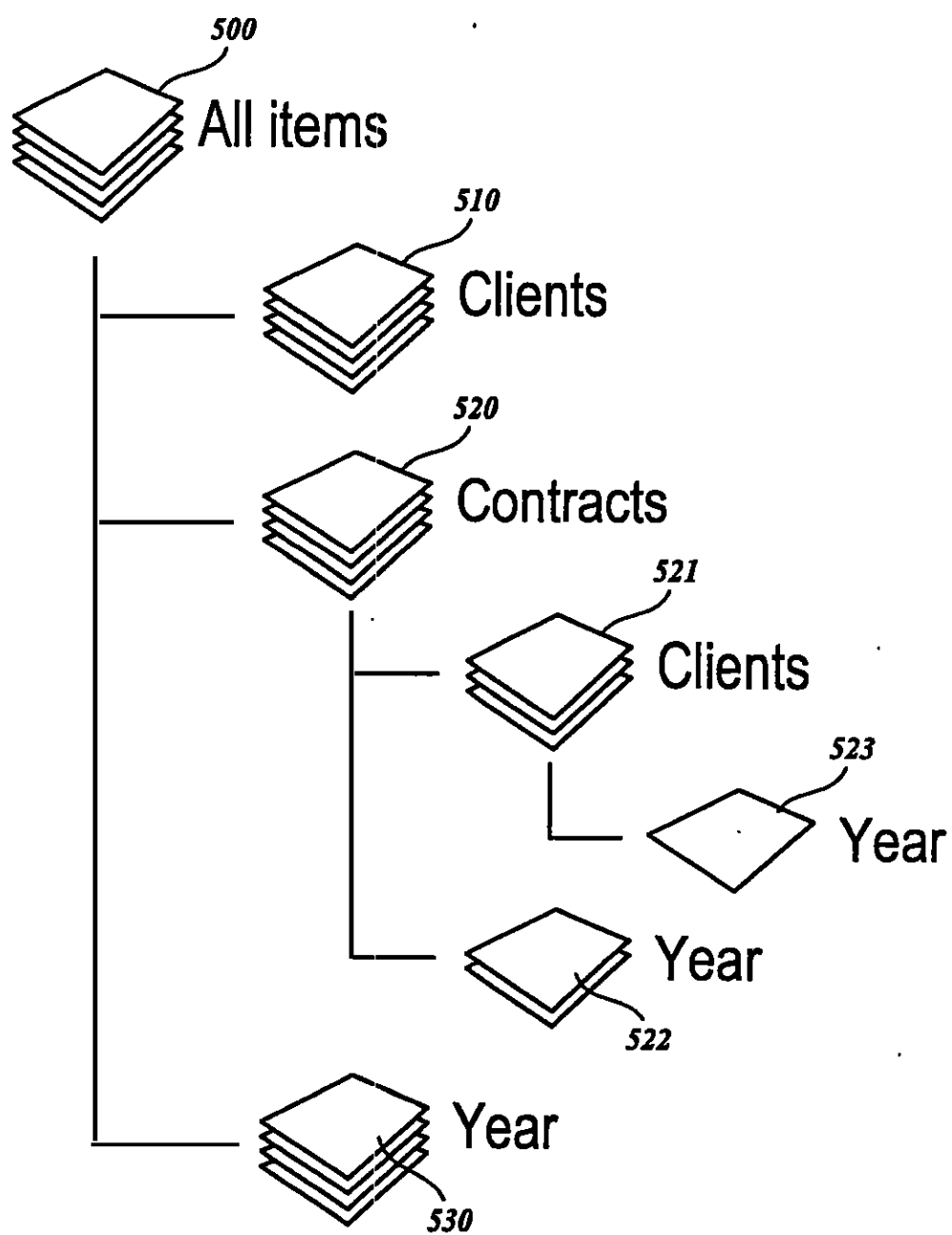


Fig.9.

348 348

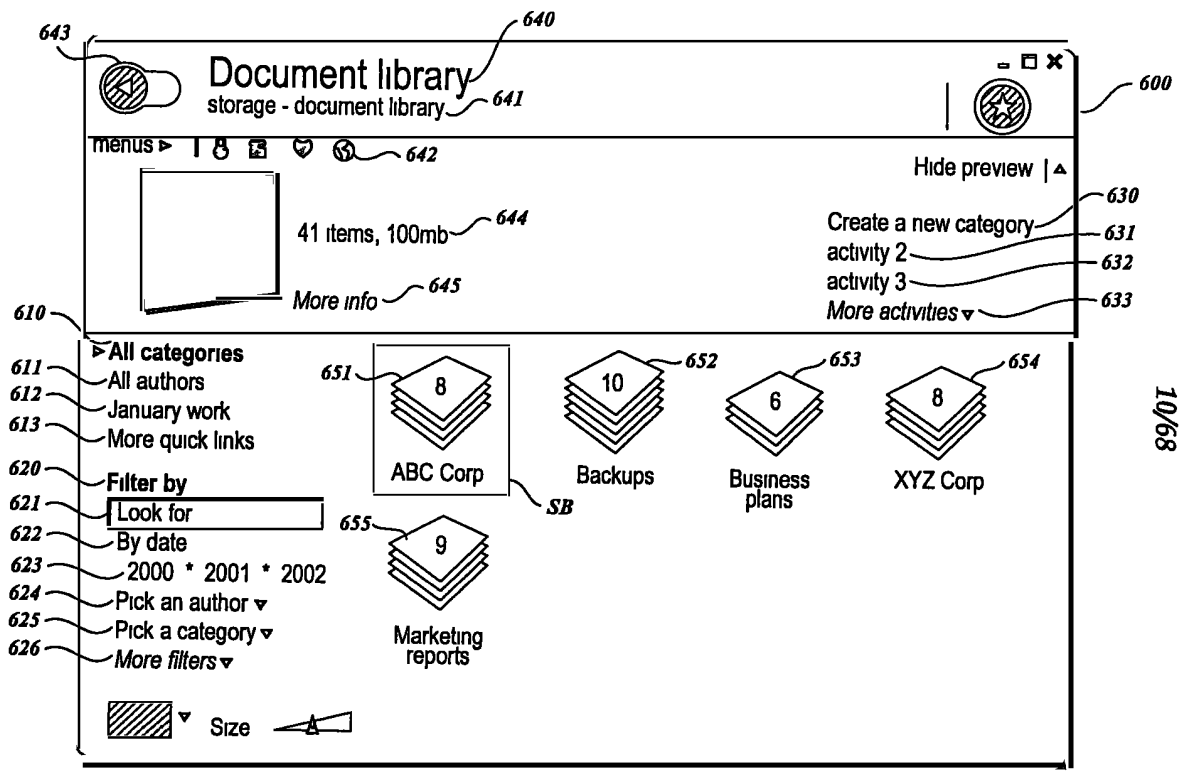


Fig.10.

348

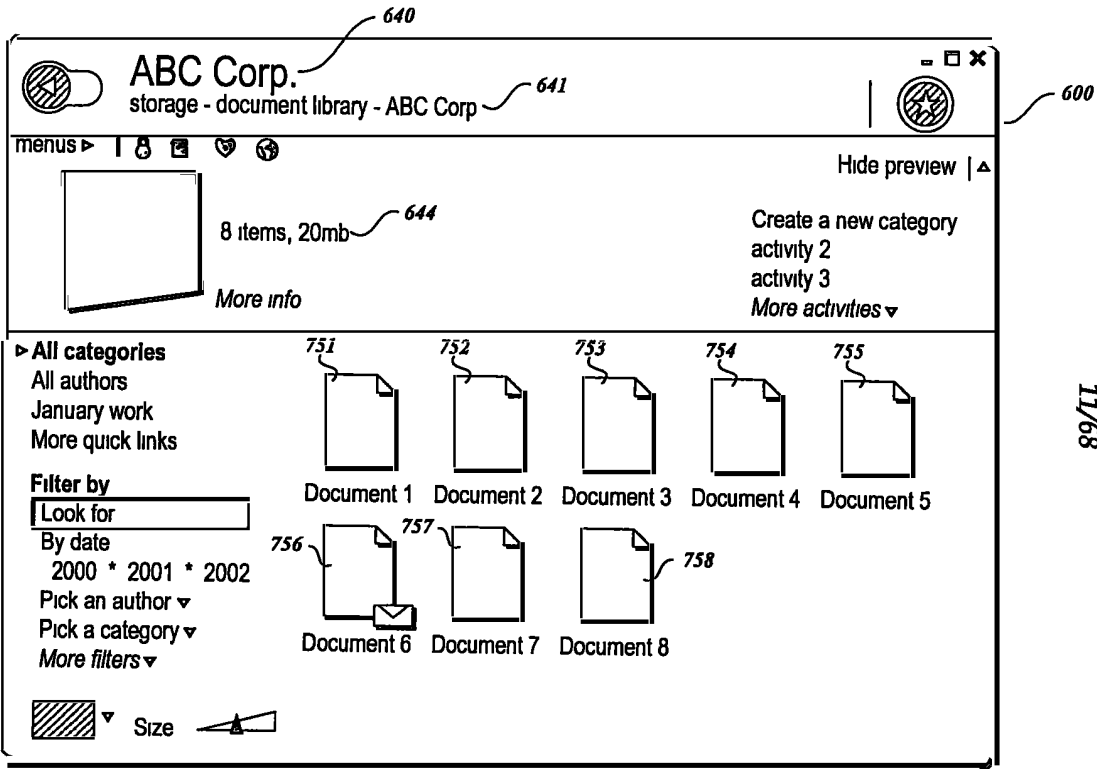
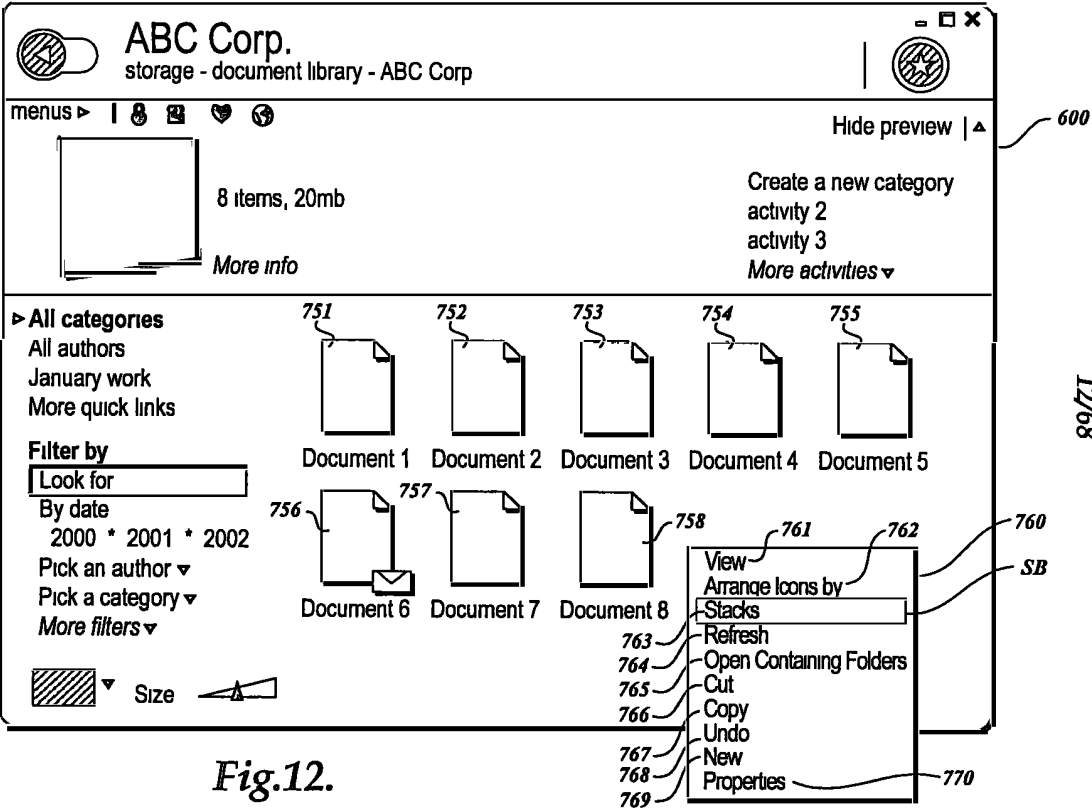
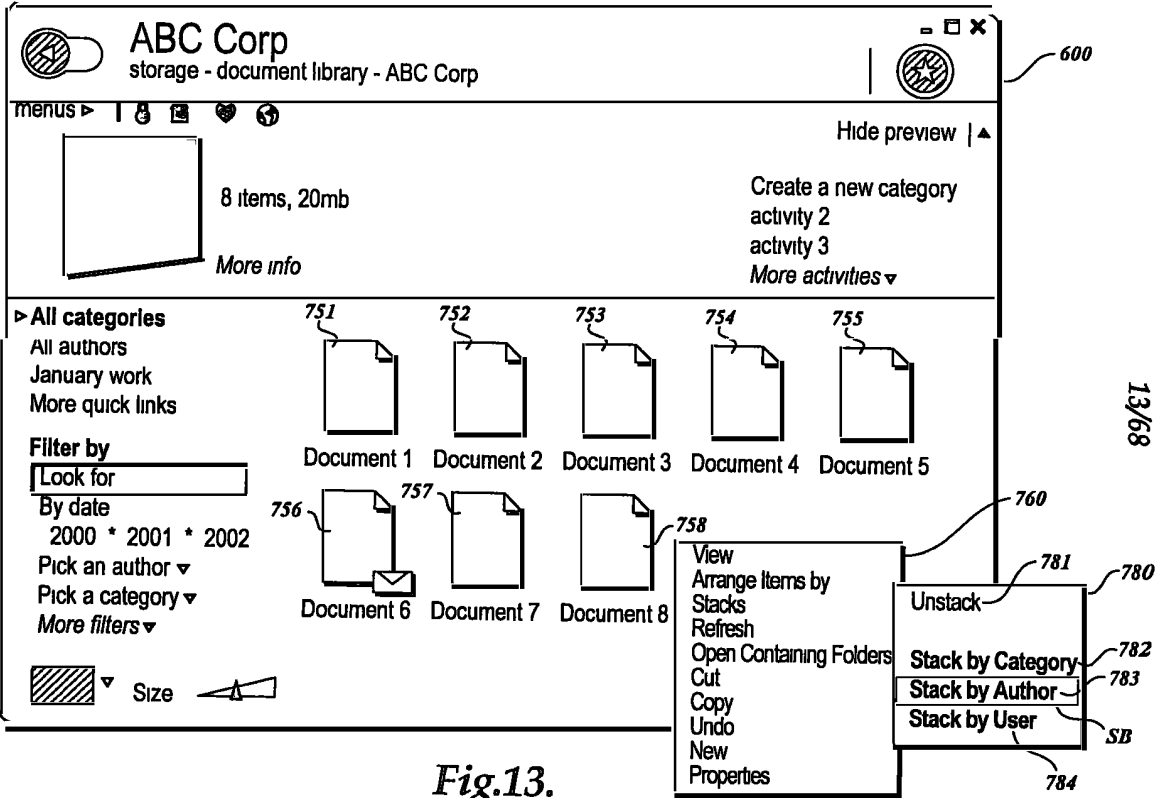


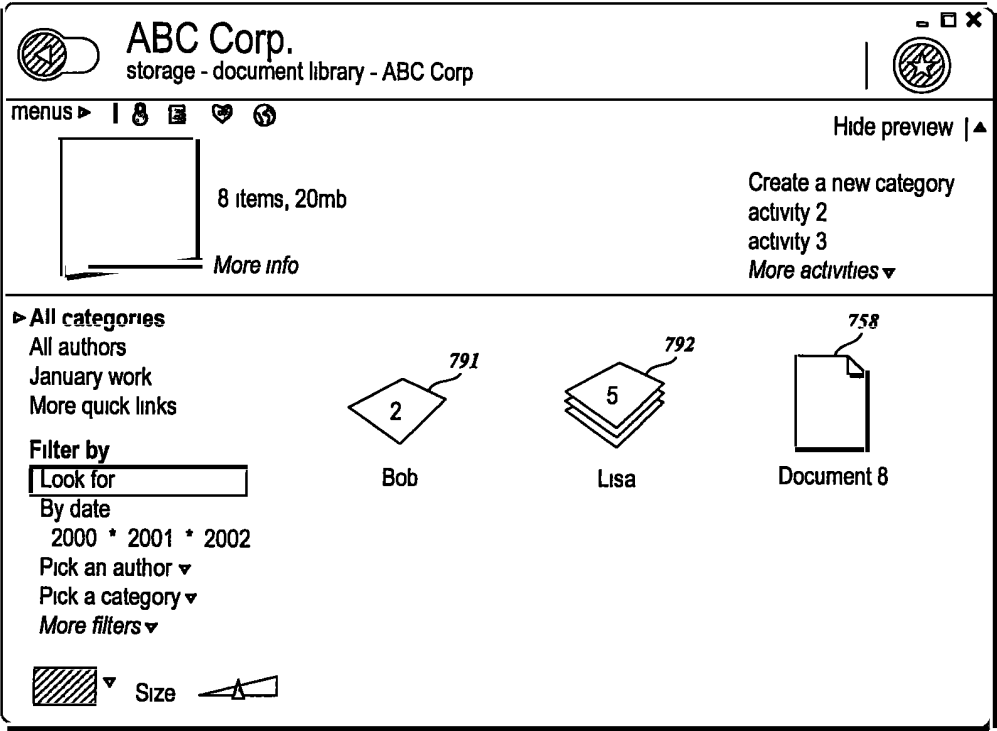
Fig.11.



850



951



600

14/68

Fig.14.

962

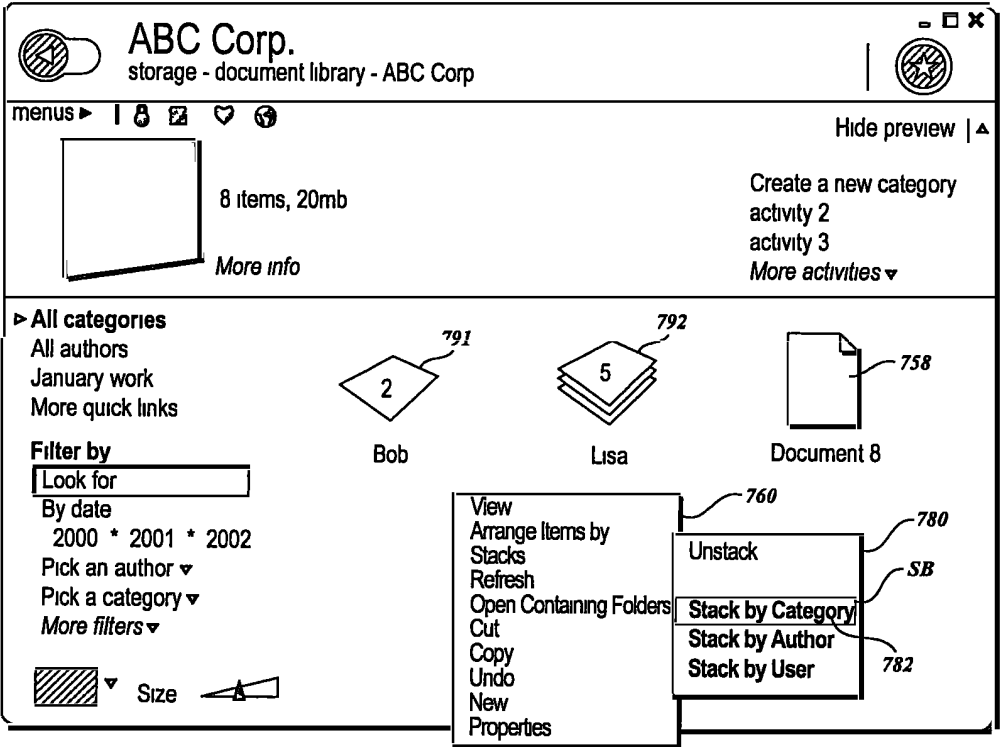


Fig.15.

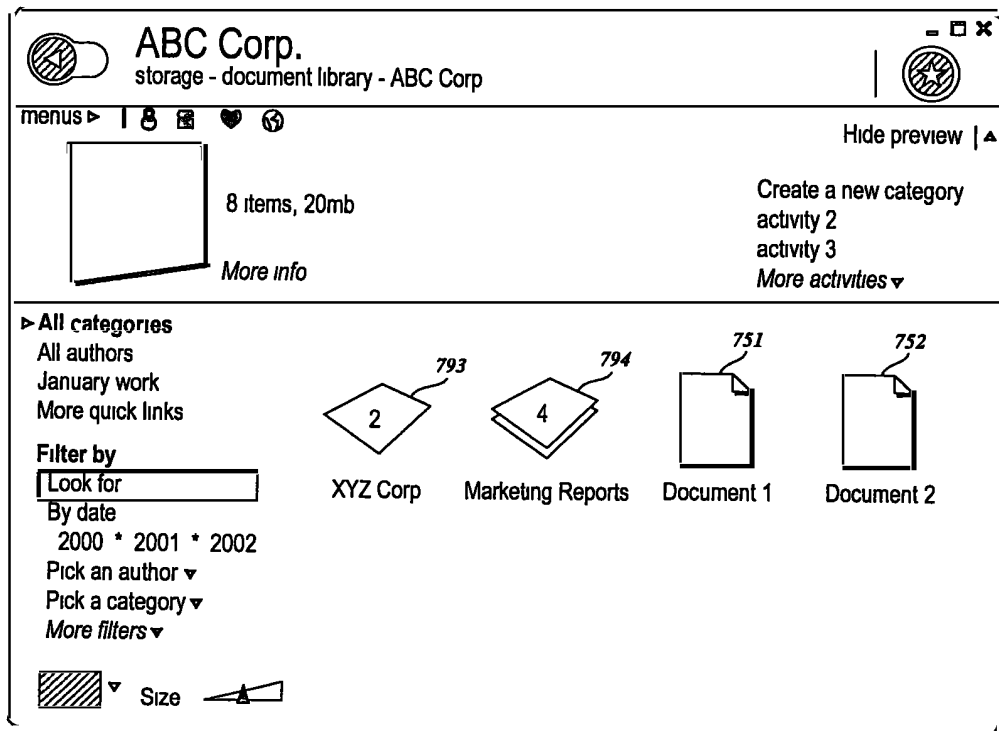


Fig.16.

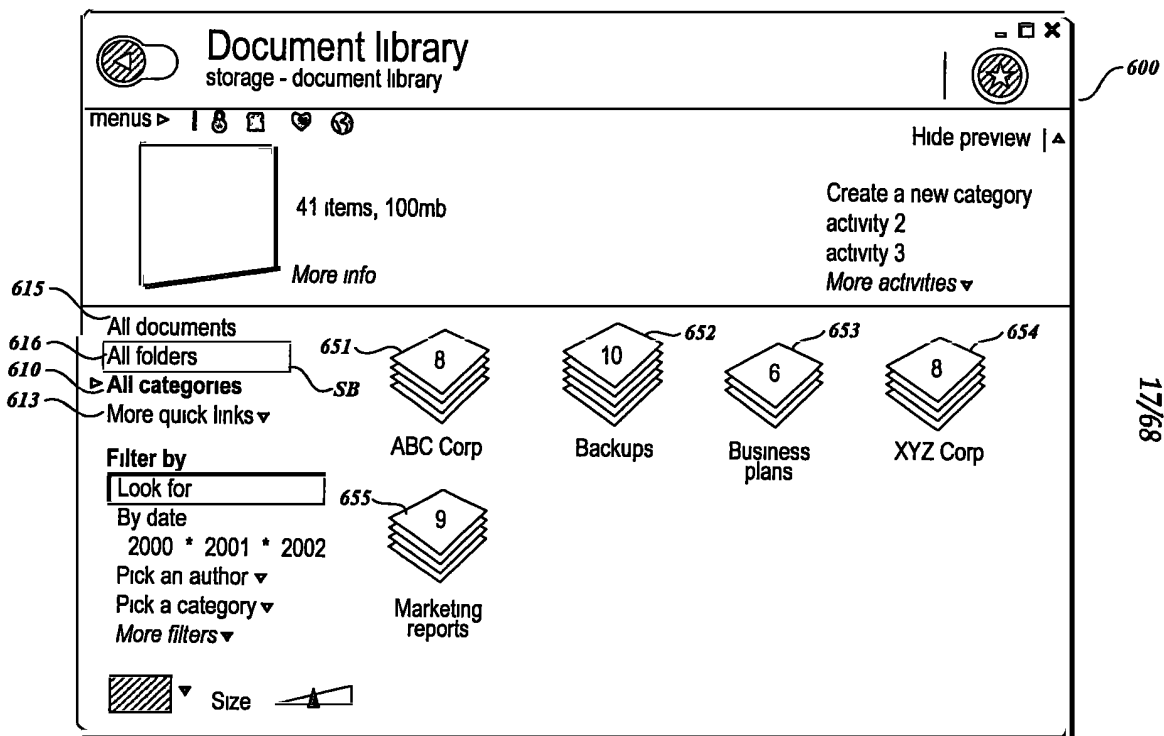


Fig.17.

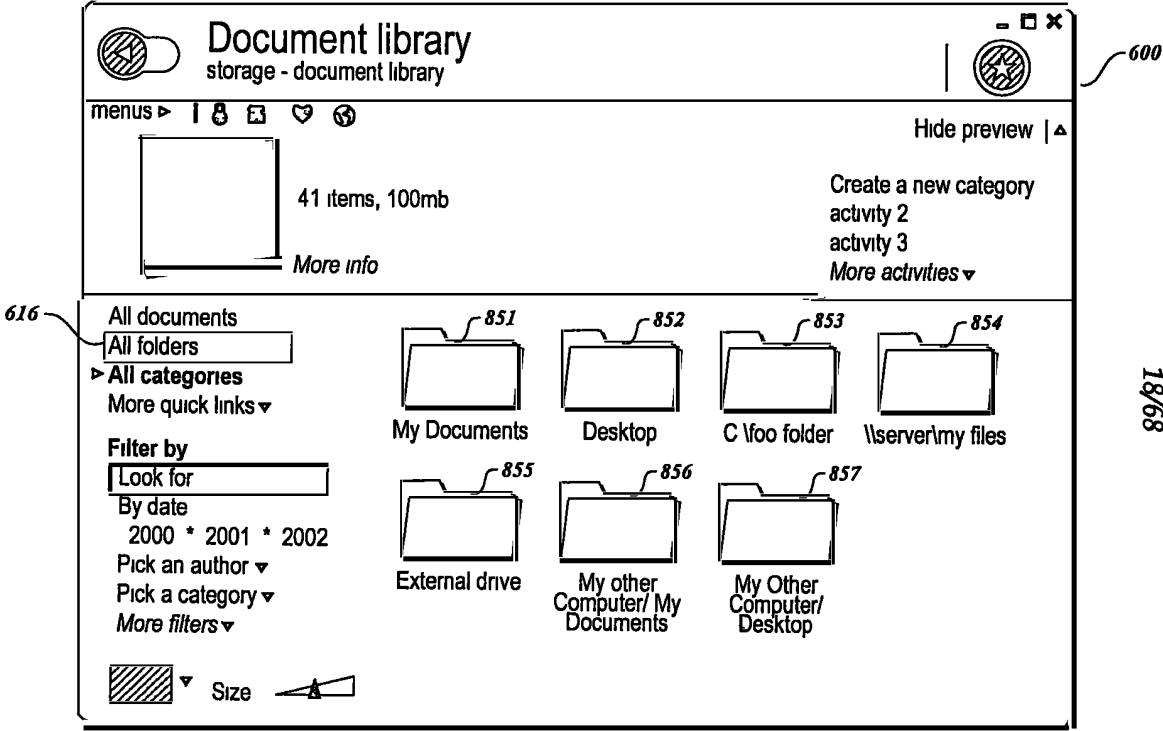


Fig.18.

356

19/68

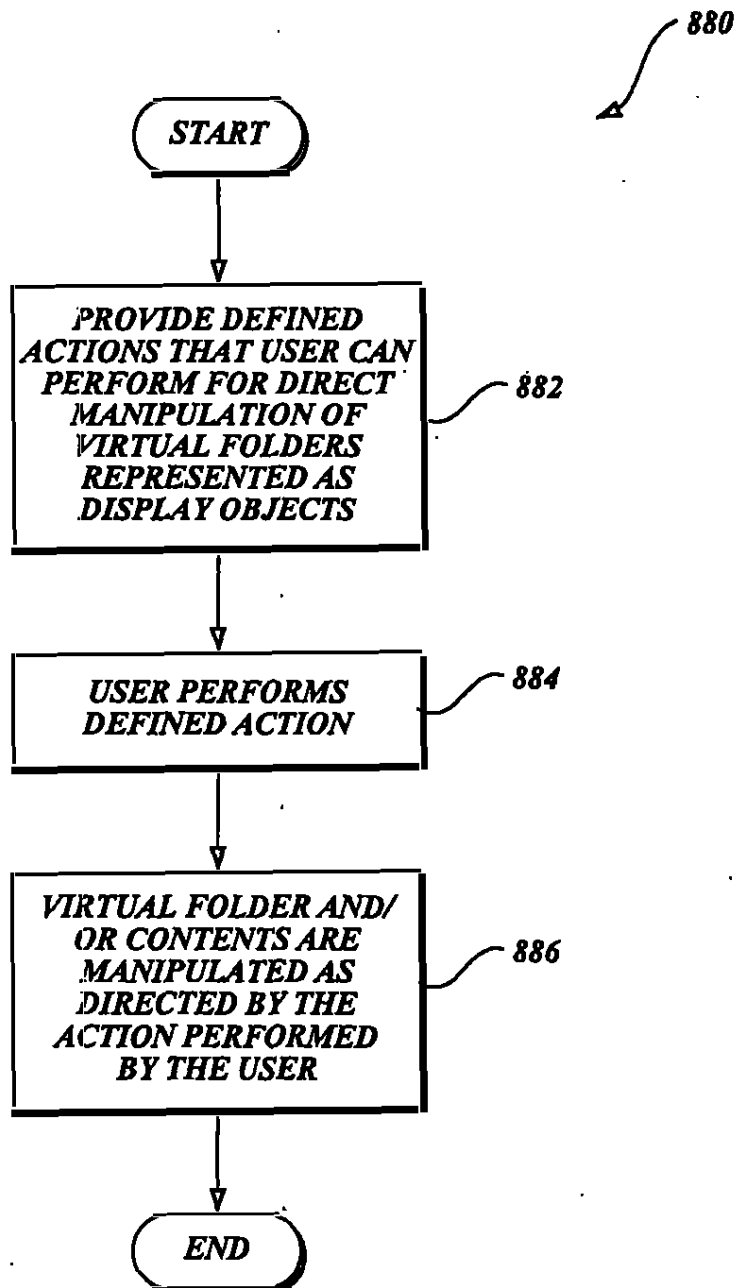


Fig.19.

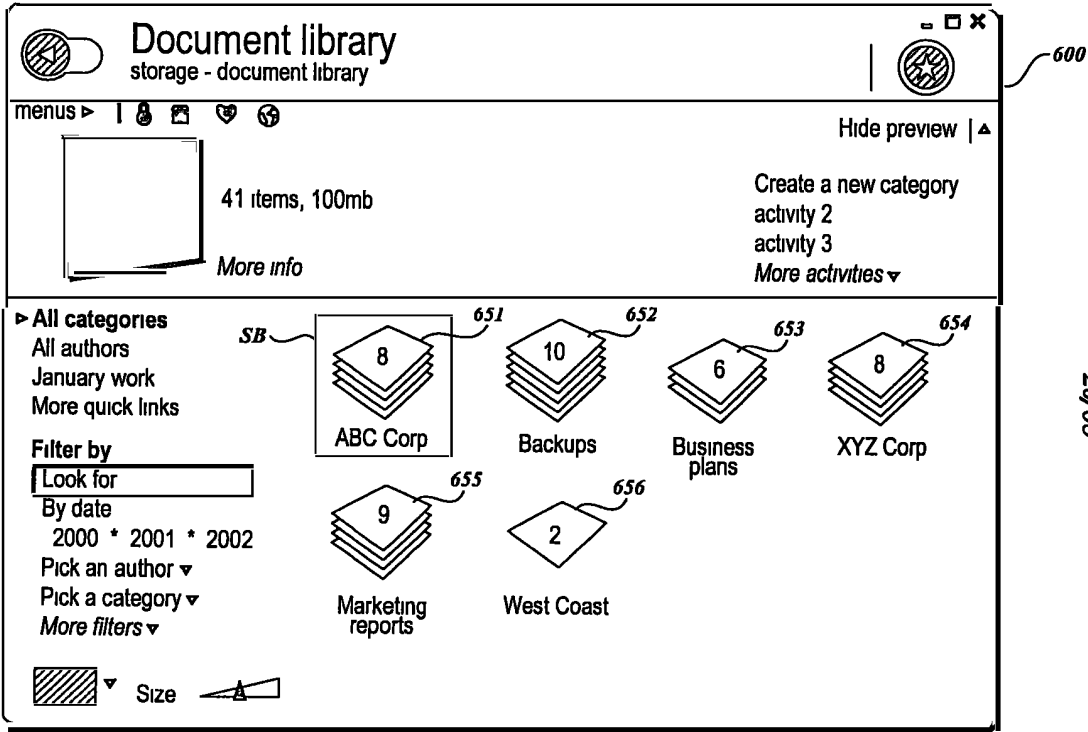


Fig.20.

258

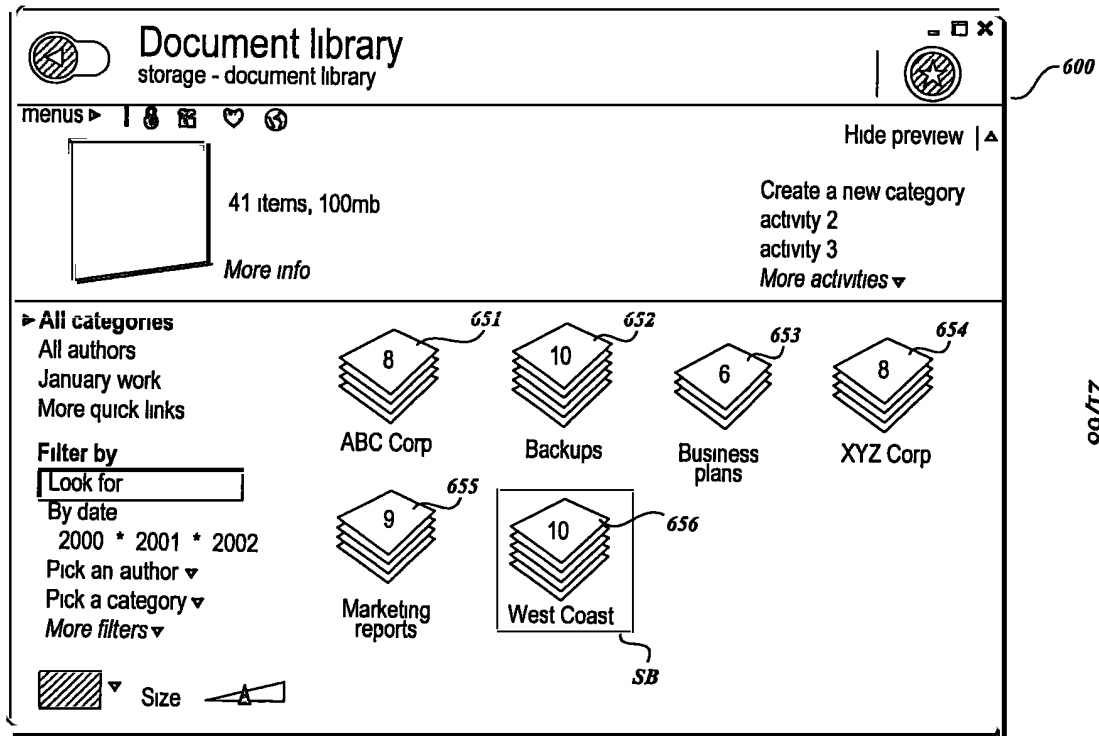


Fig.21.

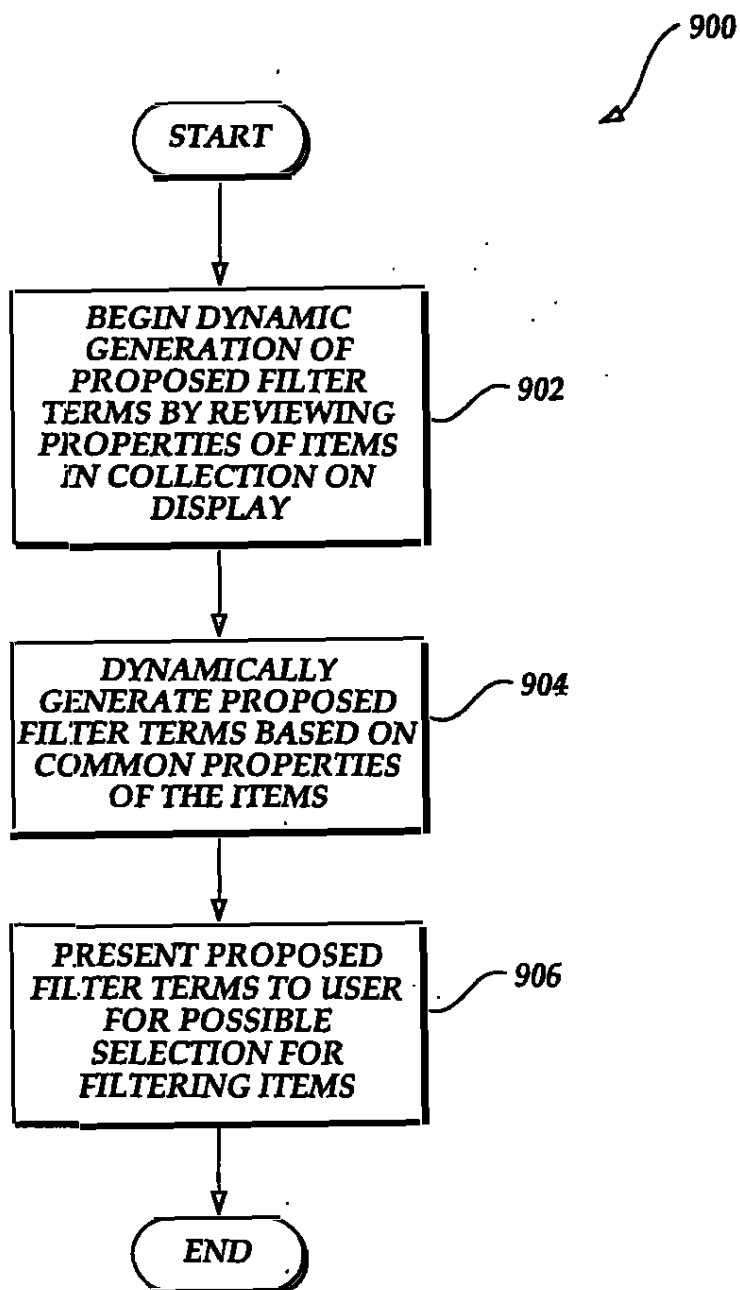


Fig.22.

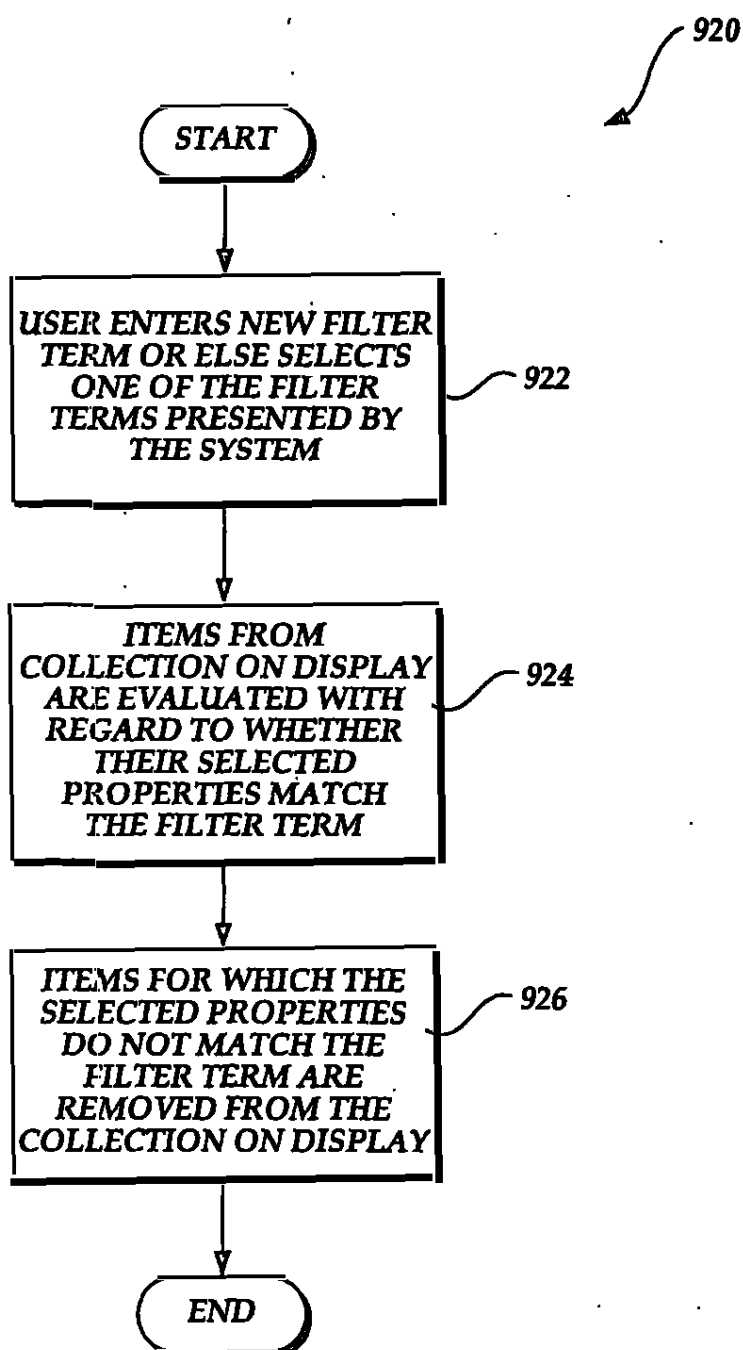


Fig.23.

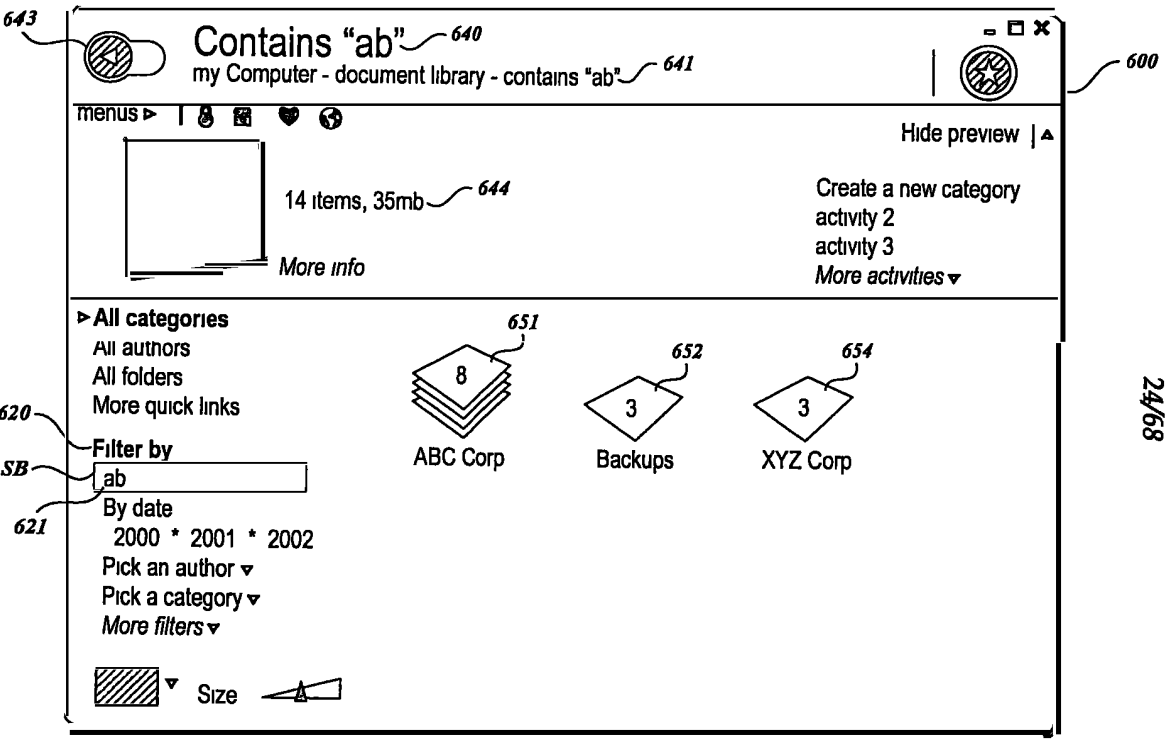


Fig.24.

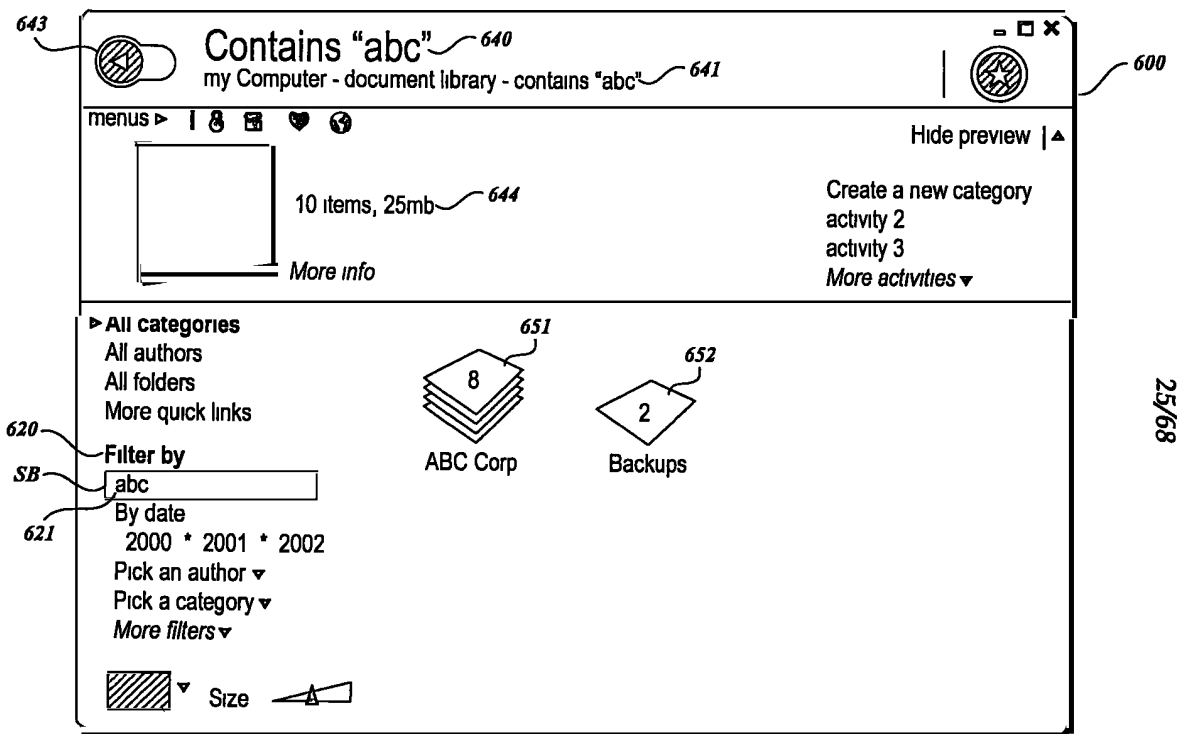


Fig.25.

363

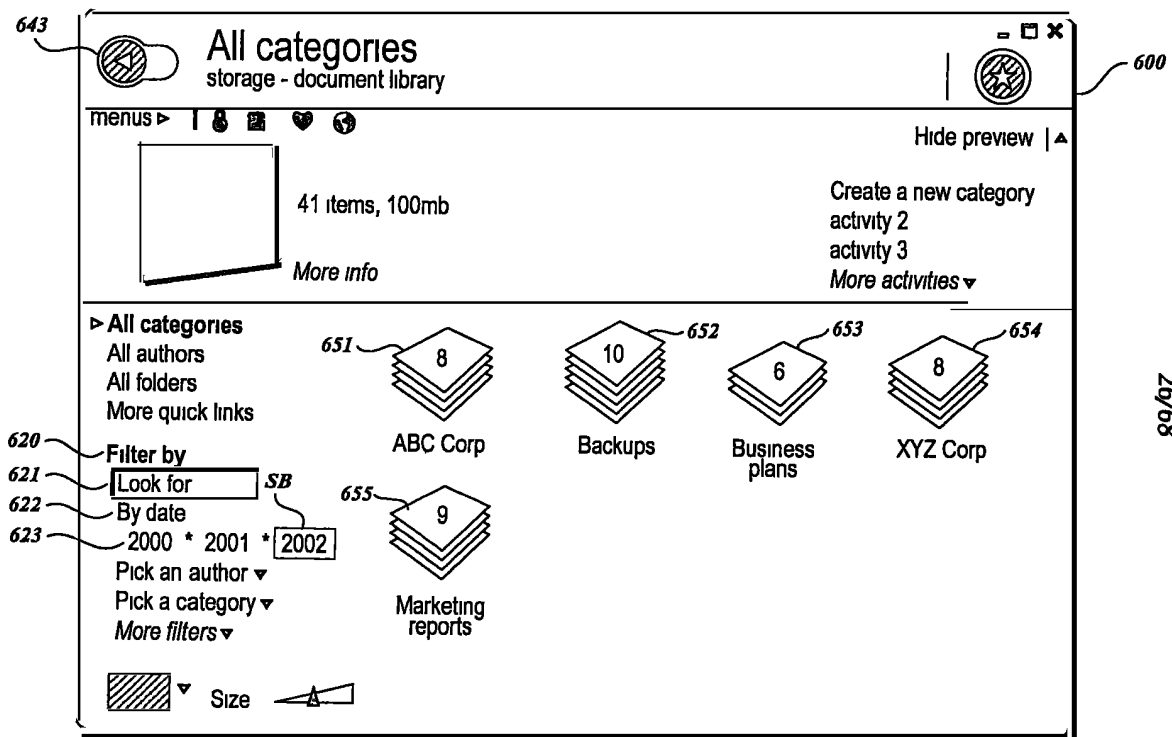


Fig.26.

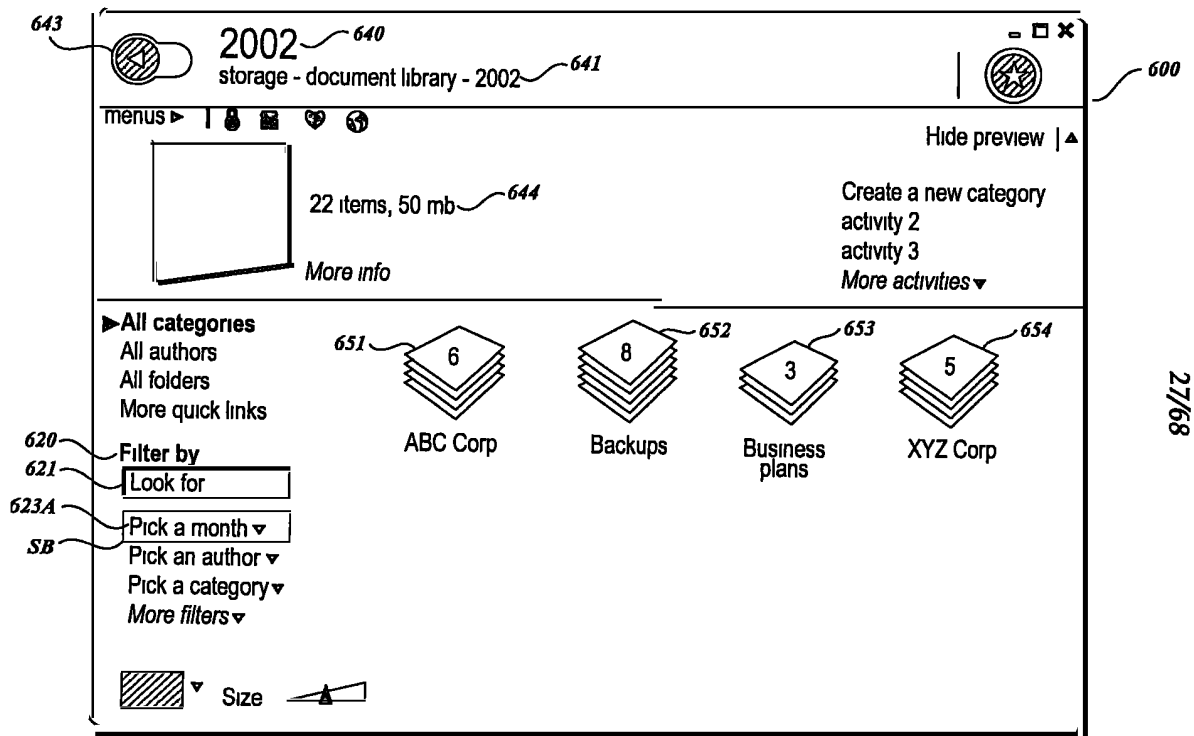


Fig.27.

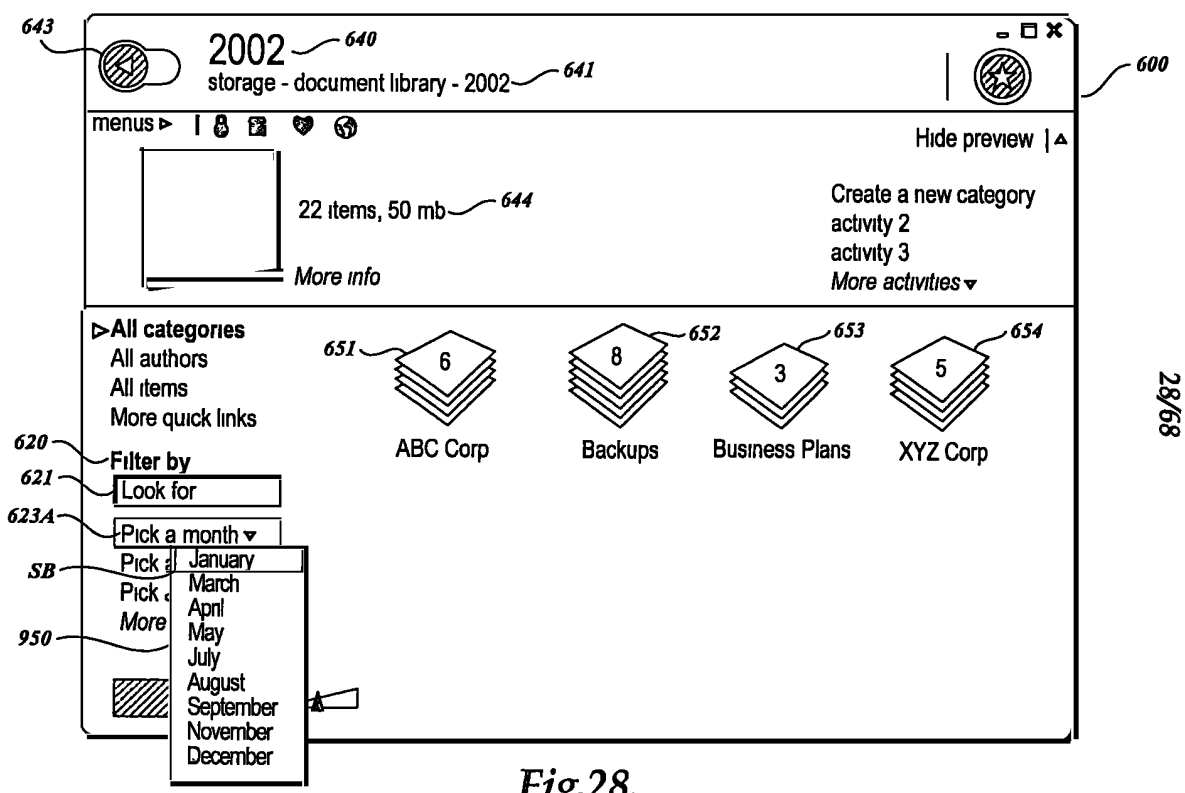


Fig. 28.

266

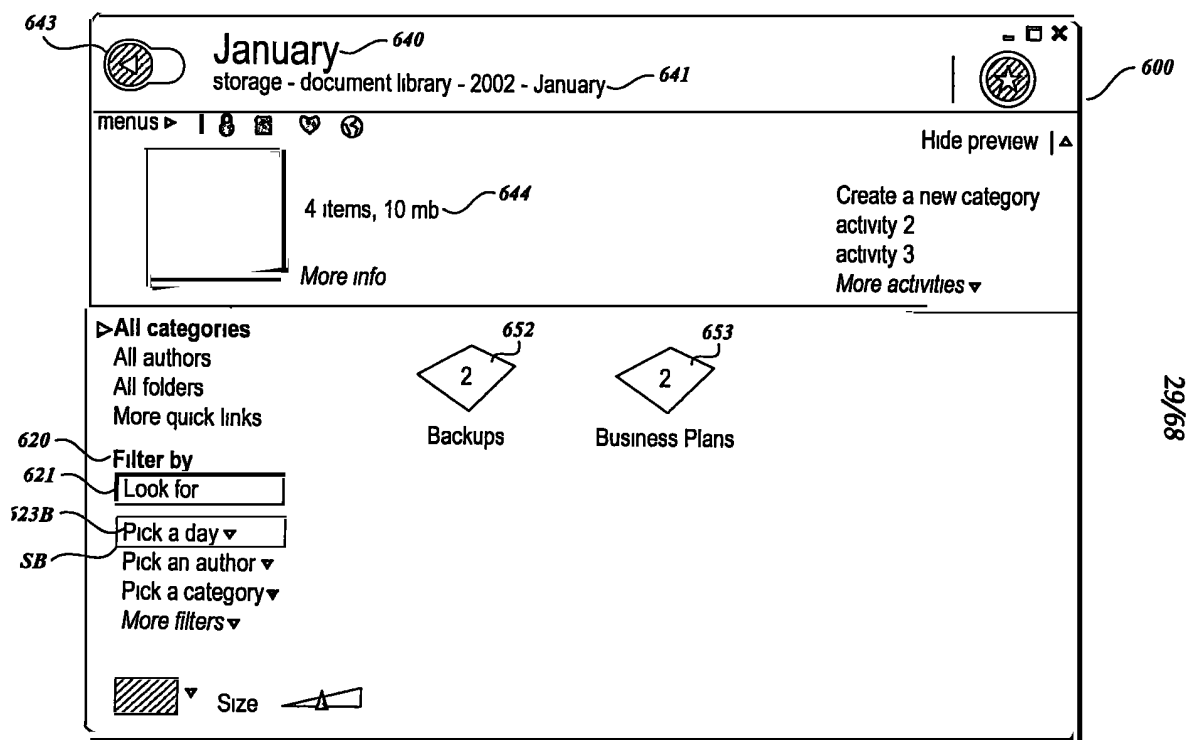


Fig.29.

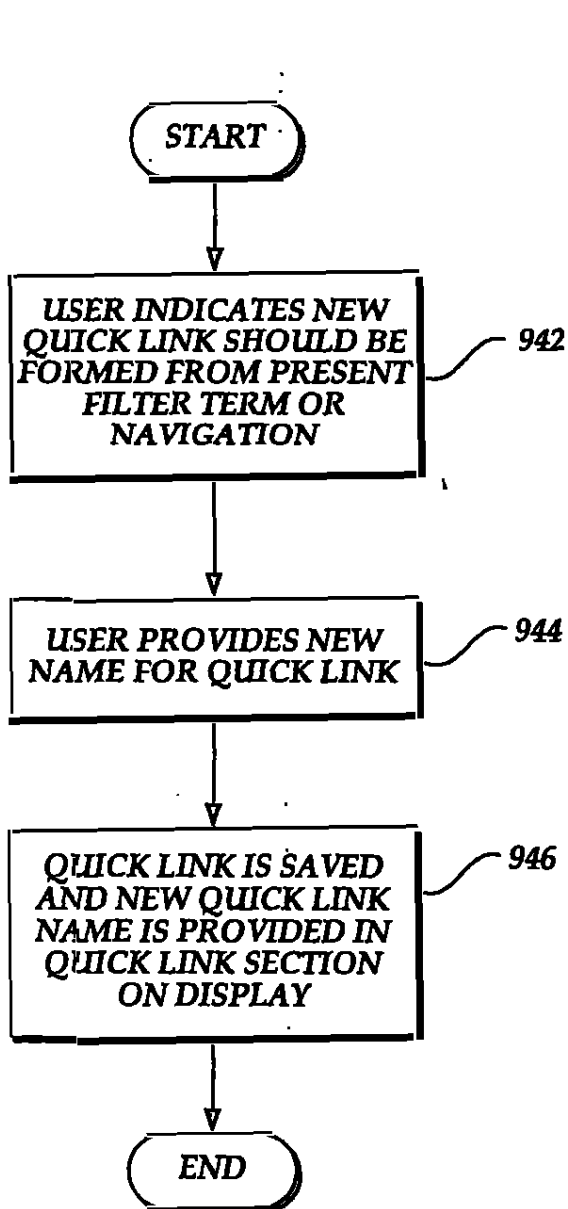


Fig.30.

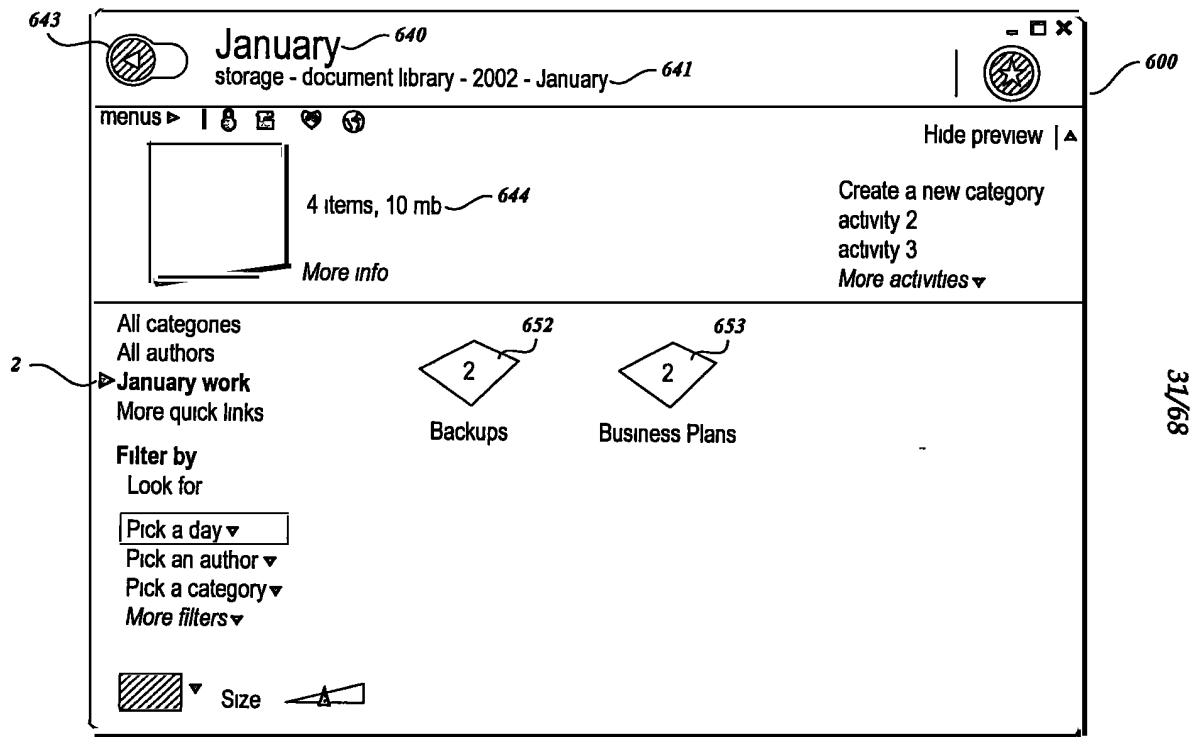


Fig.31.

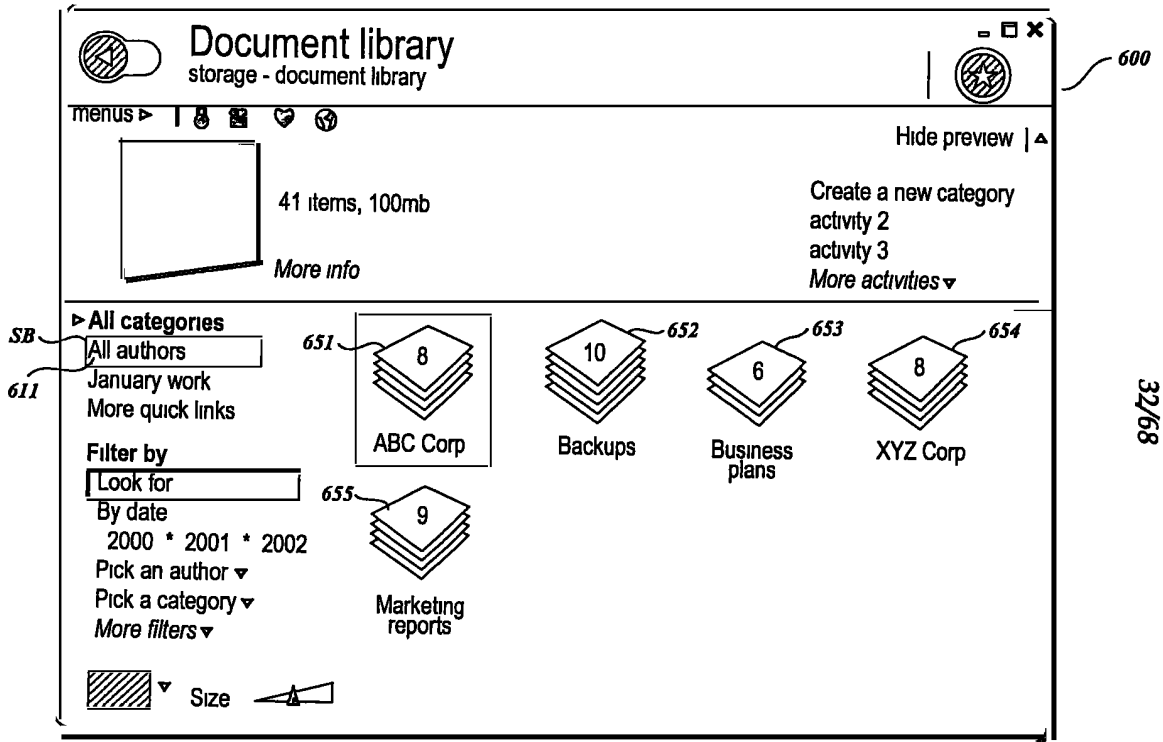


Fig.32.

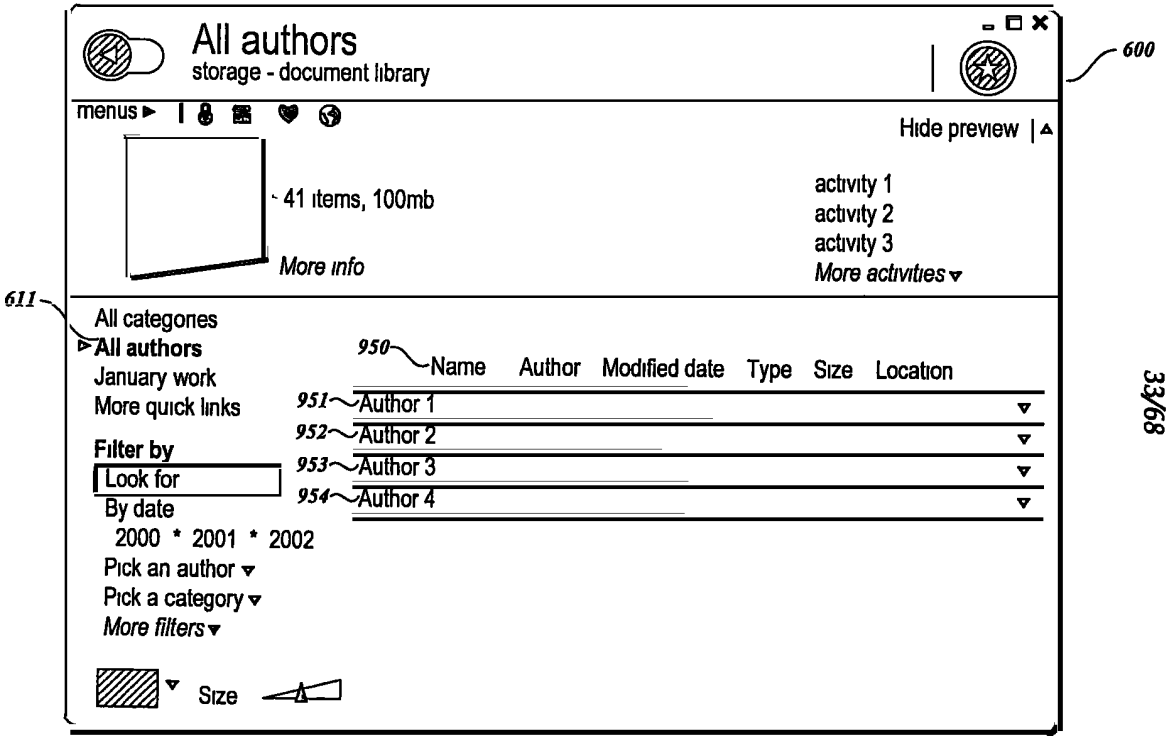


Fig.33.

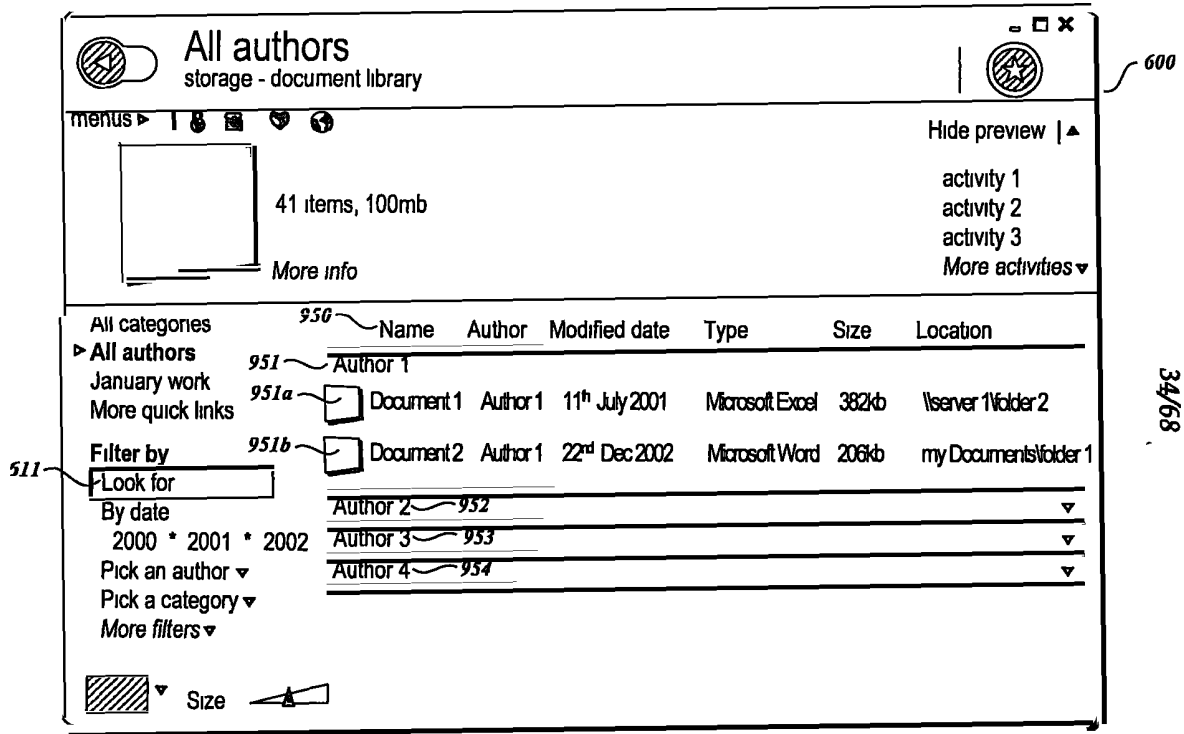


Fig.34.

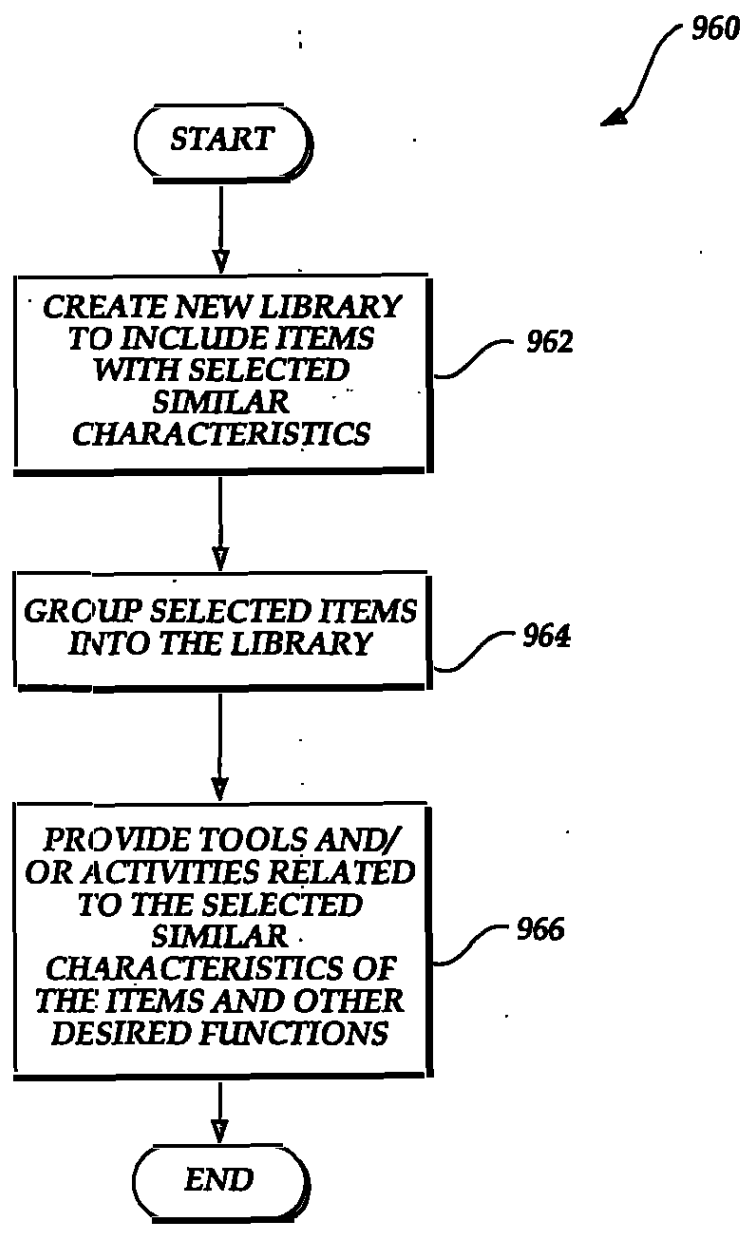


Fig.35.

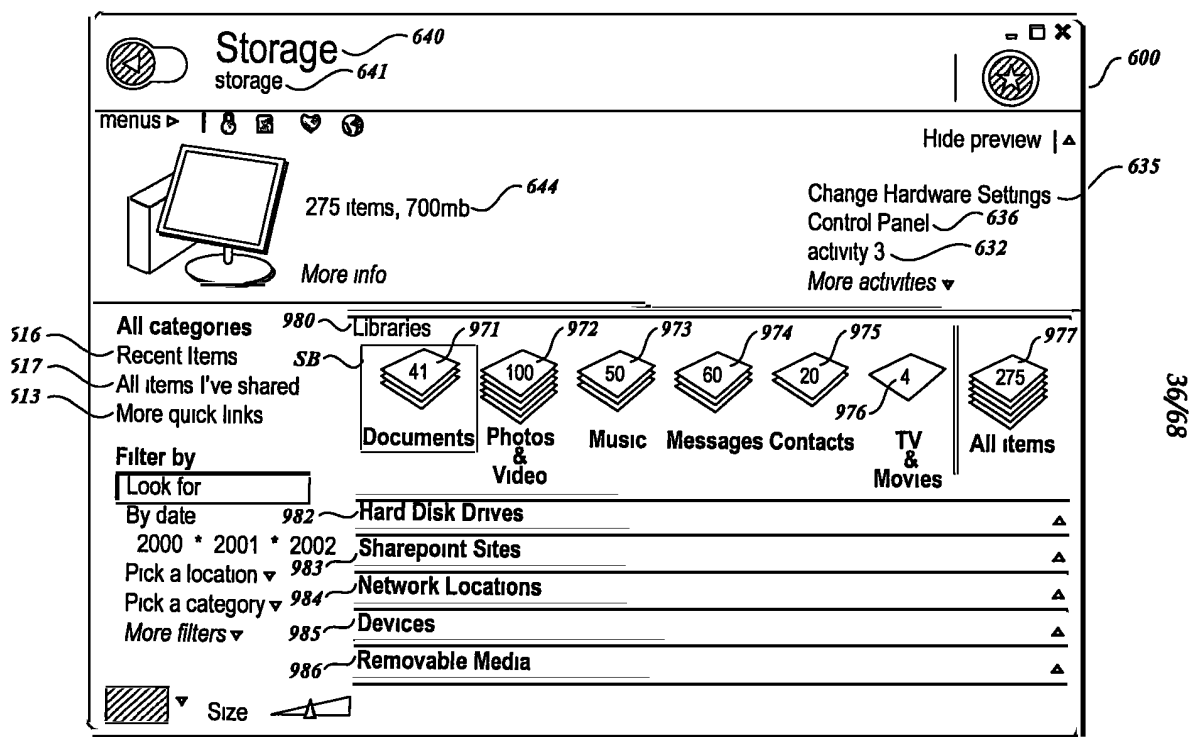


Fig.36.

374

37/68

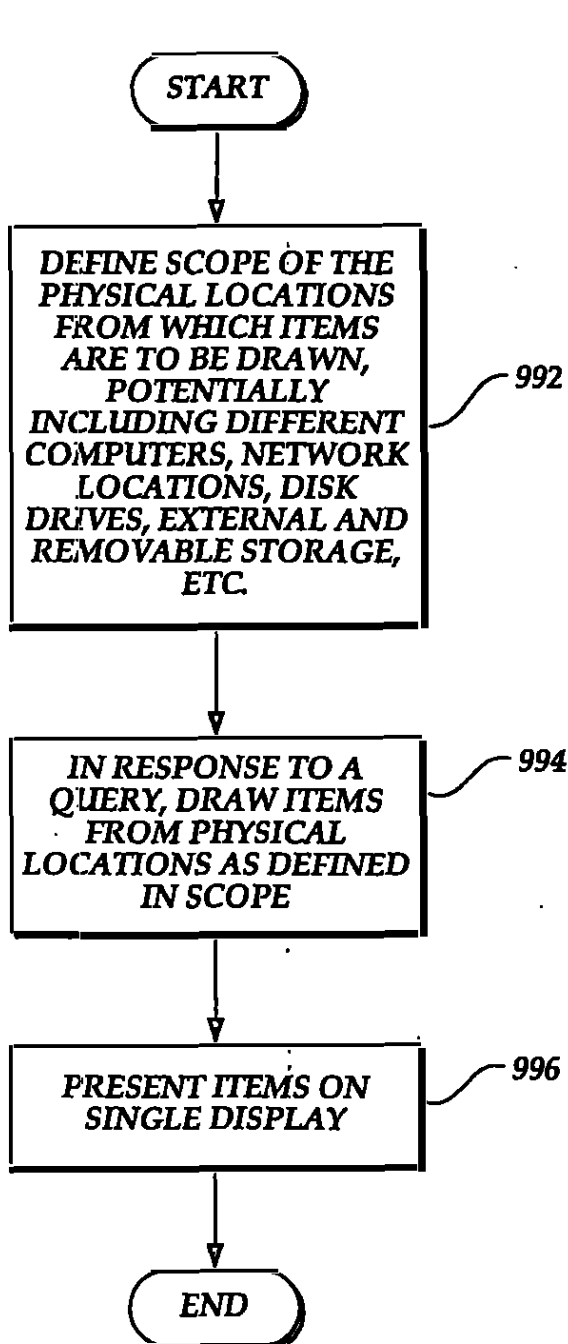


Fig.37.

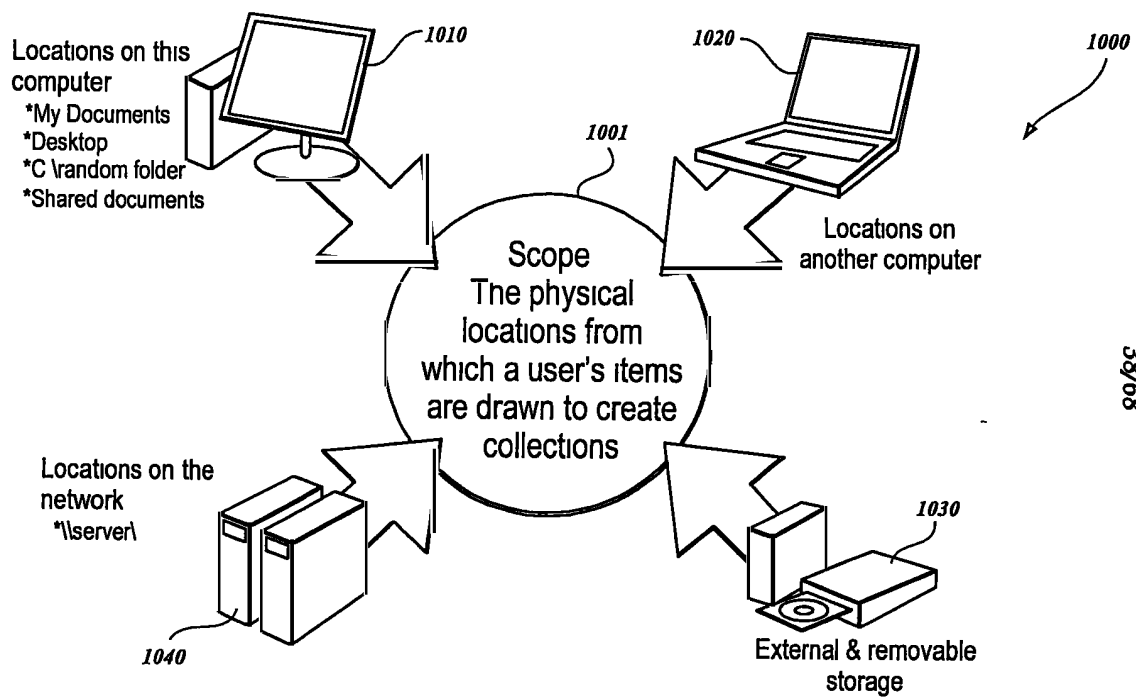


Fig.38.

39/68

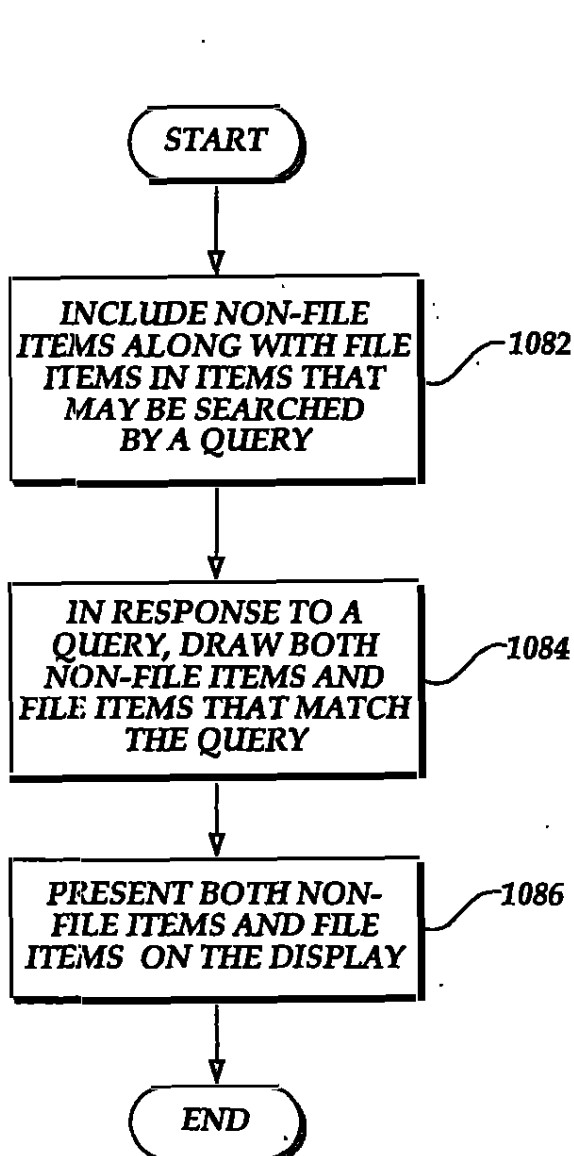


Fig.39.

378

378

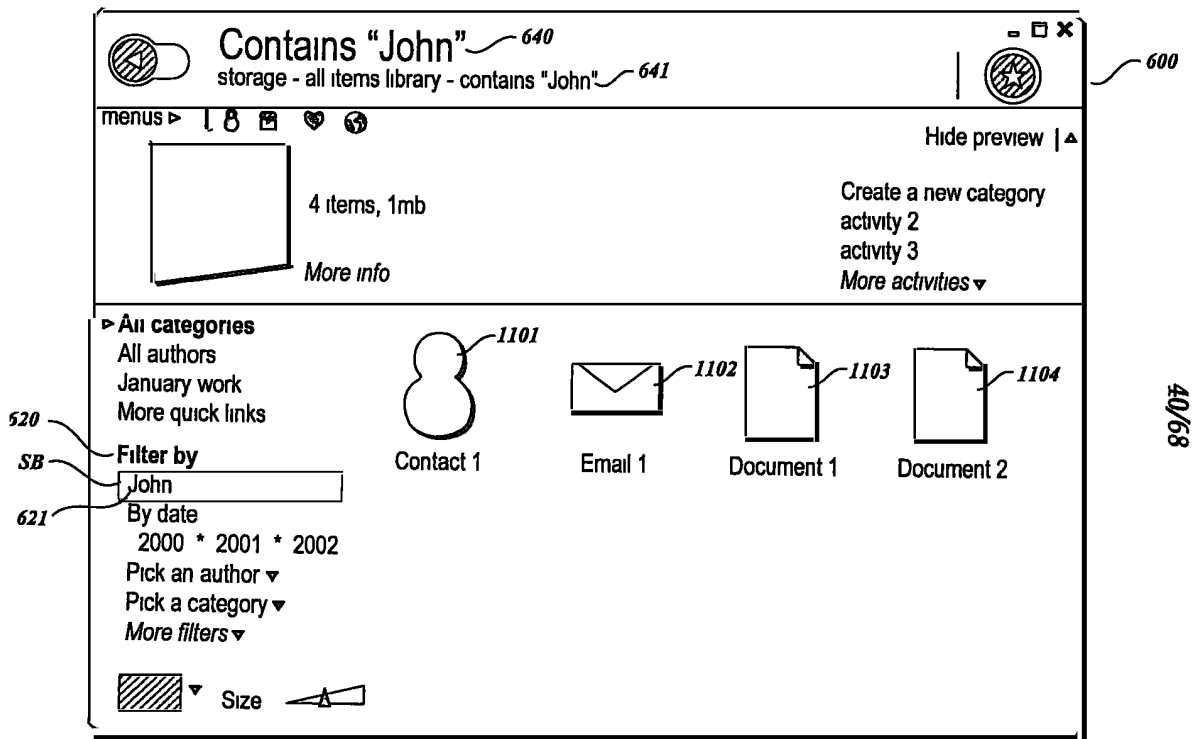
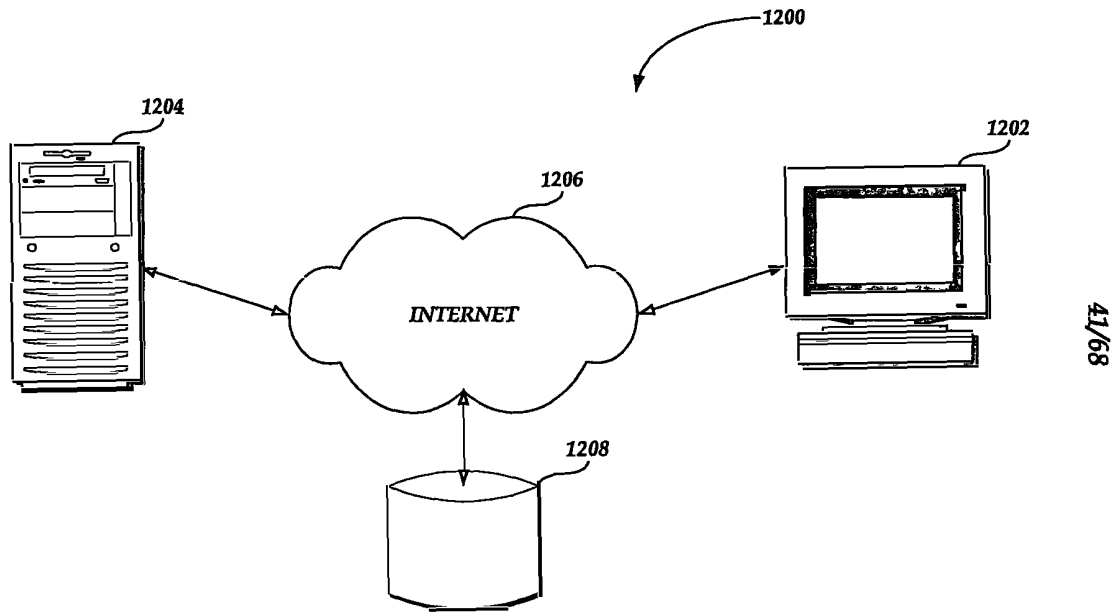


Fig.40.

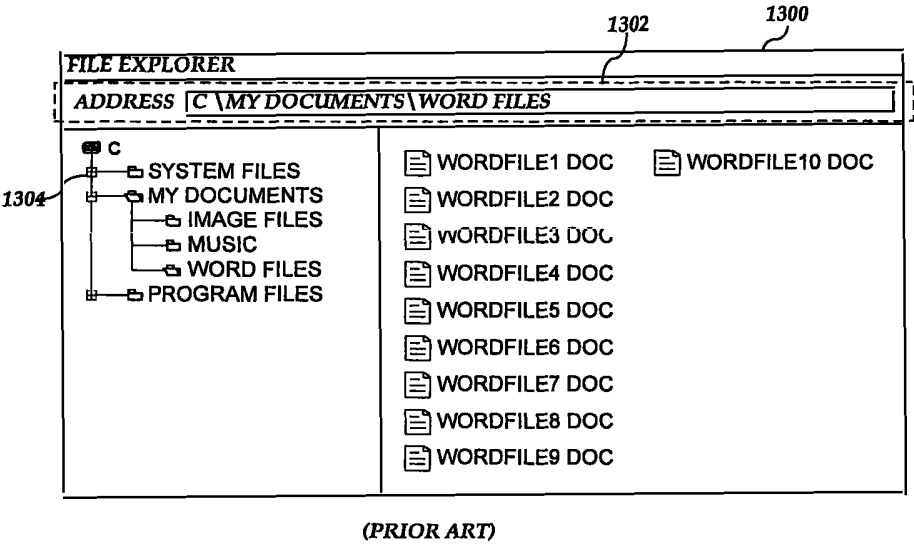
378



41/68

Fig.41.

379

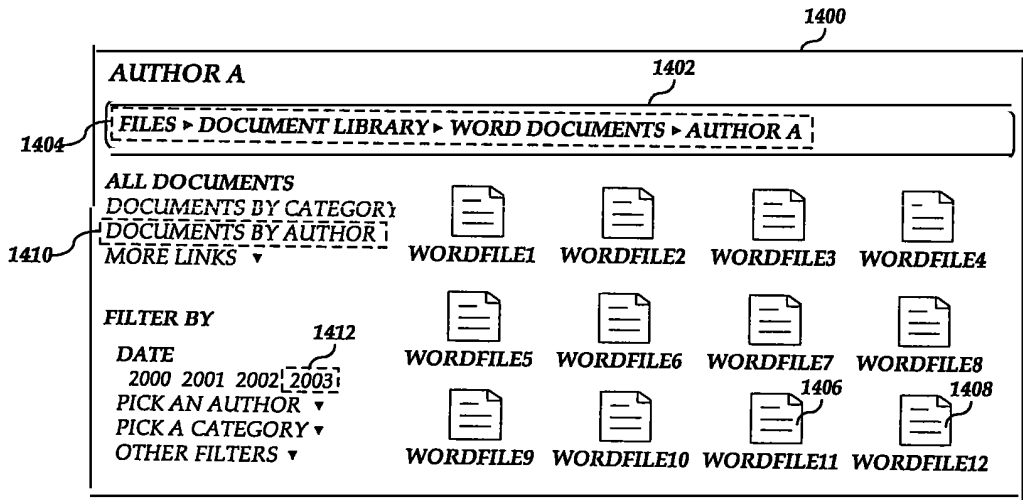


42/68

(PRIOR ART)

Fig.42.

880



43/68

Fig.43.

887

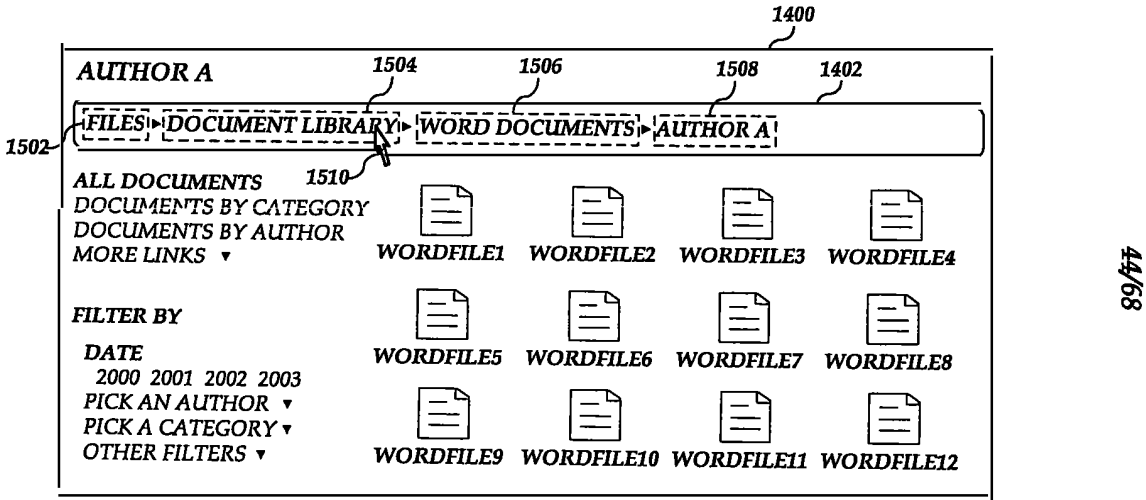


Fig.44A.

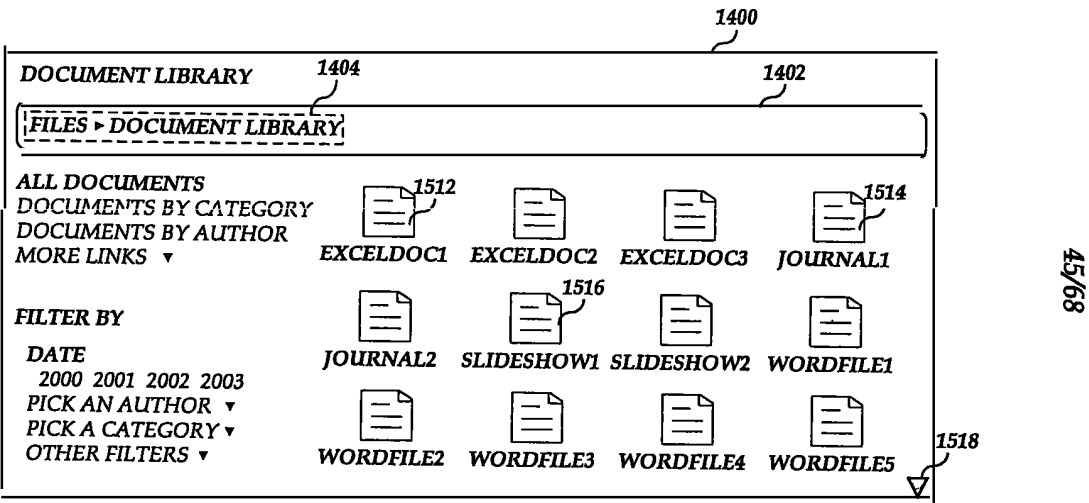


Fig.44B.

46/68

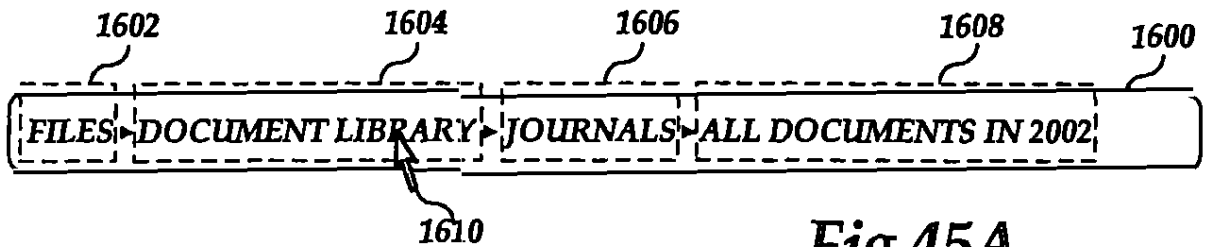


Fig. 45A.

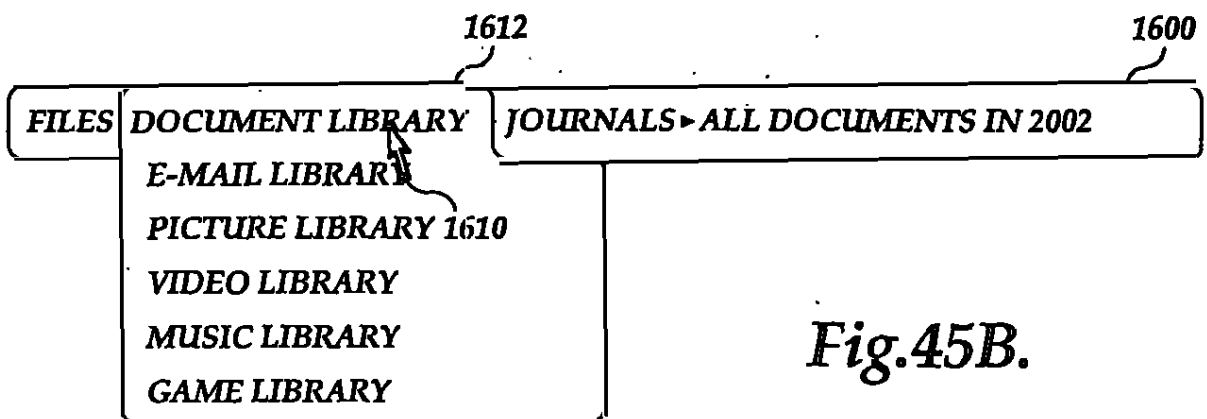


Fig. 45B.

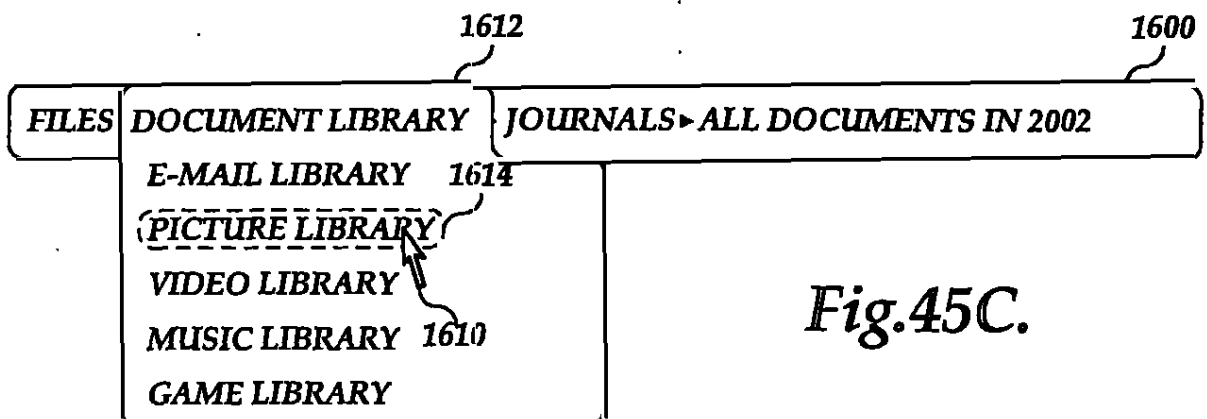


Fig. 45C.

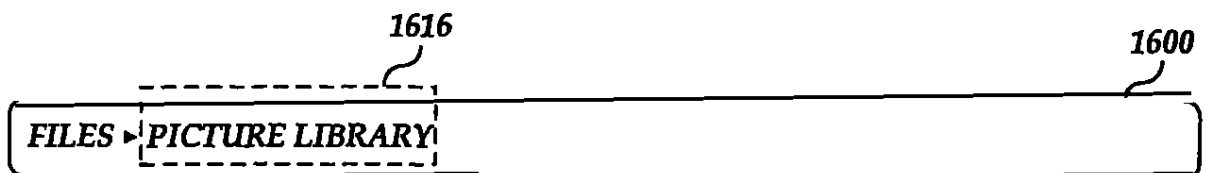


Fig. 45D.

47/68

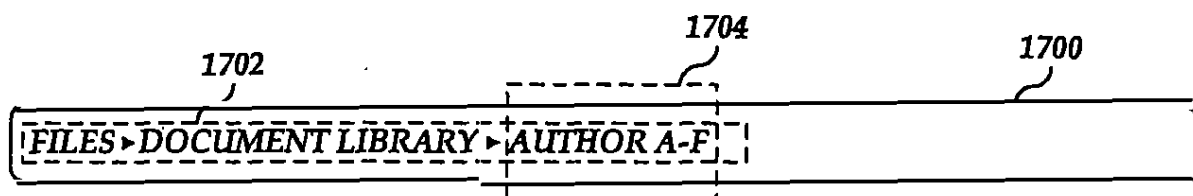


Fig.46A.

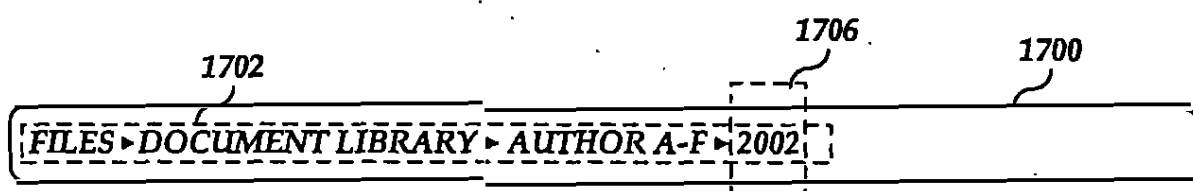


Fig.46B.

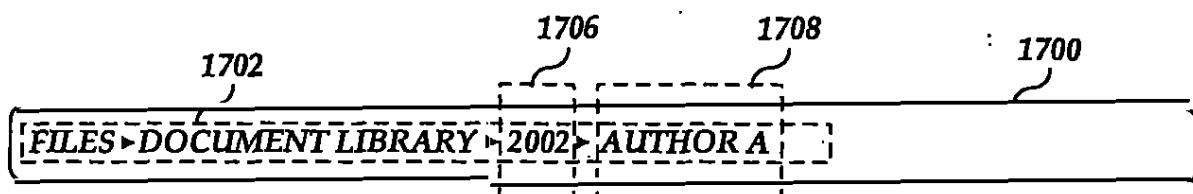


Fig.46C.

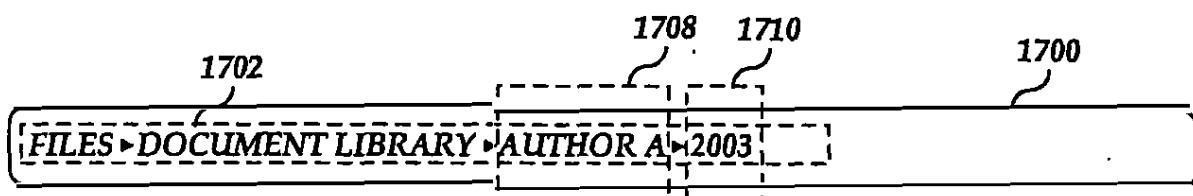


Fig.46D.

48/68



Fig.47A.

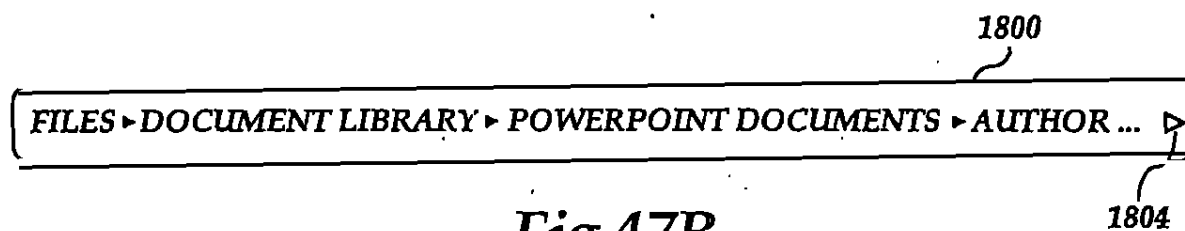


Fig.47B.

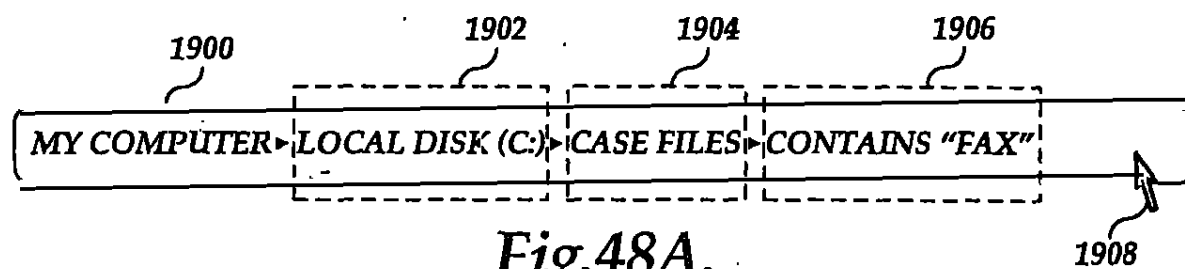


Fig.48A.

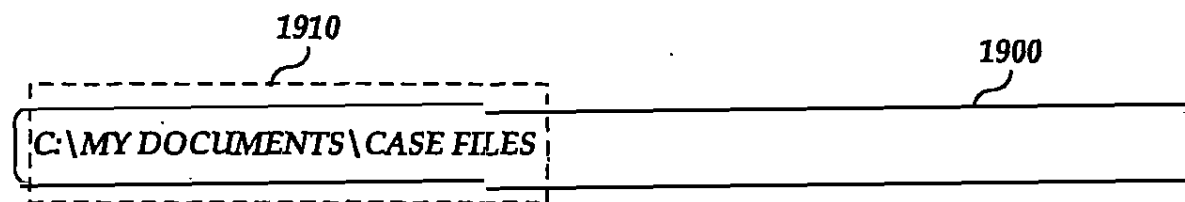
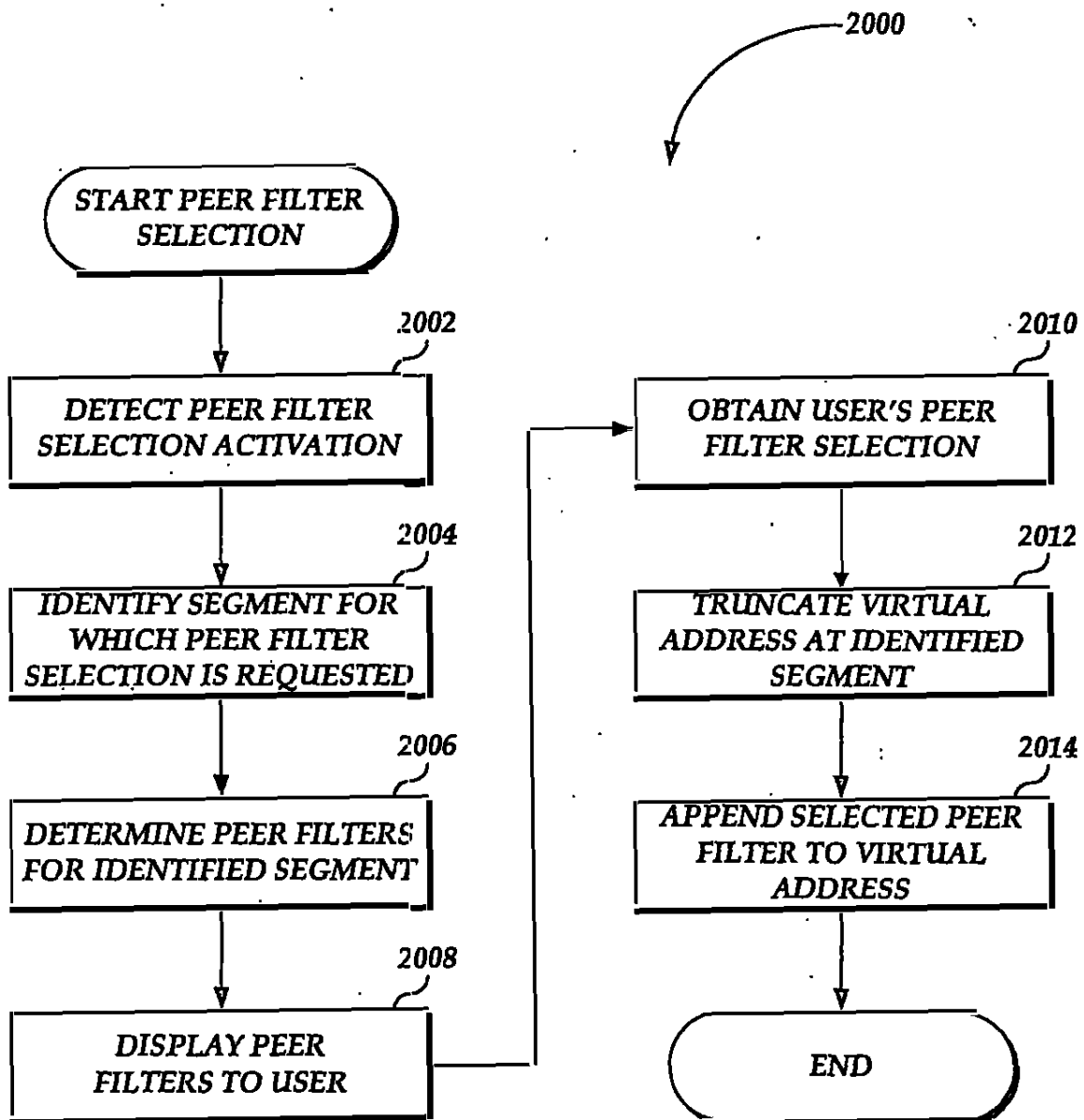


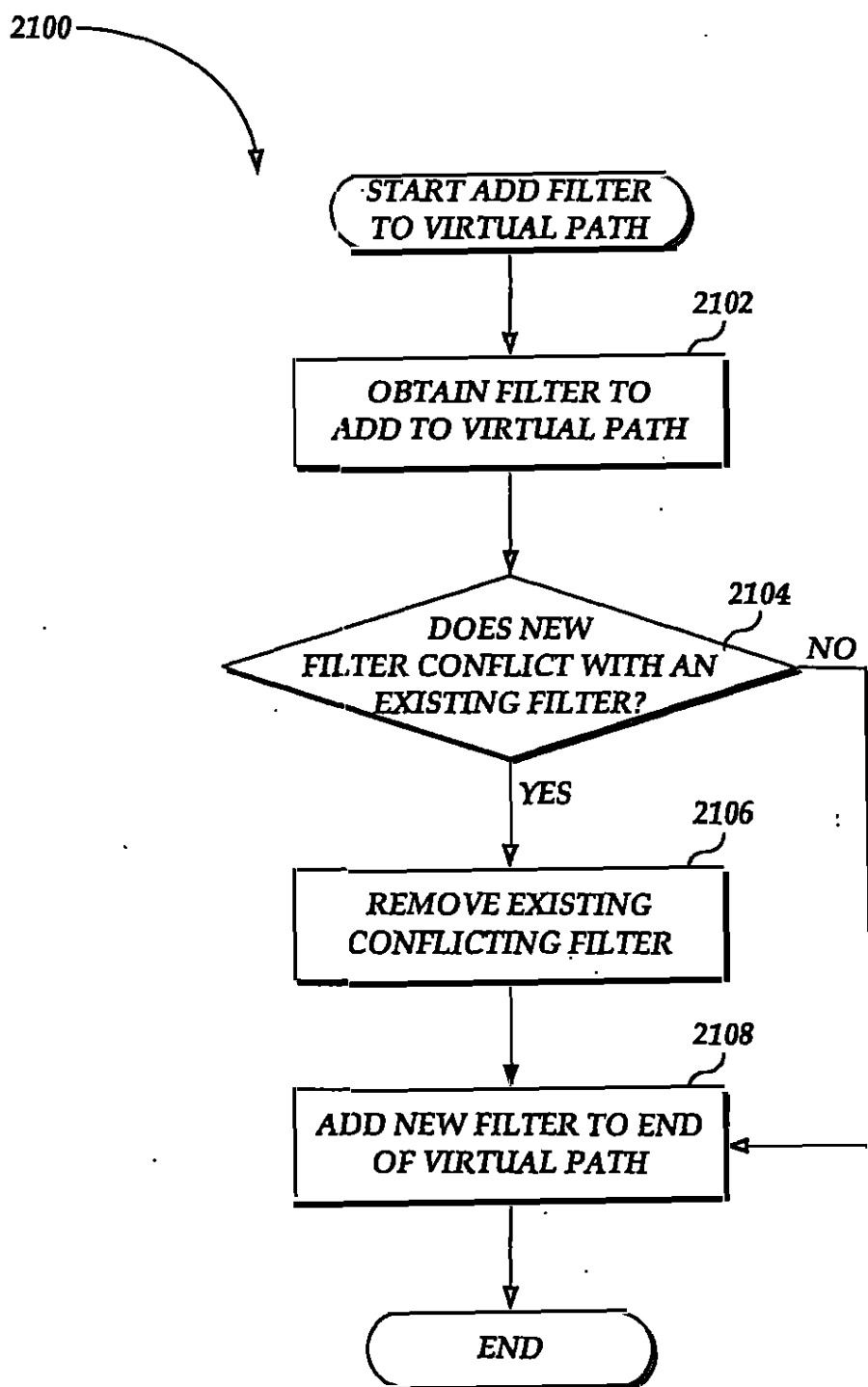
Fig.48B.

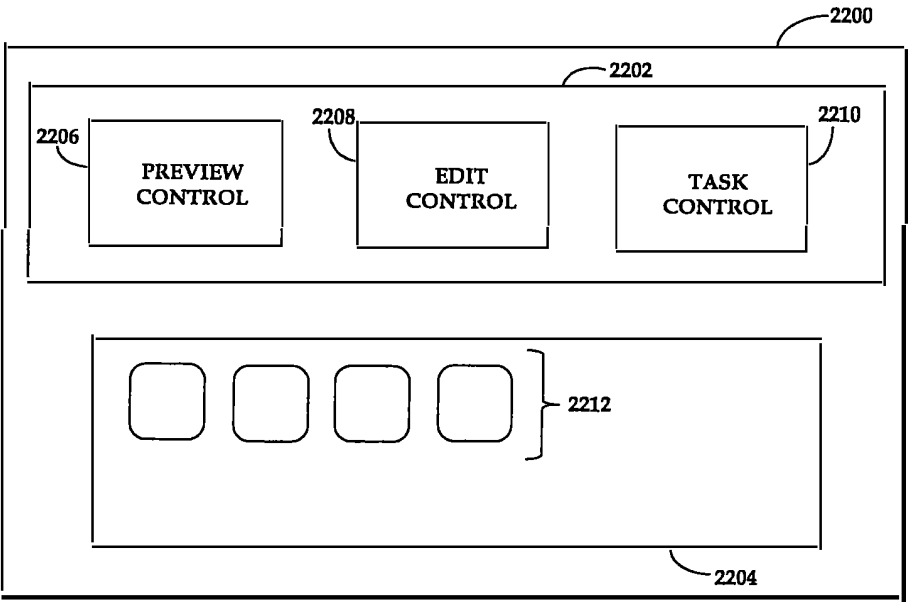


0.65

388

50/68

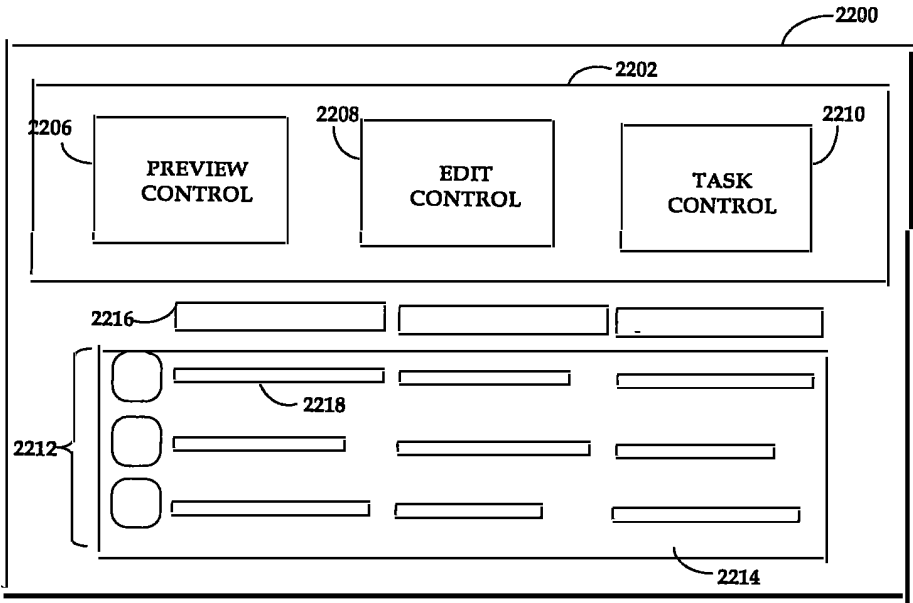




51/68

Fig.51A.

289



52/68

Fig.51B.

298

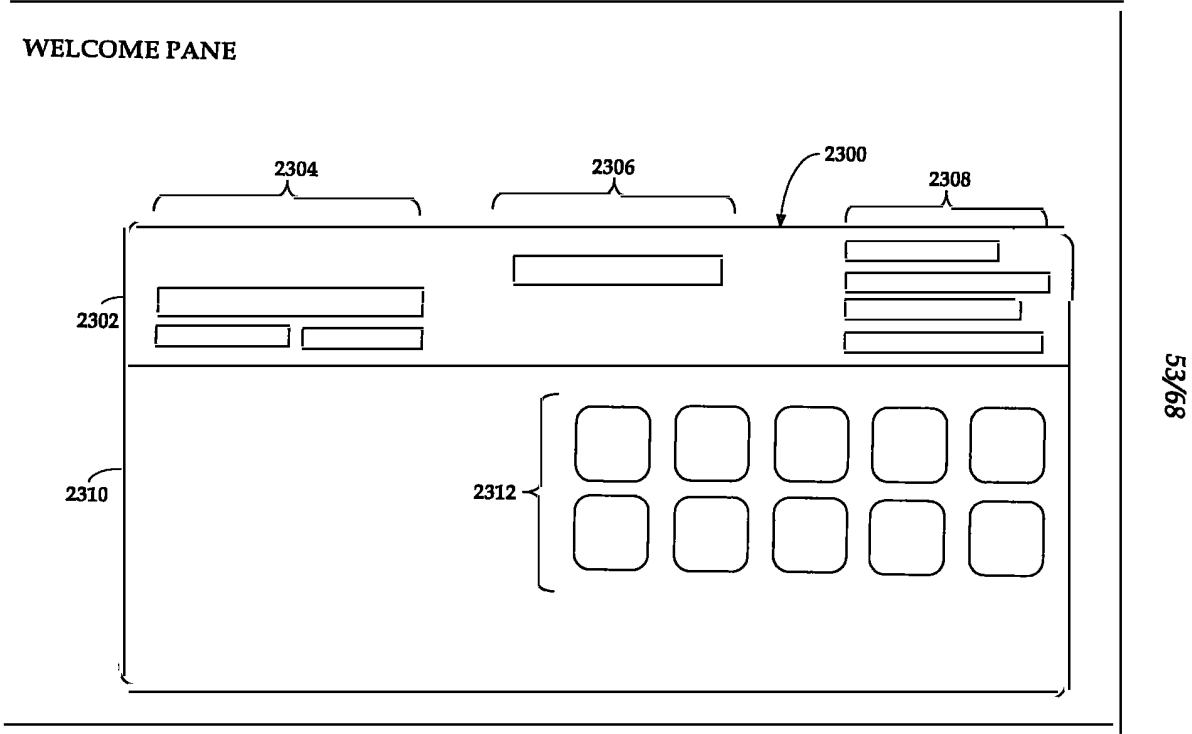


Fig.52.

391

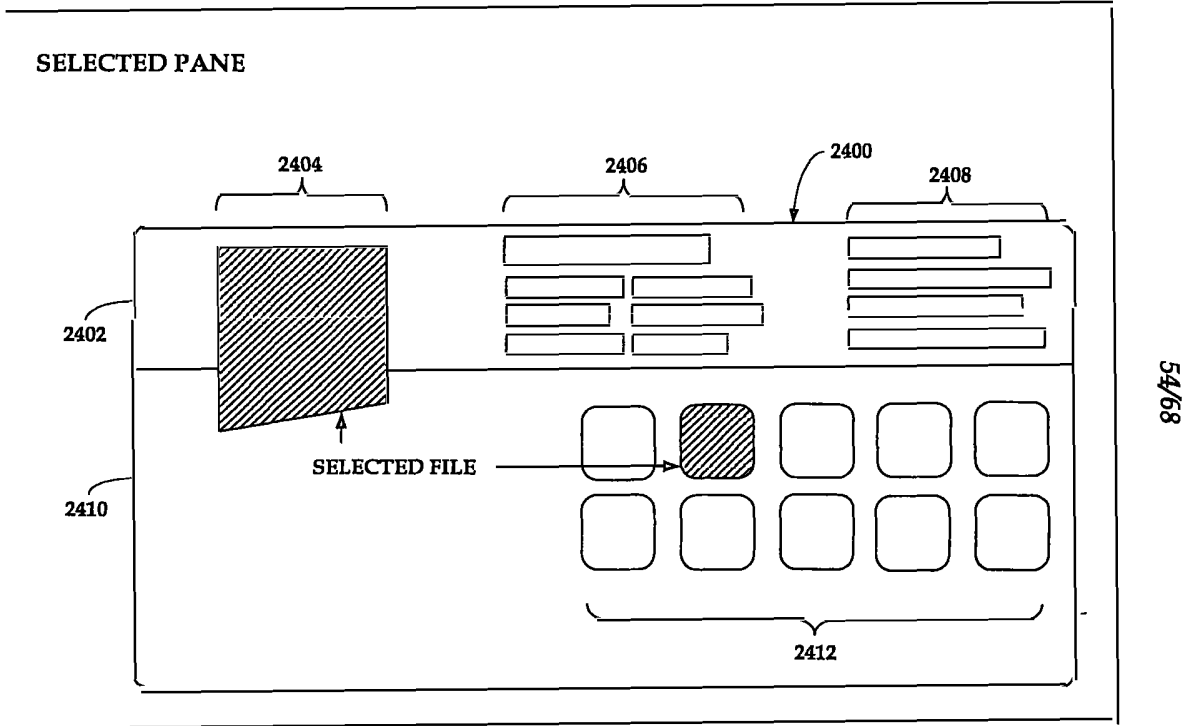


Fig.53.

28/7/87

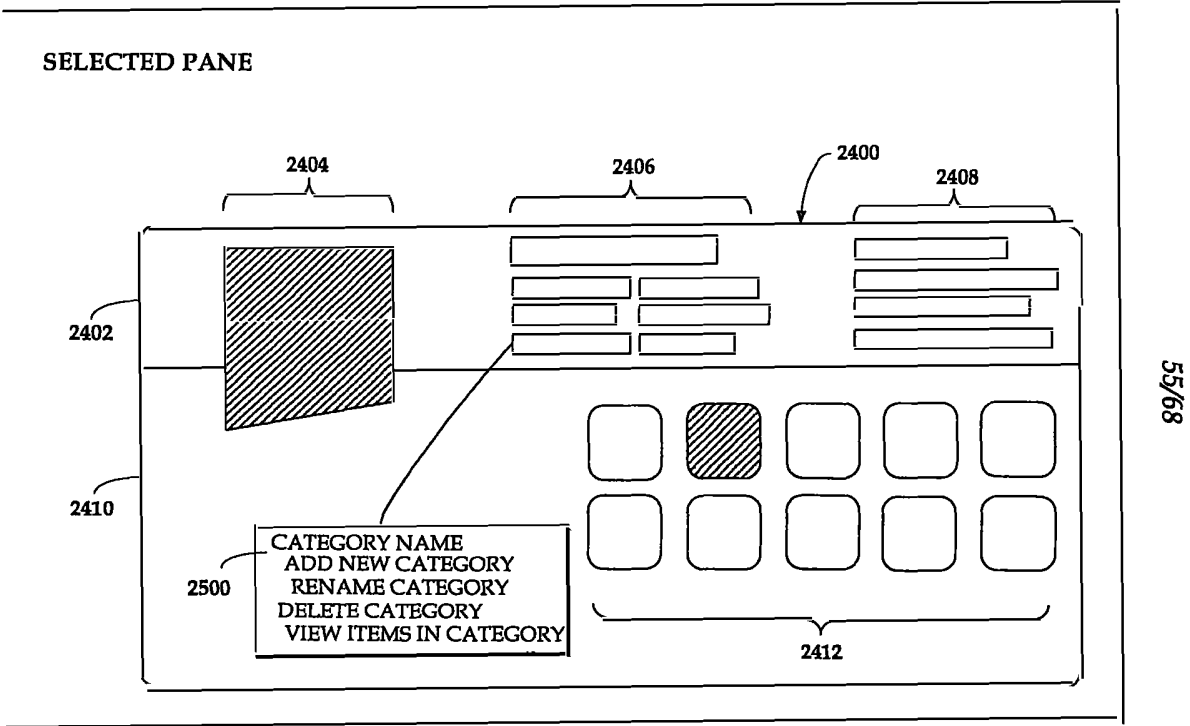
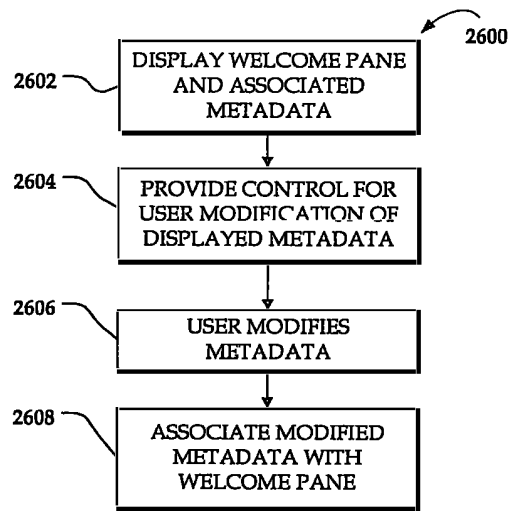


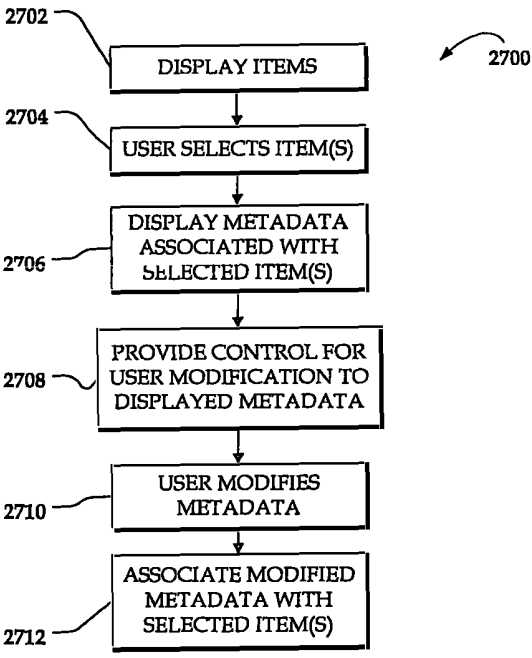
Fig.54.



56/68

Fig.55.

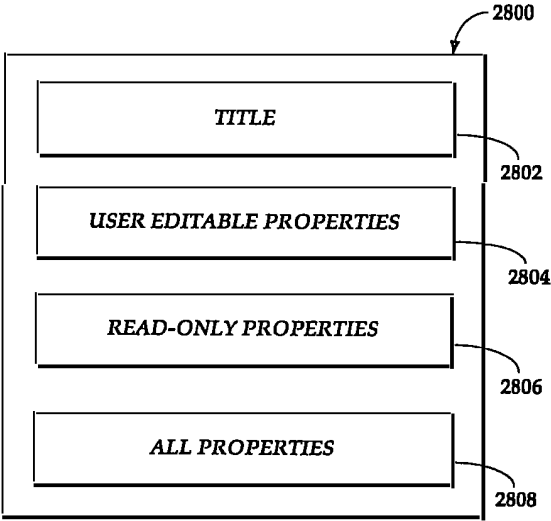
394



57/68

Fig.56.

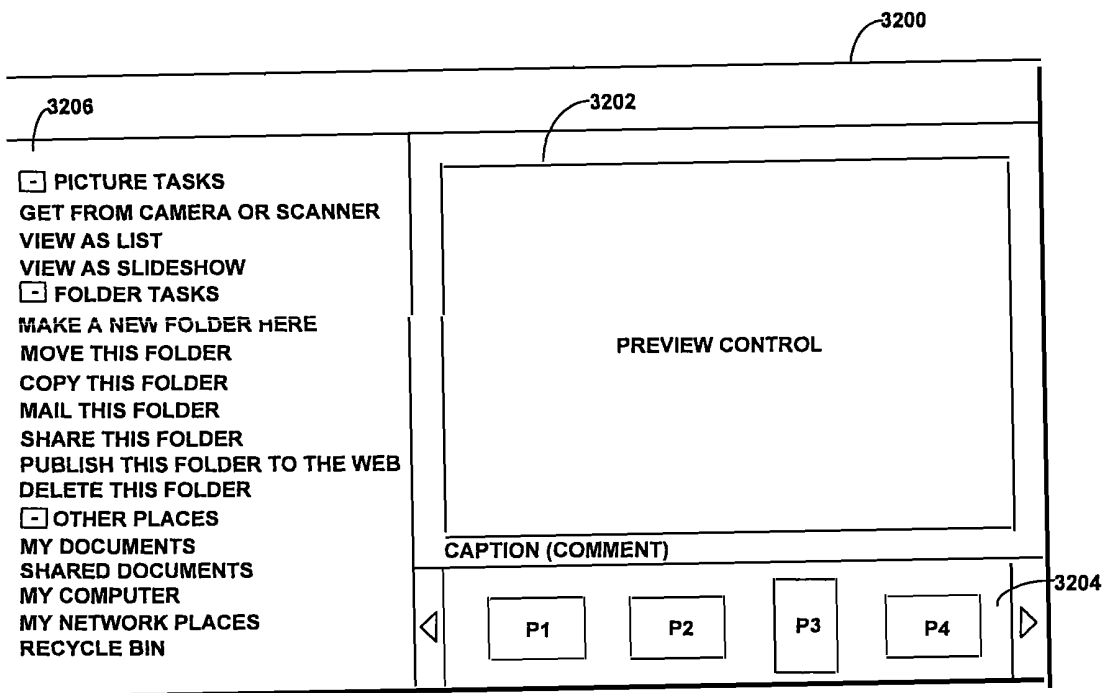
295



58/68

FIG.57.

286

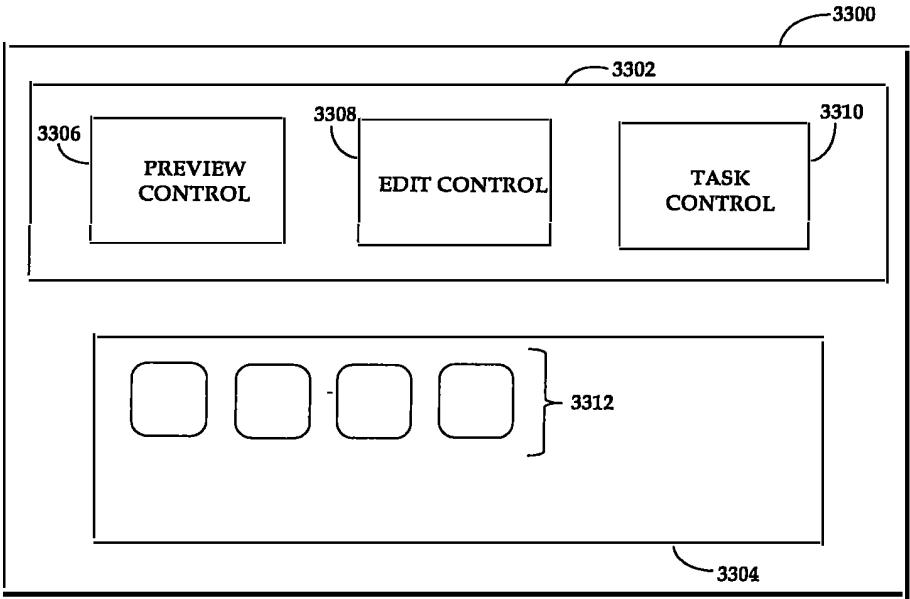


59/68

(PRIOR ART)

Fig.58.

397



89/09

Fig.59.

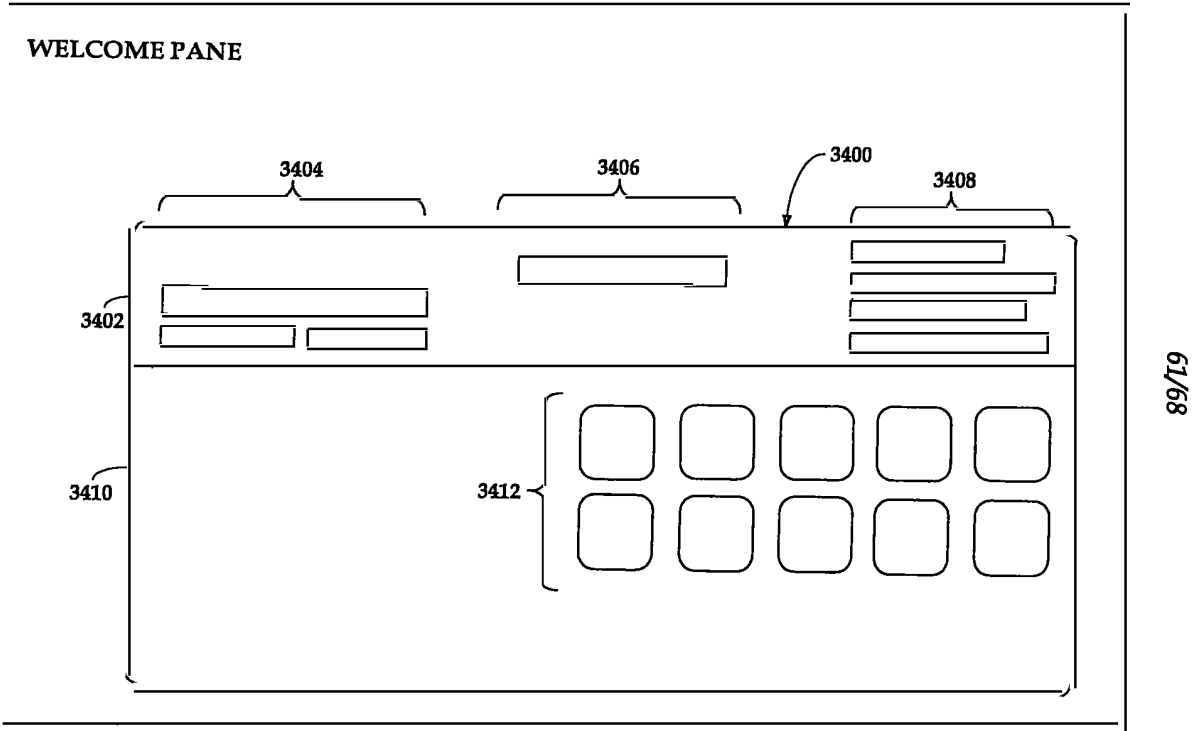


Fig.60.

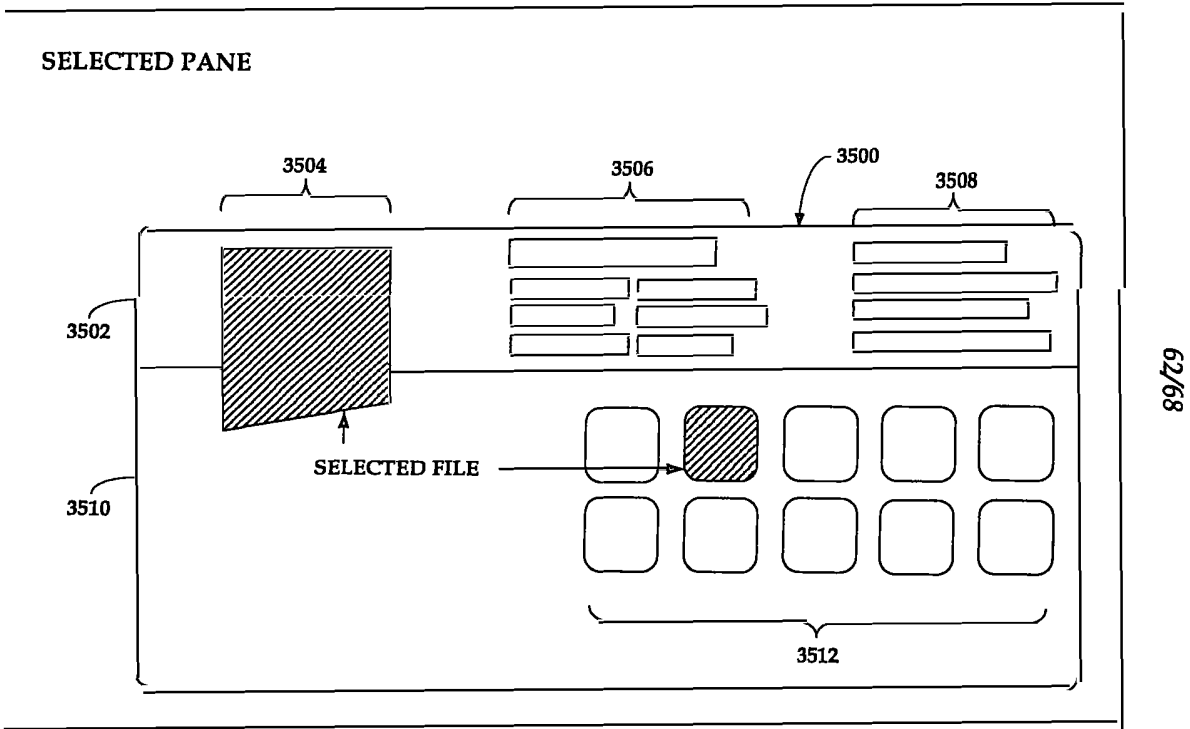


Fig.61.

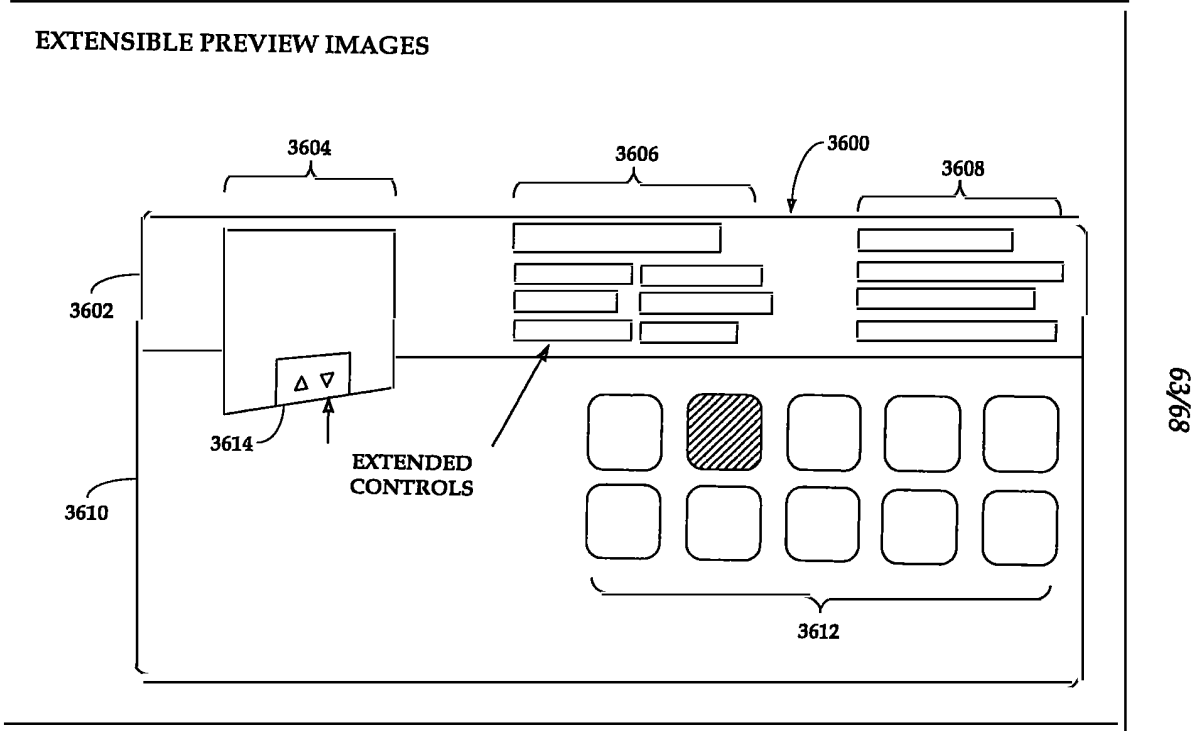


Fig.62.

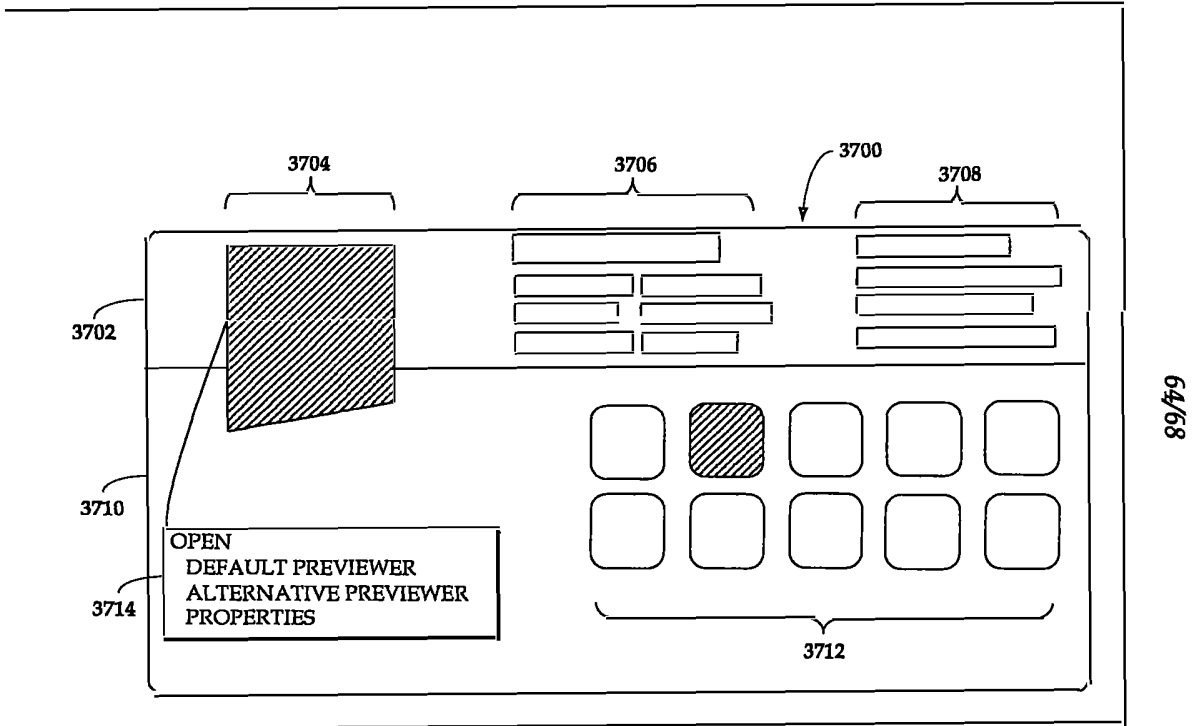
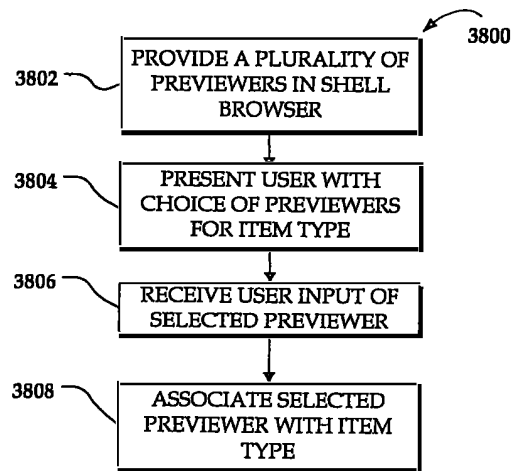


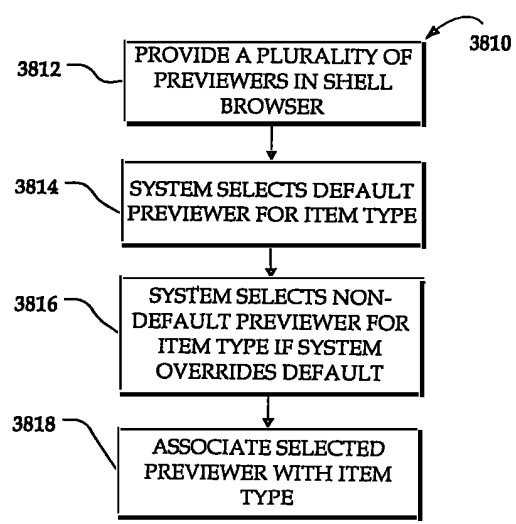
Fig.63.



65/68

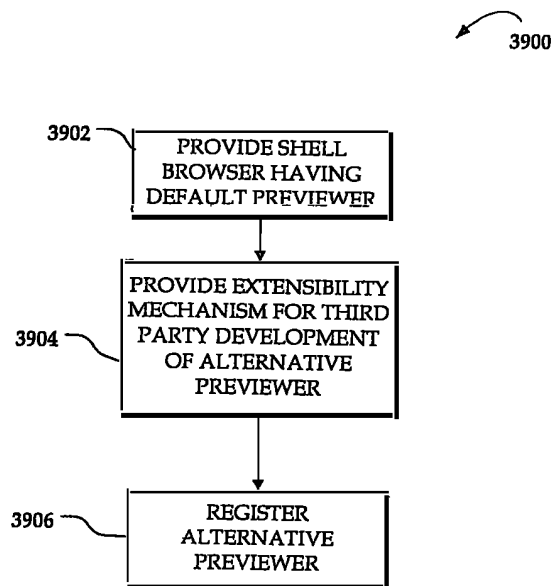
Fig.64A.

403



89/99

Fig.64B.



67/68

Fig.65.

407

SUPERINDUSTRIA Y COMERCIO
Radicacion : 04061817 00000000 Folios: 400
Fecha (AMD): 2004-06-29 16:59:56
Tramite : 011 PCT-PATENT 378 FASE NNCI 411 PRESENTAC
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 04 - 50,356
FECHA : JUNIO 25 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	28716291	25/06/2004	5,439,000.00

***** CONCEPTO *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01	SOLICITUDES	
		1 TRAMITES DE SOL. DE PATENTE DE IN	422,000.00
		TOTAL :	\$ 422,000.00

SON: CUATROCIENTOS VEINTIDOS MIL PESOS

RESPONSABLE : _____

CARDENAS Y CARDENAS
ABOGADOS

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____
"SHELL" DEL SISTEMA DE ARCHIVOS

3

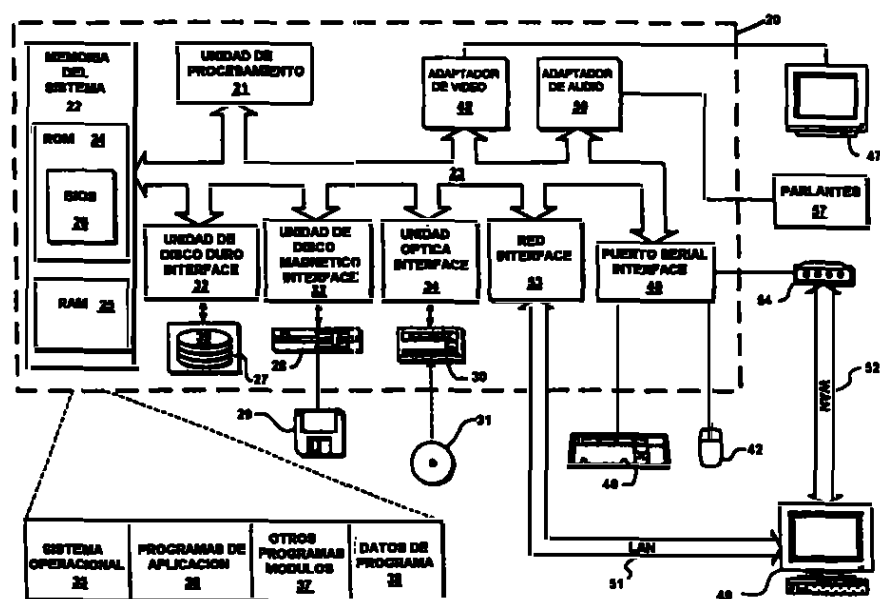


FIGURA 1

409

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

	USUARIO	RADICADOR
PATENTE DE INVENCION ☒	()	(☒)
- 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 o 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:



Fecha:

2020
Bogotá, D.C.,

Doctor:
DARIO CARDENAS NAVAS
Apoderado
MICROSOFT CORPORATION

Oficio N° 08842

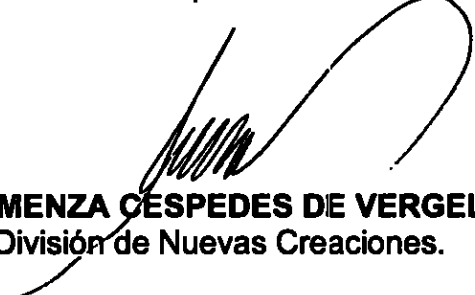
ASUNTO:	Radicación N°	04-61817
	Solicitud internacional	PCT/US03/15610
	Fecha inicio fase nacional	16/12/2005
	Trámite	011
	Evento	379
	Actuación	416
	Folios	001

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, artículo 27 del Tratado de Cooperación en Materia de Patentes (PCT), regla 51 bis del Reglamento y Circular Única de la Superintendencia de Industria y Comercio, se le comunica que:

☒ La solicitud no contiene la copia del documento en que consta la cesión del derecho a la patente del inventor al solicitante o a su causante. (Art. 26-k) Decisión 486).

En cumplimiento de lo dispuesto por el artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos 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 Única N° 10 de 2001.


ALIX CARMENZA CÉSPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones.

El anterior requerimiento fue notificado por fijación en lista de fecha, 24 SET 2004.

202047

Sede Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Conmutador: 3 820840 Fax: 3 505220
Sede CAN: Tr. 40A No. 38-50 3153265 al 69
Fax: 3 505220. Línea 9800-910 165

