



Industria y Comercio

SUPERINTENDENCIA

Delegatura de propiedad Industrial

División de Nuevas Creaciones

SOLICITUD

PATENTE DE INVENCION

21. EXPEDIENTE No.

06-019553

54. TÍTULO MAPA DE UN MODELO DE SISTEMA DE ARCHIVO

CON OBJETO DE UNA BASE DE DATOS

51. CLASIFICACIÓN INTERNACIONAL G06F 17/30

71. SOLICITANTE MICROSOFT CORPORATION

DOMICILIO REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA.

74. APODERADO DAZIO CARDENAS NAUAS

22. BOGOTÁ, D. C., FEBRERO 17, 2006

PETITORIO

SUPERINDUSTRIA Y COMERCIO

Radicación: 06019553 00000000

Folios: 149

Fecha (AMB): 2006-02-28 11:19:59

Trámite: 002 PATENTES 0

1 REGISTRO/ 411 PRESENTAR

Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

Industria y Comercio
SUPERINTENDENCIA

FORMULARIO UNICO DE SOLICITUD DE PATENTE 2000-3

1

SOLICITUD DE:

☒

Patente de Invención

☐

Patente de Modelo de Utilidad

SOLICITANTE
(71)

Nombre: MICROSOFT CORPORATION

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

Teléfono: 3123600 Fax: 3122410

E-Mail: general@cardenasycardenas.com

Lugar de Constitución: Estado de Washington, Estados Unidos de
America

Domicilio: REDMOND, WASHINGTON, ESTADOS UNIDOS
DE AMERICA.

IDENTIFICACION

C.C.

☐

NIT

☐

C.E.

☐

Otro

☒

Número

REPRESENTANTE
O APODERADO
(74)

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

☐

Número 17.066.629 BTA

TP 1.250 C.S.J.

INVENTOR(ES)
(72)

Nombre:

1. Amit Shukla
2. Anil Kumar Noh
3. Bekim Demiroski
4. Gregory S. Friedman
5. Jason T. Hunter
6. Jeffrey T. Pearce
7. Michael J. Newman
8. Nigel R. Ellis
9. Srinivasamurthy P. Acharya

Dirección:

1. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
2. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
3. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
4. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
5. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
6. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
7. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
8. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América
9. One Microsoft Way Redmond, Washington 98052,
Estados Unidos de América

Teléfono: 3123600 Fax: 3123600

E-Mail: general@cardenasycardenas.com

Domicilio:

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

C.C.

☐

NIT

☐

C.E.

☐

Otro

☒

Título(54):

MAPEO DE UN MODELO DE SISTEMA DE ARCHIVO A UN OBJETO DE UNA BASE DE DATOS

(6) Prioridad	(33) País de Origen	(32) Fecha	(31) N° de Solicitud
	USA	Marzo 28, 2005	11/091,079

SI ☒ NO ☐

Para publicar a partir de la fecha de la presente solicitud a los:

☐ 6 meses ☐ 12 meses ☐ 18 meses ☐ Otro

Comprobante de Pago No.:

06-13,595 / 06-13,596

Fecha

Febrero 17, 2006

ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud. (06 - 13,595)
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☒ Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (06 - 13,596)
- ☒ Poderes, si fuere el caso. (Protocolo No 16.069)
- ☒ Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- ☐ Copia certificada de la primera solicitud, si se reivindica prioridad.
- ☒ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☒ Descripción de la invención.
- ☒ Una o más reivindicaciones.
- ☒ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☒ Arte final 12x12 y 6x6 cm.
- ☒ Resumen y extracto para publicación.

10

NOMBRE:

DARIO CARDENAS NAVAS

FIRMA:



C.C. 17.066.629 BTA

TP. 1.250

C.S.J.

LH

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

MIT : 800176089-2

SUPERINDUSTRIA Y COMERCIO

Radicacion : 06019553 00000000 Folios: 149
Fecha (RD): 2006-02-26 11:19:58
Tramite : 002 PATENTES / 1 REGISTRO/ 411 PRESENTA
Dependencia: 8020 DIVISION DE NUEVAS CREACIONES

RECIBO OFICIAL DE CAJA : 06 - 13,595
FECHA : FEBRERO 17 DE 2006

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I. CONSIGNACION		BANCO POPULAR	050-00110-6	47523610	17/02/2006	6,151,000.00

***** CONCEPTO *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	1 TRAMITES DE SOL. DE PATENTE DE IN	463,000.00
		TOTAL :	\$ 463,000.00

SON: CUATROCIENTOS SESENTA Y TRES MIL PESOS

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

MAPEO DE UN MODELO DE SISTEMA DE ARREGLOS
A UN OBJETO DE UNA BASE DE DATOS

ABOGADOS

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

NIT : 800176089-2

SUPERINDUSTRIA Y COMERCIO

Radicacion : 06019363 00000000

Folios: 149

Fecha (AMB): 2006-02-20 11:19:59

Tramite : 002 PATENTES 0 1 REGISTRADO/ 411 PRESENTAC

Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

RECIBO OFICIAL DE CAJA : 06 - 13,596
FECHA : FEBRERO 17 DE 2006

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I. CONSIGNACION		BANCO POPULAR	050-00110-6	47523610	17/02/2006	6,151,000.00

***** CONCEPTO *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	82 INVOCACION DE UNA PRIORIDAD EN NU	130,000.00
		TOTAL :	\$ 130,000.00

SON: CIENTO TREINTA MIL PESOS

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

MAPEO DE UN MODELO DE SISTEMA
A UN OBJETO DE UNA BASE DE DATOS

CARDENAS & CARDENAS
ABOGADOS

+

LOGO DEL DEPARTAMENTO DE COMERCIO

DEPARTAMENTO DE COMERCIO DE ESTADOS UNIDOS
Oficina de Patentes y Marcas Registradas
Asistente del Secretario y el Comisionado de Patentes y Marcas Registradas
Washington, D.C. 20231

*** 500046958A ***

*** 500046958A ***

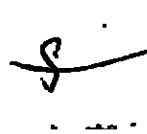
Agosto 25 de 2005 PTAS

Himanshu S. Amin
24th Floor, 1900 E. 9th Street
Cleveland, OH 44114

Oficina de Patentes y Marcas Registradas
Notificación de Registro de Documento de Cesión

El documento anexo fue registrado por la División de Cesiones de la Oficina de Patentes y Marcas Registradas. Una copia completa en microfilm se encuentra disponible en la sala de búsqueda de cesiones en el carrito y el número de pantalla indicado a continuación.

Por favor revise toda la información de esta notificación. La información en esta notificación de registro refleja la información presente en el sistema de cesiones de patentes y marcas registradas. En caso de encontrar errores o tener inquietudes con respecto a esta notificación, puede contactar al empleado cuyo nombre aparece en la misma al 703-308-9723. Por favor enviar la solicitud de



corrección a: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP:
ASSIGNMENT SERVICES DIVISION, P.O. BOX 1450, ALEXANDRIA, VA 22313.

Fecha de Registro: 8/25/05

Carrete/Pantalla: 016451/0719 -

No. de Páginas: 4

Resumen: Cesión de los Intereses del Cedente (Referirse al documento para más
detalles)

Cedente:

Ellis, Nigel R.

Fecha de documento: 03/24/2005

Cedente:

Friedman, Gregory S.

Fecha de documento: 03/24/2005

Cedente:

Demiroski, Bekim

Fecha de documento: 03/24/2005

Cedente:

Nori, Anil Kumar

Fecha de documento: 03/24/2005

Cedente:

Shukla, Amit

Fecha de documento: 03/24/2005

Cedente:

Acharya, Srinivasmurthy P.

Fecha de documento: 03/24/2005

SEGUNDA PÁGINA: RightFax; 8/25/05 8:47 Página 003/005 Fax Server
016451/0719 Página 2

6

Cedente:

Pearce, Jeffrey T.

Fecha de documento: 03/24/2005

Cedente:

Newman, Michael J.

Fecha de documento: 03/24/2005

Cedente:

Hunter, Jason T.

Fecha de documento: 03/24/2005

Cesionario:

MICROSOFT CORPORATION

ONE MICROSOFT WAY

REDMOND, WASHINGTON 98052

Serial: 11091079

Fecha de Solicitud: 03/28/2005

No. de Patente:

Fecha de Emisión:

**Título: MAPEO DE UN MODELO DE SISTEMA DE ARCHIVO A UN OBJETO DE
UNA BASE DE DATOS**

Dorothy Riley

Auxiliar Legal

División de Cesiones

Oficina de Registro Público

**TERCERA PÁGINA: RightFax; 8/25/05 8:47 Página 004/005 Fax Server. En el
margen derecho: OP \$40.00 11091079**

CESIÓN DE LA PATENTE

Versión Electrónica v1.1

08/25/06

Versión Impresa v1.1

500046958

7

Tipo de Presentación:	Nueva Cesión
Característica de la Transferencia:	Cesión

Información de La Parte que Transfiere

Nombre	Fecha de Ejecución
Nigel R. Ellis	03/24/2005
Gregory S. Friedman	03/24/2005
Bekim Demiroski	03/24/2005
Anil Kumar Nori	03/24/2005
Amit Shukla	03/24/2005
Srinivasmurthy P. Acharya	03/24/2005
Jeffrey T. Pearce	03/24/2005
Michael J. Newman	03/24/2005
Jason T. Hunter	03/24/2005

Información de la Parte Receptora

Nombre:	MICROSOFT CORPORATION
Dirección:	One Microsoft Way
Ciudad:	Redmond
Estado/País:	WASHINGTON
Código Postal:	98052

Número Total de Propiedades: 1

Tipo de Propiedad:	Número
Número de Solicitud:	11091079

CORRESPONDENCIA

Número de telefax: (216) 696-8731

La correspondencia será enviada por correo de los Estados Unidos si el envío por

8

telefax falla

Teléfono: 216-696-8730
 Correo Electrónico: voleszewski@thepatentattorneys.com
 Nombre del Correspondiente: Himanshu S. Amin
 Renglón 1 de la Dirección: 24th Floor 1900 E. 9th Street
 Renglón 4 de la Dirección: Cleveland, OHIO 44114

CUARTA PÁGINA: RightFax; 8/25/05 8:47 Página 005/005 Fax Server. En el margen derecho

NOMBRE DE QUIEN ENVÍA:	Himanshu S. Amin
------------------------	------------------

Número de Anexos: 2 Fuente = MS311235.01Assignment#page1.tif Fuente = MS311235.01Assignment#page2.tif

QUINTA PÁGINA:

No. de Expediente de MS: MS311235.01

No. de Expediente de OC: MSFTP852US

CESIÓN DE PATENTE

NOSOTROS, Nigel R. Ellis, Gregory S. Friedman, Bekim Demiroski, Anil Kumar Nori, Amit Shukla, Srinivasmurthy P. Acharya, Jeffrey T. Pearce, Michael J. Newman, Jason T. Hunter ("CEDENTES") son los creadores de un asunto ("INVENTO") revelado y/o reivindicado en una solicitud de patente llamado MAPEO DE UN MODELO DE SISTEMA DE ARCHIVO A UN OBJETO DE UNA BASE DE DATOS ("SOLICITUD") la cual:

☒ Será registrada sin esta CESIÓN DE PATENTE ejecutada. Los CEDENTES por

este documento autoriza y solicita a los representantes legales del CESIONARIO, de Amin and Turocy, LLP, 1900 E. 9th Street, Cleveland, OHIO 44114, los cuales están asociados al número de cliente 27195, la inserción en este paréntesis (No. de serial U.S. 11/091,079, solicitado el 3/28/2005) del No. de Serie US de la SOLICITUD y su fecha de presentación, si esto es conocido;

_ Fue solicitada el _____ y le fue otorgada el No. de Serial U.S. _____;

_ Se presenta junto con estos documentos;

Microsoft Corporation, una corporación del estado de Washington, en su nombre y en el de sus sucesores y cesionarios ("CESIONARIO") tiene derecho e intenta adquirir la totalidad y la exclusividad de los derechos, títulos e intereses en el INVENTO y la SOLICITUD (y de cualesquiera otra solicitudes y patentes que de éstos se deriven, tales como solicitudes continuas en y para los Estados Unidos, sus territorios y en cualquiera y todos los países extranjeros ("DERIVADOS DE LA SOLICITUD"));

A cambio de contraprestaciones onerosas, el recibo de las cuales se reconocen en este documento por los CEDENTES, éstos por el presente venden, ceden y transfieren al CESIONARIO la totalidad y la exclusividad de los derechos, títulos e intereses en dicho INVENTO y en la SOLICITUD (y en los DERIVADOS DE LA SOLICITUD);

Los CEDENTES acuerdan ejecutar todos los instrumentos y documentos requeridos para elaborar y proseguir la SOLICITUD (y los DERIVADOS DE LA SOLICITUD) para fines de litigios relacionados con los mismos, o para proteger y perfeccionar dicho título sobre la SOLICITUD (y los DERIVADOS DE LA SOLICITUD).

03/24/2005

(Firma)

70

Fecha

Nigel R. Ellis

03/24/2005

(Firma)

Fecha

Gregory S. Friedman

03/24/2005

(Firma)

Fecha

Bekim Demiroski

SEXTA PÁGINA:

No. de Expediente de MS: MS311236.01

No. de Expediente de OC: MSFTP852US

03/24/2005

(Firma)

Fecha

Anil Kumar Nori

03/24/2005

(Firma)

Fecha

Amit Shukla

03/24/2005

(Firma)

Fecha

Srinivasmurthy P. Acharya

03/24/2005

(Firma)

Fecha

Jeffrey T. Pearce

03/24/2005

(Firma)

Fecha

Michael J. Newman

03/24/2005

(Firma)

Fecha

Jason T. Hunter

DOCUMENTO QUE CONSTA DE SEIS PÁGINAS.

PATENTE

MS311235.01/MSFTP852US

Certificado de Transmisión de Facsímile

Por el presente certifico que esta correspondencia (junto con cualquier documento referido como anexo o adjunto) se transmite por fax al 703-872-9306 en la fecha que aparece más abajo a Mail Stop Non-Fee Amendment (enmienda sin pago), Commissioner for Patents (Comisionado de Patentes), P.O.Box 1450, Alexandria, Virginia, 22313-1450

Fecha: (en manuscrito) 5-25-05

(firmado) Himanshu S. Amin

**EN LA OFICINA DE PATENTES Y MARCAS REGISTRADAS DE LOS
ESTADOS UNIDOS**

Con referencia a la solicitud de:

Solicitante(s): Nigel Ellis, et al.

Examinador: Desconocido

Nº de Serie: 11/091,079

Unidad de Arte: Desconocida

Fecha de Solicitud: 28 de marzo de 2005

Título: MAPEADO DE UN MODELO DE SISTEMA DE ARCHIVO A IN
OBJETO DE BASE DE DATOS

Mail Stop Non-Fee Amendment

Commissioner for Patents

P.O.Box 1450

Alexandria, Virginia 22313-1450

12

ENMIENDA PRELIMINAR

Apreciado señor:

Respetuosamente se solicita la entrada de esta enmienda preliminar antes de que se realice un examen sustancial de la solicitud de patente identificada arriba.

SEGUNDA PÁGINA:

11/091,079

MS311235.01/MSFTP852US

ENMIENDAS A LAS REIVINDICACIONES

El listado de reivindicaciones reemplazará todas las versiones anteriores y listas de reivindicaciones en la solicitud:

Lista de Reivindicaciones

1. (Original) Un sistema que facilita el mapeado de un modelo de datos que comprende:
un modelo de almacenamiento de archivo que se utiliza para definir un esquema de mapeado; y
un componente de almacenamiento de tipos que mapea un objeto de esquema a un objeto de base de datos basado en el esquema de mapeo asociado, donde se almacena una instancia de un tipo en el esquema de mapeo.
2. (Original) El sistema de la reivindicación 1, la instancia del tipo, es por lo menos uno de: un documento; una imagen; música; un video; un contacto;

un mensaje; una persona; una organización; un correo electrónico; un fax; una voz y un clip de audio.

3. (Original) El sistema de la reivindicación 1, el componente de almacenamiento de tipo, además proporciona una consulta para encontrar por lo menos uno de los siguientes: por lo menos un ítem en el sistema que satisfaga un criterio respectivo; por lo menos un documento en el sistema que satisfaga un criterio respectivo; y al menos un contacto que satisfaga un criterio respectivo.
4. (Original) El sistema de la reivindicación 3, el contacto, puede incluir por lo menos uno de los siguientes: un individuo; una organización y un grupo.
5. (Original) El sistema de la reivindicación 1, además conteniendo un componente relacional que utilice por lo menos uno de un almacenamiento relacional y una capacidad de consulta relacional.
6. (Original) El sistema de la reivindicación 5, el componente relacional, utiliza una técnica de base de datos relacional que es una colección de ítems de datos organizados como una serie de tablas descritas formalmente, donde los datos pueden ser por lo menos uno de poder tener acceso sin reorganización y reensamblados sin reorganización.

PÁGINA TRES:

11/091,079

MS311235.01/MSFTP852US

7. (Original) El sistema de la reivindicación 1, el mapeado, describe por lo menos uno de los objetos de bases de datos creados basados en el esquema y cómo la instancia del tipo descrito en el esquema se almacena y se puede tener acceso a ella.

TF

8. (Original) el sistema de la reivindicación 1, el tipo en el esquema, mapea a una clase de ejecución de tiempo de lenguaje común (common language runtime – CLR) en un almacenamiento del modelo de datos de almacenamiento del archivo,
9. (Original) El sistema de la reivindicación 8, el tipo, es por lo menos uno de los siguientes: un ítem; una extensión del ítem; un fragmento del ítem; y un vínculo.
10. (Original) El sistema de reivindicación 1, además comprendiendo una tabla que tiene por lo menos uno de los siguientes: 1) una columna que contiene por lo menos una instancia del tipo respectivo, y 2) una fila que contiene una representación en serie de la instancia de clase de CLR que representa la instancia de tipo.
11. (Original) El sistema de reivindicación 1, además comprendiendo un componente de vista que genera una vista que expone por lo menos una instancia del tipo, donde la vista proyecta una sub-serie del tipo respectivo de la vista asociada con el tipo de base.
12. (Original) El sistema de reivindicación 1, el tipo, contiene por lo menos uno de los siguientes, una jerarquía de tipo y un legado.
13. (Antes 14) (Enmendado Actualmente). Un medio legible por computador que ha almacenado allí los componentes del sistema de la reivindicación 1.

PÁGINA CUATRO:

11/091.079

MS311235.01/MSFTP852US

15

14. (antes 15) (Enmendado Actualmente) Un método implementado por computador que facilita el mapeado de un modelo de datos ue comprende:
la obtención de un modelo de datos de almacenamiento de archivo;
recibir por lo menos uno de un esquema, un tipo, un criterio y una información de consulta:
mapear un objeto de esquema a un objeto de base de datos basado en el esquema asociado al modelo de datos de almacenamiento de archivo; y
almacenar una instancia del tipo desde el esquema.
15. (Antes 16) (Enmendado Actualmente) El método de la reivindicación 14 (antes 15) además comprendiendo la consulta para encontrar por lo menos uno de los siguientes: por lo menos un item en el sistema que satisfaga un criterio respectivo; por lo menos un documento en el sistema que satisfaga un criterio respectivo y, por lo menos un contacto que satisfaga un criterio respectivo.
16. (Antes 17) (Enmendado Actualmente) El método de la reivindicación 14 (antes 15), además comprendiendo la utilización de un motor de base de datos relacional para proporcionar un almacenamiento relacional y una capacidad de consulta relacional.
17. (antes 18) (Enmendado Actualmente) El método de la reivindicación 14 (antes 15), además comprendiendo la generación de una vista que expone por lo menos una instancia del tipo, donde la vista proyecta una sub serie del tipo respectivo de la vista asociada con el tipo base.
18. (antes 19) (Enmendado Actualmente) Un paquete de datos que comunican entre un modelo de datos de almacenamiento de archivo y un componente

+6

de almacenamiento de tipo, el paquete de datos facilita el método de la reivindicación 14 (antes 15).

19. (antes 20) (Enmendado Actualmente) Un sistema implementado por computador que facilita el mapeado de un modelo de datos, que comprende:

los medios para recibir por lo menos uno de los siguientes: un esquema, un tipo, un criterio y un criterio de consulta; y

los medios para mapear un objeto de esquema a un objeto de base de datos basado en un esquema asociado a un modelo de datos de almacenamiento de archivo, donde se almacena una instancia de un tipo en el esquema.

PÁGINA CINCO:

11/091,079

MS311235.01/MSFTP852US

20. (Nuevo) El sistema de la reivindicación 11, el componente de vista invoca una interfaz de usuario para interactuar entre un usuario y el componente de almacenamiento de tipo.

PÁGINA SEIS:

11/091,079

MS311235.01/MSFTP852US

COMENTARIOS

Favor agregar la nueva reivindicación 20 a la solicitud relevante de acuerdo con lo presente. Por favor tenga en cuenta que las reivindicaciones originales presentadas del 14-20, han sido reenumeradas del 13-19 y la adición de la

17

reivindicación 20 no causa derechos adicionales. Se puede encontrar un listado completo de las reivindicaciones de las páginas 2-5. El representante de los solicitantes declara que no se ha introducido ningún nuevo asunto a la presente solicitud por esta enmienda preliminar.

Si el Examinador considera que una entrevista telefónica sería de ayuda para hacer expedito el proseguimiento favorable, se invita al Examinador a que se ponga en contacto con el representante que suscribe en el número telefónico que aparece abajo.

En el caso de que se deban derechos adicionales en conexión con la presentación de este documento, incluyendo extensiones adicionales de tiempo que se puedan requerir, el Comisionado está autorizado para cobrar estos derechos a la Cuenta de Depósito N° 50-1063.

Respetuosamente,

AMIN & TUROCY, LLP

(FIRMADO) – Himanshu S. Amin – Reg. N° 40,894

AMIN & TUROCY, LLP

24th. Floor, National City Center

1900 E. 9th. Street

Cleveland, Ohio 44114

Teléfono: (216) 696-8730

Fax: (216) 696-8731

Title: MAPEADO DE UN MODELO DE SISTEMA DE ARCHIVOS A UN OBJETO DE BASE DE DATOS

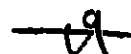
CAMPO TÉCNICO

[0001] El actual invento generalmente se relaciona con bases de datos y en particular con los sistemas y métodos que facilitan el almacenamiento de una instancia de tipos y/o datos de consulta.

ANTECEDENTES DEL INVENTO

[0002] Los avances en la tecnología de computadores (e.g., velocidad del microprocesador, capacidad de memoria, ancho de banda de transferencia de datos, funcionalidad del software, y similares) han contribuido en general al incremento de aplicaciones de computador en varias industrias. Aún los más potentes sistemas de servidores, los cuales frecuentemente están configurados como conjunto de servidores, se proveen generalmente, por ejemplo, a requerimientos de servicios originados de fuentes externas tales como la World Wide Web.

[0003] A medida que la cantidad de datos electrónicos disponibles crece, es mucho más importante almacenar dicha información de manera manejable que facilite las búsquedas y la recuperación de datos rápida y amigable al usuario. Actualmente, el enfoque común es almacenar los datos electrónicos en una o más bases de datos. En general, se denomina una base de datos típica como una colección organizada de información con estructura de datos de tal forma que, por ejemplo, un programa de computador pueda rápidamente buscar y seleccionar las porciones de datos que se quieren. Generalmente, los datos dentro de una base de datos están organizados vía una o más tablas. Dichas tablas pueden estar organizadas con filas y columnas.



[0004] Las tablas pueden comprender un registro o más, mientras que un registro puede incluir una serie de campos. Los registros se indexan como filas dentro de una tabla y los campos de los registros se indexan comúnmente como columnas, de tal manera que un par de índices fila/columna puede referenciar unos datos particulares dentro de una tabla. Por ejemplo, una fila puede almacenar un registro completo de datos que se refieren a una transacción de ventas, persona o proyecto. Asimismo, las columnas de la tabla pueden definir discretas porciones de las filas que tienen el mismo formato general de datos, mientras que las columnas pueden definir los campos de los registros.

[0005] Cada porción individual de datos, por sí sola, generalmente no es muy informativa. Las aplicaciones de bases de datos hacen que éstos sean más útiles porque le ayudan a los usuarios a que organicen y procesen los datos. La aplicación de base de datos permite al usuario comparar, escoger, ordenar, juntar, separar e interconectar los datos de tal manera que se pueda generar información útil de los datos. Empero, la capacidad y versatilidad de las bases de datos han crecido de manera tan increíble que permiten una capacidad de almacenamiento virtualmente sin fin utilizándolas. Además, los sistemas comunes de bases de datos ofrecen una limitada habilidad para consulta basados en el tiempo, la extensión del archivo, la locación y el tamaño. Por ejemplo, con el fin de buscar las enormes cantidades de datos asociados a una base de datos, una búsqueda normal se limita al nombre del archivo, el tamaño, fecha de creación, donde tales técnicas son deficientes e ineptas.

[0006] Con la continua e incrementada creación de datos de los usuarios finales, los problemas y dificultades alrededor de la búsqueda, relación y almacenamiento de dichos datos está llegando a su tope. Los usuarios finales escriben documentos, almacenan fotos, extraen música de discos compactos, reciben correo electrónico, guardan copias de correos electrónicos enviados, etc. Por ejemplo, en el simple proceso de crear un disco compacto de música, el

usuario final puede crear megabytes de datos. La extracción de la música de un disco compacto, la conversión del archivo a un formato adecuado, la creación de la cobertura de la caja, el diseño de la etiqueta del disco, todo requiere la creación de datos.

[0007] Las complicaciones no solo están alrededor de los usuarios sino que los que desarrollan tienen asuntos similares con los datos. Los que desarrollan crean y escriben miríadas de aplicaciones que varían desde las personales hasta las empresariales altamente desarrolladas. Los desarrolladores frecuentemente, si no siempre, mientras están creando y/o desarrollando, están recolectando datos. A medida que obtienen dichos datos, éstos deben almacenarse. En otras palabras, los problemas y dificultades alrededor de la búsqueda, relación y almacenamiento de datos afectan tanto al desarrollador como al usuario final. En vista de lo anterior, existe la necesidad de mejorar y/o proveer sistemas y/o métodos que mitiguen las deficiencias asociadas con los sistemas y bases de datos convencionales.

RESUMEN DEL INVENTO

[0008] Lo siguiente presenta un resumen simplificado del invento con el fin de proporcionar un entendimiento básico de algunos de los aspectos de éste. Este resumen no es una extensa información general. No se propone ni identificar elementos claves o críticos ni delinear el alcance del invento. Su único objetivo es presentar algunos conceptos del invento de manera simple como preludio a la descripción más detallada que se presenta más adelante.

[0009] La materia del invento se relaciona con los sistemas y/o métodos que facilitan el mapeado de un modelo de datos a un objeto de bases de datos. Un sistema de almacenamiento de tipos puede proporcionar almacenamiento de una instancia de tipo y/o para consulta. El almacenamiento del tipo puede describir el

objeto de la base de datos y cómo los tipos de instancia se almacenan y/o se puede tener acceso a ellas. La consulta pueden encontrar por lo menos uno de los siguientes, un ítem, un documento y un contacto que cada uno satisfaga un cierto criterio. Además, el sistema de almacenamiento de tipos puede utilizar una interfase que reciba datos, mientras que los datos pueden incluir un esquema, un tipo, un criterio, una consulta de criterios, etc.

[0010] De acuerdo con un aspecto del objeto del invento, el sistema de almacenamiento de tipos puede incluir un componente de almacén que almacena por lo menos una instancia de tipo. El almacenamiento puede ser un mapeado del modelo de datos, donde el modelo de datos puede representar un sistema de almacenamiento de archivos. Además, el sistema de almacenamiento de tipos puede incluir también un componente de consulta. El componente de consulta puede suministrar por lo menos una consulta en el modelo de datos del sistema de almacenamiento de archivos para satisfacer un criterio. De acuerdo con otro aspecto del objeto del invento, el sistema de almacenamiento de tipos puede incluir un componente relacional que puede utilizar un almacenamiento relacional y/o capacidad de consulta relacional. El componente relacional puede invocar un motor de bases de datos para que proporcione dichas técnicas relacionales, donde estas técnicas pueden facilitar el almacenamiento de instancias y/o consultas de tipo.

[0011] Además, el sistema de almacenamiento de tipo puede también incluir un componente de vista que puede proporcionar una vista que expone todas las instancias de un tipo dado. El tipo puede ser una estructura jerárquica y/o una estructura de legado. La vista se puede relacionar con un tipo particular que proyecte una sub serie de tipos con un tipo base. En otros aspectos del invento, se proveen métodos que facilitan el mapeado de un modelo de datos a un objeto de base de datos.

[0012] La siguiente descripción y los dibujos anexos presentan en detalle ciertos aspectos ilustrativos del invento. Sin embargo, estos aspectos son indicativos de solo algunas de las varias maneras en que los principios del invento pueden emplearse y el asunto del invento tiene el propósito de incluir todos los dichos aspectos y sus equivalentes. Otras ventajas y características novedosas del invento se volverán aparentes de la siguiente descripción detallada del invento cuando se consideran en conjunto con los dibujos.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0013] La Fig. 1 ilustra un diagrama en bloque de un sistema ejemplar que facilita el almacenamiento de un tipo de instancia asociado con el modelo de datos.

[0014] La Fig. 2 ilustra un diagrama de bloque de un sistema ejemplar que facilita el almacenamiento de un tipo de instancia y/o de consulta para encontrar por lo menos uno de los siguientes, un ítem, un documento y un contacto.

[0015] La Fig. 3 ilustra un diagrama de bloque de un sistema ejemplar que facilite la utilización de almacenamiento relacional y/o capacidades de consulta relacionales.

[0016] La Fig. 4 ilustra un diagrama en bloque de un sistema ejemplar que facilita el mapeado y/o la vista en conjunto con un sistema de almacenamiento de tipo.

[0017] La Fig. 5 ilustra un diagrama en bloque de un sistema ejemplar que facilita el almacenamiento de in tipo de instancia asociado con un modelo de datos.

[0018] La Fig. 6 ilustra un diagrama en bloque de una estructura de alto nivel de un almacén dentro de un modelo de datos de almacenamiento de archivos.

[0019] La Fig. 7 ilustra un diagrama en bloque de una instancia de tipo con tablas asociadas.

[0020] La Fig. 8 ilustra un diagrama en bloque de una jerarquía de tipo y su correspondiente proyección de vista.

[0021] La Fig. 9 ilustra una metodología ejemplar para el almacenamiento de un tipo de instancia que mapea a un modelo de datos y/o que suministra una consulta.

[0022] La Fig. 10 ilustra una metodología ejemplar para el almacenamiento de instancia de tipo que mapea a un modelo de datos y/o que suministra una consulta.

[0023] La Fig. 11 ilustra un ambiente ejemplar de operaciones en red, donde pueden ser empleados los aspectos novedosos del objeto del invento.

[0024] La Fig. 12 ilustra un ambiente operativo ejemplar que puede ser empleado de acuerdo con el objeto del invento.

DESCRIPCIÓN DEL INVENTO

[0025] Tal como se utilizan en esta aplicación, los términos "componente", "sistema," "interfaz," "esquema," y similares, se refieren a una entidad relacionada con computación, ya sea hardware, software (e.g., en ejecución), y/o programación fija. Por ejemplo, un componente puede ser un proceso que se está ejecutando en

un procesador, un procesador, un objeto, un ejecutable, un programa y/o un computador. Para ilustración, tanto una aplicación que se ejecuta en un servidor y el servidor mismo pueden ser un componente. Uno o más componentes pueden residir dentro de un proceso y un componente puede estar localizado en un computador y/o distribuido entre dos o más computadores.

[0026] El objeto del invento se ha descrito con referencia a los dibujos, donde los numerales similares de referencia se utilizan para referenciar elementos similares por todo este. En la siguiente descripción, como explicación, se presentan numerosos detalles específicos con el fin de suministrar una comprensión completa del objeto del invento. Sin embargo, puede ser evidente que el objeto del invento puede practicarse sin estos detalles específicos. En otras instancias, se muestran en forma de diagramas en bloque las estructuras y aparatos bien conocidos con el fin de facilitar la descripción del objeto del invento.

[0027] Ahora, volviendo a las figuras, la Fig. 1 ilustra un sistema 100 que facilita el almacenamiento de una instancia de tipo asociada con un modelo de datos. El modelo de datos puede ser un modelo de datos de almacenamiento de archivo 102 que permite el almacenamiento, el hallazgo y relacionar la información. Por ejemplo, un tipo de información puede ser, pero no se limita a ser, un documento, una imagen, un video, un contacto, un mensaje, un correo electrónico, un clip de audio, etc. Los tipos informacionales pueden considerarse como unidades de información, los cuales pueden ser representados como instancias de tipos complejos que son parte de un sistema de tipos que apoyan el legado. Se puede definir el legado como una situación en la cual algunas características se pasan de un contexto a otro. En particular a la programación orientada hacia el objeto, los objetos heredan atributos y/o comportamientos de otros objetos. Se debe apreciar que la herencia (legado) puede considerarse como una estructura y/o formato jerárquico.

[0028] Un sistema de almacenamiento de tipo 104 puede almacenar la instancia de tipo y consulta para encontrar eficiente y efectivamente por lo menos uno de los siguientes, ítems, documentos y/o contactos. El sistema de almacenamiento de tipo 104 puede recibir datos, donde éstos pueden incluir un tipo, un criterio, un esquema, un criterio de consulta, Específicamente, almacenar la instancia de tipo que se relaciona con un tipo informacional (e.g., un documento, una imagen, un video, un contacto un mensaje, un correo electrónico, un clip de audio) puede proporcionar por lo menos uno de los siguientes: 1) encontrar por lo menos un ítem en el modelo de datos de almacenamiento de archivo 102 que satisfaga un cierto criterio. 2) encontrar por lo menos un documento en el modelo de datos de almacenamiento de archivo 102 que satisfaga un criterio en particular; y 3) encontrar por lo menos un contacto (e.g., incluyendo un individuo, organización y grupo) que satisfaga un cierto criterio.

[0029] El sistema de almacenamiento de tipo 104 puede utilizar una técnica de base de datos relacional asociada con un almacenamiento relacional y una consulta relacional. Se debe reconocer que dichas capacidades pueden ser suministradas por un motor de bases de datos (no ilustrado). Las técnicas de base de datos relacionales pueden asociarse a las bases de datos relacionales, donde una base de datos relacional es una colección de ítems de datos organizada como una serie de tablas descritas formalmente. Se puede tener acceso a los datos dentro de la tabla y/o pueden ser reensamblados de varias maneras sin requerir reorganización de las tablas de bases de datos. Además, las bases de datos relacionales se pueden ampliar fácilmente, tal como agregándoles nuevas categorías sin modificaciones a aplicaciones y/o datos existentes. Se debe apreciar que el objeto del invento no está limitado a bases de datos relacionales y/o a técnicas asociadas y que se puede utilizar cualquier técnica apropiada.

[0030] El sistema 100 además incluye un componente de interfase 106, el cual provee varios adaptadores, conectores, canales, vías de comunicación, etc.

para integrar el sistema de almacenamiento de tipos 104, a virtualmente cualquier sistema operativo. Además, el componente de interfase 106 puede proveer varios adaptadores, conectores, canales, vías de comunicación, etc. que proporcionen una interacción con los datos y con el sistema de almacenamiento de tipos 104. Se debe apreciar que aunque el componente de interfase 106 está incorporado dentro del sistema de almacenamiento de tipos 104, dicha implementación no es tan limitada. Por ejemplo, el componente de interfase 106 puede ser un componente que esté solo para recibir o transmitir los datos con relación al sistema 100.

[0031] La Fig. 2 ilustra un sistema 200 que facilita el almacenamiento de una instancia de tipo y/o de consulta para encontrar un ítem, un documento, y un contacto asociado al modelo de datos 202. El modelo de datos 202 puede ser la representación de un modelo de un sistema de almacenamiento de archivos que utiliza una característica jerárquica y/o legado. El tipo puede incluir un documento, una imagen, un video, un contacto, un mensaje, un correo electrónico, un clip de audio, etc. Sin embargo, se debe apreciar que un tipo puede ser un tipo de información común que está almacenado en un sistema representado por el modelo de datos 202. Un sistema de almacenamiento de tipos 204 puede almacenar la instancia de tipo y suministrar consultas que puedan eficientemente por lo menos encontrar una de las siguientes; un ítem en el modelo de datos 202, un documento en el modelo de datos 202, y un contacto en el modelo de datos 202. El sistema de almacenamiento de tipo 204 puede recibir datos, donde los datos pueden ser un criterio de consulta, un esquema, un criterio, una definición de esquema, un modelo de datos, un tipo, El sistema 200 puede además emplear una interfase 206 para facilitar el suministro de varios adaptadores, conectores, canales, vías de comunicación, etc. para integrar el sistema de almacenamiento de tipos 204 en virtualmente cualquier sistema operativo y facilitar la comunicación.

[0032] El sistema de almacenamiento de tipos 204 puede incluir además un componente de almacenamiento 208 ("almacén 208") que puede almacenar las instancias de tipo. El almacenamiento puede ser un mapeado del modelo de datos 202, donde el mapeado del almacenamiento puede describir el objeto de la base de datos que está creado basado en una definición de esquema y cómo las instancias de tipos descritas en el esquema se almacenan y/o pueden ser accedidas. En otras palabras, el almacén 208 puede almacenar las instancias del tipo y por lo menos una regla asociada a mapear una declaración de tipo dentro del objeto de la base de datos.

[0033] El sistema de almacenamiento de tipo 204 puede incluir un componente de consulta 210 ("consulta 210") que proporciona consulta de datos. La consulta 210 puede obtener por lo menos uno de los siguientes: un ítem en el sistema representado por el modelo de datos 202; un documento en el sistema representado por el modelo de datos 202; y un contacto (e.g., incluyendo un individuo, una organización y un grupo) en el sistema representado por el modelo de dato 202. Se debe apreciar y comprender que la consulta puede estar basada en por lo menos ciertos criterios y/o criterios de consulta obtenidos vía la interfase 206. Además, la consulta 210 no está tan limitada al ítem, documento y contacto y cualquier tipo de información apropiada almacenada en el sistema representado por el modelo de datos 202 puede ser utilizado.

[0034] La Fig. 3 ilustra un sistema 300 que facilita la utilización de un almacenamiento relacional y capacidades de consulta. Un sistema de almacenamiento de tipo 304 puede almacenar una instancia y consulta de tipo para eficiente y efectivamente se encuentren ítems, documentos y contactos, donde dichos elementos están asociados a un sistema de archivos representado por un modelo de datos 302. Se debe apreciar que el sistema de archivos puede ser un sistema de almacenamiento de archivos que utiliza una estructura jerárquica y/o una característica de herencia (legado). El sistema de

Handwritten signature

almacenamiento de tipo 304 puede ser sustancialmente similar al sistema de almacenamiento de tipos 104, 204, como se muestra en las Figs 1 y 2 respectivamente. El sistema de almacenamiento de tipos 304 puede invocar una interfase 306 para facilitar las comunicaciones y/o recibir datos para que sean utilizados más adelante de acuerdo con el objeto del invento.

[0035] El sistema de almacenamiento de tipos 304 puede incluir un componente de almacén 308 (almacén 308) y un componente de consulta 310 ("consulta 310"). El almacén 308 puede proporcionar cualquier técnica de almacenamiento adecuado para almacenar instancias de tipo. La consulta 310 puede suministrar una consulta que pueda eficiente y efectivamente obtener por lo menos una de las siguientes: por lo menos un ítem en un sistema que cumpla con un criterio; por lo menos un documento en un sistema que satisfaga un criterio; y por lo menos un contacto que cumpla con un criterio. Se debe apreciar y comprender que el almacén 308 y la consulta 310 pueden ser sustancialmente similares al almacén 208, y a la consulta 210 como se muestra en la Fig. 2.

[0036] El sistema de almacenamiento de tipos 304 puede también incluir un componente relacional 312 ("relacional 312"). El relacional 312 utiliza técnicas de bases de datos (e.g., utilizando un motor de bases de datos) para construir un almacenamiento relacional y/o suministrar una capacidad de consulta relacional para facilitar el almacenamiento de las instancias y/o consultas de tipos. El relacional 312 puede incorporar técnicas asociadas a una base de datos relacional, mientras que la base de datos relacional es una colección de ítems de datos organizados como una serie de tablas descritas formalmente, como se describen en detalle más arriba. Se debe apreciar que el objeto del invento no está limitado a la base de datos relacional y técnicas asociadas y que se puede utilizar cualquier técnica apropiada.

[0037] La Fig. 4 ilustra un sistema 400 que facilita el mapeado y/o vista en conjunción con un modelo de datos 402. El modelo de datos 402 puede representar un sistema de almacenamiento de archivo que permite el almacenamiento, el hallazgo y el relacionar la información. Un tipo de información típica que pueda ser almacenado en el sistema puede incluir un documento, una imagen, música, un video, un contacto, un mensaje, etc. El tipo de información puede ser representado como instancias de tipos complejos que son parte de un sistema de tipos que apoyan la herencia. Un sistema de almacenamiento de tipos 404 puede almacenar las instancias de tipo y suministrar una consulta para encontrar ítems, documentos y contactos que satisfaga ciertos criterios. El sistema 400 puede también emplear una interfase 406 para facilitar la comunicación y/o recibir datos que pueden incluir criterios, tipos, esquemas, modelos y criterios de consulta.

[0038] El sistema de almacenamiento de tipos 404 puede también incluir un componente de mapa 408 para facilitar el almacenamiento de la instancia del tipo. El componente de mapa 408 suministra el mapeado de tipos descritos en un esquema para definir tipos y objetos de bases de datos. El componente de mapa 408 puede ser un mapeado de almacenamiento que describe por lo menos un objeto de base de datos, donde el objeto de base de datos puede ser creado basado en una definición de esquema y/o cómo las instancias de tipos descritos en el esquema se almacenan y/ o se puede tener acceso a ellos. En otras palabras, las instancias de tipos pueden almacenarse y se pueden utilizar reglas para mapear una declaración de tipo en el objeto de la base de datos. Cada tipo en el esquema mapea a una clase (e.g., tiempo de ejecución de lenguaje común (common language runtime -CLR)) en un almacenamiento.

[0039] El sistema de almacenamiento de tipos 404 puede incluir un componente de vista 410 para proveer una proyección de vista. La proyección de vista puede exponer instancias del tipo que se está viendo. Se debe apreciar que

los tipos pueden estar en una estructura jerárquica utilizando por lo menos el legado. En otras palabras, cada tipo es parte de una jerarquía de tipo. La vista puede asociarse con un tipo dado y puede proyectar una sub serie de los respectivos tipos de la vista asociados con su tipo base. La vista puede proyectar instancias asociadas con el tipo particular. Por ejemplo, para un tipo "mensaje", solo las instancias en las cuales el mensaje es un padre pueden ser vistas basadas pro lo menos en la estructura de jerarquías.

[0040] El componente de vista 410 puede proporcionar varios tipos de interfases de usuario para facilitar la interacción entre un usuario y cualquier componente unido al sistema de almacenamiento de tipos 404. Como se demostró, el componente de vista 410 es una entidad separada que está incorporada dentro del sistema de almacenamiento de tipos 404. Sin embargo, se debe apreciar que el componente de vista 410 y/o similares componentes de vista pueden ser componentes separados del sistema de almacenamiento de tipos 404 y/o de una unidad que está sola. El componente de vista 410 puede proveer una o más interfases gráficas de usuario (graphical user interfaces -GUI), interfases de línea de comando, y similares. Por ejemplo, se puede presentar una GUI que le suministra al usuario con una región o con los medios para cargar, importar, leer, etc. datos y puede incluir una región para presentar dichos resultados. Estas regiones pueden comprender textos conocidos y/o regiones gráficas que comprenden cajas de diálogos, controles estáticos, menús desplegables, cajas de listas, menús emergentes, como controles de edición, cajas de combos, botones de radio, cajas de chequeos, botones de empujar, y cajas gráficas. Adicionalmente, herramientas para facilitar la presentación tales como barras desplegables verticales y/o horizontales para navegación y botones de barras de herramientas para determinar si una región que se podrá ver podrá emplearse. Por ejemplo, el usuario puede interactuar con uno o más de los componentes unidos al sistema de almacenamiento de tipos 404.

[0041] El usuario también puede interactuar con las regiones para seleccionar y suministrar información v/a varios aparatos, tales como por ejemplo, un ratón, un roller ball, un teclado numérico, un teclado, un puntero y/o activación de voz. Generalmente, un mecanismo como un botón ara pulsar o la tecla de "enter" en el teclado puede emplearse luego de que se introduce la información con el fin de iniciar la búsqueda. Sin embargo, se debe apreciar que el invento no es tan limitado. Por ejemplo, simplemente al resaltar una caja de chequeo puede iniciarse la transmisión de la información. En otro ejemplo, se puede utilizar una interfase de línea de comando. Por ejemplo la interfase de línea de comando puede invitar al usuario (e.g., vía un mensaje de texto en una pantalla y un tono de audio) a que mande la información al proporcionarle un mensaje de texto. El usuario entonces puede suministrar la información adecuada, como una entrada alfa numérica correspondiente a una opción suministrada en el mensaje de la invitación de la interfase o una respuesta a una pregunta hecha en la invitación. Se debe apreciar que la interfase de línea de comando puede ser utilizada en conexión con una GUI y/o API. Adicionalmente, la interfase de línea de comando puede utilizarse en conexión con hardware (e.g., tarjeta de video) y/o despliegues (e.g., negro y blanco y EGA) con soporte gráfico limitado y/o canales de comunicación con ancho de banda bajo.

[0042] La Fig. 5 ilustra un sistema 500 que emplea la inteligencia para facilitar el almacenamiento de una instancia de tipo asociada con un modelo de datos 502. El sistema 500 incluye un sistema de almacenamiento de tipos 504, una interfase 506, y el modelo de datos 502, los cuales pueden ser sustancialmente similares a los componentes descritos en las figuras anteriores. La interfase 506 puede facilitar la comunicación asociada a los datos, lo cual puede incluir un criterio, un tipo, un esquema, un modelo de datos y un criterio de consulta. El sistema 500 puede proporcionar el almacenamiento de un tipo, consultar y/o suministrar una vista. Se debe apreciar y comprender que un motor

de base de datos puede proveer un almacenamiento relacional y una consulta relacional al sistema 500.

[0043] El sistema 500 además incluye un componente inteligente 508. El componente inteligente 508 puede ser utilizado por el sistema de almacenamiento de tipos 504 para facilitar el almacenamiento y/o consulta para el sistema 500. Por ejemplo, el componente inteligente 510 puede ser utilizado para facilitar el determinar el tipo de usuario definido a ser almacenado. Los datos históricos, en conjunción con los perfiles del usuario pueden permitir al componente inteligente 508 el determinar el almacenamiento del tipo y/o consulta con cierto criterio.

[0044] Se debe entender que el componente inteligente 508 puede proporcionar razonamientos sobre los estados del sistema, el ambiente y/o del usuario o inferir sobre ellos desde una serie de observaciones que han sido capturadas vía eventos y/o datos. La inferencia puede emplearse para identificar un contexto específico o acción, o puede generar una distribución de probabilidades sobre los estados, por ejemplo. La inferencia puede ser probabilística – esto es, la computación de una distribución de probabilidades sobre los estados de interés basados en una consideración de datos y eventos. La inferencia puede también referirse a técnicas empleadas para componer eventos a alto nivel de una serie de eventos y/o datos. Dicha inferencia resulta en la construcción de nuevos eventos o acciones de una serie de eventos observados y/o datos de eventos almacenados, sean éstos correlacionados en proximidad temporal cercana o no y si los eventos y los datos provienen de una o de varias fuentes de eventos o datos. Se pueden emplear varias clasificaciones (entrenadas explícita y/o implícitamente) esquemas y/o sistemas (e.g., máquinas vectoras de soporte, redes neutras, sistemas expertos, redes de creencia de Bayes (Bayesian belief networks), lógica difusa, motores de fusión de datos...) en conexión con la acción automática desempeñada y/o inferida en conexión con el objeto del invento.

[0045] El clasificador es una función que mapea un vector de atributos de entrada, $x = (x_1, x_2, x_3, x_4, x_n)$, a una confianza que la entrada pertenece a una clase, que es, $f(x) = \text{confidence}(\text{class})$. Dicha clasificación puede utilizar análisis probabilística y/o basado en estadísticas (e.g., factorando dentro de las utilidades y costos del análisis) para diagnosticar o inferir una acción que el usuario desea que se realice automáticamente. Una máquina vectorial de soporte (support vector machine - SVM) es un ejemplo de un clasificador que se puede utilizar. La SVM opera encontrando una hiper-superficie en el espacio de posibles entradas, la cual trata de dividir el criterio iniciador de los eventos no iniciadores. Intuitivamente, esto hace que la clasificación sea correcta para comprobar los datos que están cerca, pero no idéntica a los datos de entrenamiento. Otros enfoques de clasificación de modelos directos e indirectos incluyen, e.g., Bayes crédulas (naïve Bayes), redes de Bayes, árboles de decisiones (decision trees), redes neurales (neural networks), modelos de lógica difusa y modelos de clasificación probabilística que suministran diferentes patrones de independencia que se pueden utilizar. La clasificación como se utiliza en este también es inclusiva de regresión estadística que se utiliza para desarrollar modelos de prioridad.

[0046] Además, el componente inteligente 508 puede utilizar un almacén de datos 510 para almacenar perfiles de usuarios y/o datos históricos. El almacén de datos 510 puede ser, por ejemplo, memoria volátil o no volátil, o puede incluir tanto una como la otra. Como ilustración, y no limitación, la memoria no volátil puede incluir memoria de solo lectura (ROM), ROM programable (PROM), ROM eléctricamente programable (EPROM), ROM eléctricamente programable borrrable (EEPROM), o memoria flash. La memoria volátil puede incluir memoria de acceso aleatoria (RAM), la cual actúa como una memoria caché externa. Como ilustración y no limitación, la RAM se encuentra disponible en muchas formas, tales como le RAM estática (SRAM), RAM dinámica (DRAM), DRAM sincronizada (SDRAM), SRAM de tasa de doble dato (DDR SDRAM), (double data rate SDRAM), SDRAM avanzado (ESDRAM), (enhanced SDRAM) DRAM Synchlink (SLDRAM), RAM

directo de Rambus (RDRAM), RAM dinámico directo de Rambus (DRDRAM), y RAM dinámico de Rambus (RDRAM). El almacén de datos 510 de los sistemas y métodos del objeto intentan abarcar, sin limitarse a éstas y a cualquier otro tipo de memoria adecuada. Adicionalmente, se debe apreciar que el almacén de datos 510 puede ser un servidor y/o base de datos.

[0047] La Fig. 6 ilustra la estructura de alto nivel de un almacén. Se puede proveer un esquema donde varias instancias de tipos se pueden almacenar en tablas dependientes de su tipo. Los tipos pueden ser, pero no se limitan a, un ítem, la extensión de un ítem, el fragmento de un ítem y un vínculo. Cada tipo puede tener una tabla correspondiente que puede incluir columnas que tienen instancias de objeto en ellas. Por ejemplo, una columna en la tabla de ítems puede contener todas las instancias de ítems en el almacén. Para cada fila, la columna puede contener una representación serializada de la instancia de clase CLR que representa la instancia de tipo del ítem. Debe apreciarse que para la extensión del ítem, los fragmentos de ítem y las instancias de tipo vínculo pueden estar representadas en una estructura similar. Las instancias de tipo en línea pueden almacenarse dentro de una instancia de objeto padre más que almacenarse en tablas separadas y/o columnas. Para cada ítem, extensión de ítem, fragmento de ítem y vínculo, se puede generar una vista que expone todas las instancias de ese tipo. Cada tipo es parte de una jerarquía de tipo. Por ejemplo, la vista del ítem proyecta todos los ítems del almacén.

[0048] Volvamos brevemente a la to Fig. 7, donde se muestra un ítem 702. una tabla 704 del ítem 702 puede contener todas las instancias de los ítems en el almacén (no demostrado). La tabla 704 puede contener una instancia de objeto 706 ("Documento") y una instancia de objeto 708 ("Contacto"). Se puede apreciar que cada tipo en un esquema mapea a la clase de CLR en el almacenamiento. La instancia de objeto 706 puede incluir un título, un resumen, un impreso, un autor y otros varios metadatos asociados con la instancia del objeto 706. De manera

similar, la instancia del objeto 708 puede incluir varios metadatos tales como un nombre, una dirección, un correo electrónico, etc.

[0049] La Fig. 8 ilustra una jerarquía de tipo 800 y su correspondiente proyección de vista 820. La jerarquía de tipo 800 puede incluir un ítem 802, un contacto 804, un documento 806, un mensaje 808, una persona 810, una organización 812, un correo electrónico 814, un fax 816, y una voz 818. Se debe apreciar que la jerarquía de tipo 800 es un ejemplo y cualquier jerarquía y/o tipos pueden ser utilizados de acuerdo con el objeto del invento. Como se ilustra, el ítem 802 se considera el padre de cada tipo incluido allí. Así, el contacto 804 es el padre de la persona 810 y de la organización 812, mientras que el mensaje 808 es el padre para el correo electrónico 814, fax 816, y voz 818. Además, la proyección de vista correspondiente 820 puede reflejar la jerarquía de tipo 800. La vista asociada con un tipo dado puede proyectar una sub serie de los ítems de la vista asociada con su tipo base. Por ejemplo, la vista del ítem proyecta todos los ítems en el almacén. La vista de contacto proyecta solamente los ítems que son del tipo Contacto. La vista de persona proyecta solo los contactos que tienen la mayoría de tipos como persona.

[0050] Volviendo a la Fig. 6, el mapeado de tipos puede suministrarse al utilizar un algoritmo para hacer el mapeado del tipo al tipo usado para describir la estructura de almacenamiento correspondiente en el almacén. Se debe apreciar que el mapeado puede asociarse a cualquier sistema de almacenamiento de archivo (e.g., un modelo de datos basado en un sistema que utiliza instancias complejas de tipos para describir y/o representar una unidad de información). Cada tipo declarado en un esquema está mapeado a una clase CLR que soporta el contrato UDT de SQL. Se debe apreciar que aunque el SQL se utiliza en los siguientes ejemplos, cualquier sistema de manejo de bases de datos adecuado puede utilizarse. Los tipos pertenecen a un espacio de nombre cuyo nombre corresponde a un espacio de nombre del esquema con el sufijo anexo de

“Almacén”. Los tipos en un espacio de nombre dado se compilan en un ensamblaje único que se utiliza como unidad de despliegue del esquema. El nombre del tipo de CLR es igual al nombre declarado en el esquema. Por cada propiedad declarada del tipo, se agrega lo siguiente al correspondiente tipo de CLR: 1) un campo privado con el nombre igual al nombre especificado en el esquema, con el prefijo de “m_”. Al campo se le atribuye con el atributo específico de UDT: System.Data.SqlTypes.SqlUdtField; 2) una propiedad pública con el nombre igual al nombre especificado en el esquema y que corresponde a declaraciones obtenidas/fijadas (get/set statements). A la propiedad se le atribuye el atributo específico de UDT System.Data.SqlTypes.SqlUdtProperty; y 3) el tipo de campo y la propiedad es el tipo de CLR que corresponde al tipo declarado en el esquema. Si el tipo es uno de aquellos escalares que está mapeado a uno de los tipos escalares SQL (tratados *infra*).

[0051] La siguiente tabla describe el mapeo de los tipos de escala del sistema de almacenamiento de archivos según los tipos de SQL administrados:

Tipo de Sistema de Almacenamiento de Archivo	Tipo de SQL Administrado	Descripción
String – Secuencia	SqlString	Información Unicode de longitud variable con una longitud máxima de 2^31 caracteres. La longitud se puede fijar de 1 – 4000 caracteres o sin restricciones utilizando la clave “max”
Binary - Binario	SqlBinary	Información Binaria de longitud variable con una longitud máxima de 2^32 bytes.

37.

		La longitud se puede fijar de 1 – 8000 bytes o sin restricciones utilizando la clave.
Boolean	SqlBoolean	Valor Boolean anulable
Byte	SqlByte	Un byte único sin firma
	SqlInt16	Información en enteros desde -2^15 (-32,768) hasta 2^15 - 1 (32,767).
Int32	SqlInt32	Información en enteros (número entero) desde -2^31 (-2,147,483,648) hasta 2^31 - 1 (2,147,483,647).
Int64	SqlInt64	Información en enteros (número entero) desde -2^63 (-9223372036854775808) hasta 2^63-1 (9223372036854775807).
Single - Sencillo	SqlSingle	Información en número de precisión flotante desde -3.40E + 38 hasta 3.40E + 38
Double - Doble	SqlDouble	Información en número de precisión flotante desde -1.79E + 308 hasta 1.79E + 308
Decimal	SqlDecimal	Tipos de información numérica con precisión y escala fijos. Los atributos Precisión y Escala aplican para las propiedades de tipo Decimal. La Precisión define el número máximo de dígitos decimales del valor decimal. Esto incluye los dígitos a la izquierda y a la derecha del punto decimal. El valor del atributo de Precisión en un entero entre 1 y 28.

~~38~~

		La Escala define el número máximo de dígitos decimales a la derecha del punto decimal. El valor del atributo de Escala debe estar entre 0 y el valor del atributo de Precisión. Nota para el hito B: estos atributos no están respaldados en este hito. La Precisión se fija en 28 y la Escala en 14.
DateTime – Fecha Hora	SqlDateTime	Fecha y Hora a partir de enero 1, 1753, a diciembre 31, 9999, con una precisión de trescientos partes de un segundo o 3.33 milisegundos.
Guid	SqlGuid	Un identificador único global - globally unique identifier (GUID).
Xml	SqlString	XML se almacena actualmente como un campo de secuencia. El respaldo al tipo Xml Nativo está siendo considerado para post-Beta1.
Stream	TBD	Se está planificando Respaldo para post-Beta1. En el caso de respaldarlo podrá reemplazar las declaraciones de secuencia (max) y binaria (max).

Tabla 1

[0052] Un objeto de base de datos creado en un almacén de sistema de almacenamiento de archivos puede ser almacenado en un nombre de esquema SQL derivado del nombre del esquema del sistema de almacenamiento de archivo. El sufijo ".Almacén" – (.Store) se adjunta al nombre del esquema del sistema de almacenamiento de archivo para generar el nombre del esquema SQL.

Por ejemplo, el esquema del sistema de almacenamiento de archivos genera objetos en el Esquema SQL "[Sistema-Almacenamiento.Almacén - System.Storage.Store]" como un ítem "[Sistema-Almacenamiento.Almacén - System.Storage.Store]."

[0053] El contenido del sistema de almacenamiento de archivo se puede acceder por las vistas. Las vistas descritas a continuación son de sólo lectura, pero el sujeto del invento no se encuentra tan limitado, ya que las vistas se pueden autorizar con escritura. Cada tipo de ítem puede ser mapeado a un tipo de vista. Cada tipo de vista del ítem se puede identificar en un almacén del sistema de almacenamiento de archivo utilizando la convención para el nombre [nombre de esquema <schema name>.Almacén .Store].[<Nombre de tipo de ítem - Item type name>]. Un tipo de vista para el tipo T puede retornar todos los ítems que son de tipo T y todos los tipos que se deriven de T. La vista que corresponde al tipo de Sistema.Almacenamiento.Ítem - System.Storage.Item es [Sistema.Almacenamiento.Almacén].[Ítem]. Esta vista puede retornar todos los ítems a un almacén de sistema de almacenamiento de archivos. La siguiente table describe las columnas de un tipo de vista de ítem:

Nombre de Columna	Tipo	Descripción
ItemId – Identificación del Ítem	[Sistema.Almacenamiento.Almacén].ItemId - Identificación del Ítem	Identificación del Ítem
TypeId – Tipo de identificación	[Sistema.Almacenamiento.Almacén]. Tipo de identificación – TypeId	Tipo de identificación Esquematizado
NamespaceName – Nombre espacio	nvarchar(255)	El nombre único del ítem

Nombre		
ContainerId – Contenedor de la Identificación	[Sistema.Almacenamiento.Almacén].ItemId - Identificación del Ítem	Identificación del Contenedor del Ítem
Item	Tipo de Ítem	Instancia del Tipo de Ítem
EntityState – Estado Entidad	[Sistema.Almacenamiento.Almacén]. Estado Entidad – EntityState	El estado de la información para un ítem
ObjectSize – Tamaño de Objeto	Ent Grande	Tamaño del Objeto Ítem (columna persistida)
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información – ChangeInformation	Cambiar la información de rastreo
ItemSyncMetadata – Sincronizar Metadata del Ítem	[Sistema.Almacenamiento.Almacén]. Sincronizar Metadata del Ítem – ItemSyncMetadata	Sincronizar metadata
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta del Bin-BinPathHandle	La manija de la ruta al ítem
PromotionStatus – Promoción Estado	Ent	El estado de la promoción del ítem

Tabla 2

[0054] Cada tipo de enlace queda mapeado a una vista tipo. Cada tipo de vista se identifica en un almacén de sistema de almacenamiento de archivo utilizando la convención de nombre [<schema name – nombre de



esquema>.Almacén - Store].[<enlace con tipo de nombre - link type name>]. La vista que corresponde al tipo de enlace de Sistema,Almacenamiento es [Sistema.Almacenamiento.Almacén - System.Storage.Store].[Link - Enlace]. Esta vista puede incluir todos los enlaces en un almacén de sistema de almacenamiento de archivo. La tabla siguiente describe las columnas de un tipo de vista de enlace:

Nombre de Columna	Tipo	Descripción
SourceRef – Ref Fuente	[Sistema.Almacenamiento.Almacén]. Identificación del ítem	Identificación del ítem del ítem fuente
LinkId – Identificación del Enlace	[Sistema.Almacenamiento.Almacén]. Identificación del Enlace	Identificación del Enlace
TargetRef – Referencia Objetivo	[Sistema.Almacenamiento.Almacén]. Identificación del ítem	Identificación del ítem del ítem objetivo
TypeId – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación del tipo de enlace de mayor derivación de la instancia del enlace
Link - Enlace	Tipo de Enlace	Instancia del tipo de enlace
EntityState – Estado	[Sistema.Almacenamiento.Almacén].	El estado de

de la Entidad	Estado de la Entidad	la información para el enlace
ObjectSize – Tamaño de Objeto	Ent Grande	Tamaño del objeto enlace (columna persistida)
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar la información de rastreo
LinkSyncMetadata – Sincronizar Metadata del Enlace	[Sistema.Almacenamiento.Almacén]. Sincronizar Metadata del ítem	Sincronizar metadata
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta del Bin-	La manija de la ruta al ítem que es la fuente del enlace

Tabla 3

[0055] Todos los fragmentos del ítem se pueden acceder en una sola vista [Sistema.Almacenamiento.Almacén - System.Storage.Store].[ItemFragment - Fragmentodelítem]. La siguiente tabla describe las columnas de una vista de fragmento de ítem global:

Nombre de Columna	Tipo	Descripción
ItemId - Identificación	[Sistema.Almacenamiento.Almacén].	Identificación

ijb

del ítem	Identificación del ítem	del ítem del ítem propietario
SetId - Identificación del Conjunto	[Sistema.Almacenamiento.Almacén]. Identificación del Conjunto	La utilización del fragmento del ítem
FragmentId - Identificación del Fragmento	[Sistema.Almacenamiento.Almacén]. Identificación del Fragmento	La identificación de la instancia del fragmento
TypeId – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación del tipo de instancia de Fragmento de ítem de mayor derivación
ItemFragment – Fragmento del ítem	[Sistema.Almacenamiento.Almacén]. Fragmento del ítem]	Instancia del tipo de Fragmento del ítem
EntityState - Estado de la Entidad	[Sistema.Almacenamiento.Almacén]. Estado de la Entidad	El estado de información para el enlace
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar información

		de rastreo
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta del Bin	La manija de la ruta al ítem que es la fuente del enlace

Tabla 4

[0056] Cada almacén suministra una vista de extensión global llamada [Sistema.Almacenamiento.Almacén - System.Storage.Store].[Extensión de Ítem - ItemExtension]. Las instancias de todos los tipos de extensión se pueden acceder por esta vista. La siguiente tabla describe las columnas en la vista de extensión global:

Nombre de Columna	Tipo	Descripción
ItemId - Identificación del Ítem	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem de la extensión
TypeId – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación esquematizado
ItemExtension – Extensión del Ítem	[Sistema.Almacenamiento.Almacén]. Extensión del Ítem	Instancia del tipo de Extensión
EntityState - Estado de la Entidad	[Sistema.Almacenamiento.Almacén]. Estado de la Entidad	Estado de la información para la extensión
ObjectSize –	Ent Grande	Tamaño de la

45

Tamaño de Objeto		extensión del objeto (columna persistida)
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar información de rastreo
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta	La manija de la ruta al ítem que es la propietaria de la extensión

Tabla 5

[0057] Cada tipo de extensión se mapea a un tipo de vista. Cada vista de extensión tipo se identifica en un almacén de sistema de almacenamiento de archivos utilizando la convención del nombre [<nombre de esquema - schema name>.Almacén. Store].[<nombre del tipo de la extensión - extension type name >]. La siguiente tabla describe las columnas de una vista de tipo de extensión:

Nombre de Columna	Tipo	Descripción
ItemId - Identificación del Ítem	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem de la extensión
TypeId – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación esquematizado
ItemExtension – Extensión del Ítem	Tipo de extensión	Instancia del tipo de

ad

		Extensión
ObjectSize – Tamaño de Objeto	Ent Grande	Tamaño de la extensión del objeto (columna persistida)
EntityState - Estado de la Entidad	[Sistema.Almacenamiento.Almacén]. Estado de la Entidad	Estado de la información para la extensión
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar información de rastreo
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta	La manija de la ruta al ítem que es la propietaria de la extensión

Tabla 6

En un sistema de almacenamiento de archivos, se puede almacenar una instancia tipo incluida, en las instancias de tipo de entidad. Se debe apreciar y comprender que se pueden acceder realizando una búsqueda de la vista de búsqueda apropiada.

[0058] Todos los ítems quedan almacenados en una única tabla de ítem llamada [Sistema.Almacenamiento.Almacén - System.Storage.Store].[Tabla!-Table! Ítem - Item]. La única llave en la tabla siguiente es Identificación del ítem - ItemId:

Nombre de Columna	Tipo	Descripción
ItemId - Identificación del Ítem	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem
TypeId – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación esquemático
NamespaceName – Nombre Espacio Nombre	Nvarchar(255)	El nombre único del Ítem
ContainerId – Contenedor de la identificación	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	La identificación del contenedor del ítem
Item	Item type	El ítem
EntityState – Estado de la Entidad	[Sistema.Almacenamiento.Almacén]. Estado de la Entidad	El Estado de la Entidad udt que contiene meta información sobre el ítem.
SDId – identificación SD	Ent	Uso interno únicamente (utilizado por el subsistema de seguridad)
SDLastUpdateLocal TS – Última actualización del TS Local SD	Ent Grande	El sello con la hora del último cambio de identificación

		SD. Utilizado para cambiar rastreo.
ChangeInformation - Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar información de rastreo
ItemSyncMetadata - Sincronizar Metadata de la Información	[Sistema.Almacenamiento.Almacén]. Sincronizar Metadata de la Información	Sincronizar metadata para identificaciones globales
TombstoneStatus - Estado Tombstone	Ent	Estado Tombstone
LastUpdateLocalTS - Última actualización del TS Local	Ent Grande	Uso interno únicamente (utilizado para indexar). Esta columna se mapea a Cambiar Información sobre Última actualización del TS Local - ChangeInformation.LastUpdateLocalTS. Se debe mapear para poder crearle un



		índice.
MaxOrd	Ent	Uso interno únicamente (usado para computar la Manija de Ruta para los nuevos ítems agregados a este contenedor de ítems)
TypePath – Tipo de Ruta	Hierarchical_Type_Id – Tipo de identificación jerárquica	Uso interno únicamente (columna persistida)
ObjectSize – Tamaño de Objeto	Ent Grande	Tamaño del objeto de ítem (columna persistida)
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta de Bin	La Manija de Ruta a este ítem
PromotionStatus – Estado de Promoción	Ent	El estado de promoción para el ítem
LastAccessTime – Última Hora de Acceso	Fecha Hora	La última hora de acceso para la secuencia de

		archivo asociada al ítem.
StreamSize – Tamaño de la Secuencia	Ent Grande	El tamaño de la secuencia asociada con el ítem
AllocationSize – Tamaño de la Asignación	Ent Grande	El tamaño asignado para la secuencia asociada con el ítem

Tabla 7

[0059] Los índices en la tabla de ítems se describen a continuación:

Columnas	Unico	Agrupado	Columnas Incluidas
TypePath, ItemId – Tipo de Ruta, Identificación del ítem	Sí	Sí	
ItemId - Identificación del ítem	Sí	No	TombstoneStatus – Estado Tombstone, TypePath – Tipo de Ruta, SDId – Identificación de SD
PathHandle - Manija de la Ruta	Sí	No	SDId – Identificación de SD, TombstoneStatus – Estado Tombstone
ContainerId, NamespaceName –	Sí	No	SDId – Identificación de SD, TombstoneStatus – Estado

51

identificador del Contenedor, Nombre Espacio Nombre			Tombstone
LastUpdateLocalTS - Última actualización del TS Local	No	No	SDId -- Identificación de SD, TombstoneStatus -- Estado Tombstone, PathHandle -- Maija de Ruta

Tabla 8

[0060] La totalidad de los enlaces se almacenan es la tabla de enlaces, la cual se denomina [Sistema.Almacenamiento.Almacén - System.Storage.Store].[Tabla! - Table! Enlace - Link]. La única llave en la tabla siguiente es Identificación del Ítem - ItemId, Identificación del Enlace - LinkId:

Nombre de Columna	Tipo	Descripción
SourceRef – Ref de la Fuente	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem del ítem fuente
LinkId – Identificación del Enlace	[Sistema.Almacenamiento.Almacén]. Identificación del Enlace	Identificación del Enlace
TargetRef – Ref del Objetivo	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem del ítem objetivo
TypeId – Tipo de identificación	[Sistema.Almacenamiento.Almacén]. Tipo de identificación	Tipo de Identificación esquematizado
TypePath – Tipo	Tipo de identificación jerárquica	Uso interno

de Ruta		Únicamente (columna persistida)
Link – Enlace	Tipo de enlace	El objeto enlace
EntityState – Estado de la Entidad	[Sistema.Almacenamiento.Almacén]. Estado de la Entidad	Estado de información sobre el enlace
SDId – Identificación de SD	Ent	Uso interno únicamente
TombstoneStatus – Estado de Tombstone	Ent	Estado Tombstone
ObjectSize – Tamaño de objeto	Ent Grande	Tamaño del enlace objeto (columna persistida)
ChangeInformatio n – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar la información de rastreo
LinkSyncMetadat a – Metadata de Sincronización del Enlace	[Sistema.Almacenamiento.Almacén]. Metadata de Sincronización del Enlace	Sincronización de metadata
LastUpdateLocal TS - Última actualización del TS Local	Ent Grande	Uso interno únicamente (sólo utilizado para indexar). Esta columna se

		mapea a Cambiar Información sobre Última actualización del TS Local - ChangeInformation.LastUpdateLocalITS. Se debe mapear para poder crearle un índice.
PathHandle - Manija de la Ruta	[Sistema.Almacenamiento.Almacén]. Manija de la Ruta de Bin	La Manija de Ruta a este ítem que es la fuente del enlace

Tabla 9

[0061] Los índices en la tabla de Índices se describen a continuación:

Columnas	Unico	Agrupado	Columnas Incluidas
SourceRef – Ref de la Fuente, LinkId – Identificación del Enlace	Si	Si	
TypePath – Tipo de Ruta	No	No	Identificación de SD, Estado Tombstone
SourceRef – Ref de la Fuente, TypePath - Tipo de Ruta	No	No	Identificación de SD, Estado Tombstone
PathHandle – Manija de	No	No	Identificación de SD, Estado

~~54.~~

Ruta			Tombstone
TargetRef – Ref Objetivo	No	No	Identificación de SD, Estado Tombstone
LastUpdateLocalTS - Última actualización del TS Local	No	No	Identificación de SD, Estado Tombstone , Manija de Ruta

Table 10

[0062] La totalidad de las Extensiones de Entidad - EntityExtensions se almacenan en una única tabla denominada: [Sistema.Almacenamiento.Almacén].[Tabla] - Tablel Extensión de Ítem - ItemExtension}. La siguiente tabla describe la tabla de extensión de ítem:

Nombre de Columna	Tipo	Descripción
ItemId - Identificación del Ítem	[Sistema.Almacenamiento.Almacén]. Identificación del Ítem	Identificación del Ítem de la extensión
TypeId – Tipo de identificación	[Sistema.Almacenamiento.Almacén]. Tipo de identificación	Tipo de Identificación esquematizado
TypePath – Tipo de Ruta	Tipo de identificación jerárquica	Uso interno únicamente (columna persistida)
ItemExtension – Extensión del ítem	[Sistema.Almacenamiento.Almacén]. Extensión del Ítem	El objeto de Extensión del ítem
EntityState –	[Sistema.Almacenamiento.Almacén].	Estado de la

fs

Estado de la Entidad	Estado de la Entidad	Entidad udt que contiene meta información sobre la Extensión de la Entidad.
SDId - Identificación de SD	Ent	Uso interno únicamente
TombstoneStatus - Tombstone	Ent	Estado Tombstone
ObjectSize – Tamaño de Objeto	Ent Grande	Tamaño del Objeto de la Extensión de Entidad (columna persistida)
ChangeInformation – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar la información de rastreo
LastUpdateLocal TS - Última actualización del TS Local	Ent Grande	Uso interno únicamente (sólo utilizado para indexar). Esta columna se mapea a Cambiar Información sobre Última actualización del TS Local -

36

		ChangeInformatio n.LastUpdateLoca ITS. Se debe mapear para poder crearle un índice.
PathHandle – Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta de Bin	La Manija de Ruta a este ítem que es la fuente del enlace Extensión de la Entidad

Tabla 11

[0063] Los índices de la tabla de Extensión de Ítems se describen a continuación:

Columnas	Unico	Agrupado	Columnas Incluidas
TypePath – Tipo de Ruta, Identificación de ítem - ItemId, TypeId - Tipo de identificación	Sí	Sí	
ItemId – Identificación de ítem, TypeId – Tipo de identificación	No	No	Identificación de SD, Estado Tombstone, Manija de Ruta
PathHandle - Manija de Ruta	Sí	No	Identificación de SD, Estado Tombstone
LastUpdateLocalTS - Última actualización del TS Local	No	No	Identificación de SD, Estado Tombstone, Manija de Ruta

Tabla 12

[0064] La totalidad de las instancias de tipo de Fragmento de Ítem se almacenan en una única tabla denominada [Sistema.Almacenamiento.Almacén].[Tabla - Table| Fragmento de Ítem - ItemFragment]. La siguiente tabla ilustra el tipo de Fragmento de Ítem:

Nombre de Columna	Tipo	Descripción
Itemid - Identificación de ítem	[Sistema.Almacenamiento.Almacén].l Identificación de ítem	Identificación de ítem del ítem propietario
Setid - Identificación del conjunto	[Sistema.Almacenamiento.Almacén]. Identificación del conjunto	Utilización de fragmento de ítem
Fragmentid - Identificación del fragmento	[Sistema.Almacenamiento.Almacén]. Identificación del fragmento	Identificación de la instancia del fragmento
ItemFragment – Fragmento del ítem	[Sistema.Almacenamiento.Almacén]. Fragmento del ítem	El objeto de Fragmento del ítem
Typeid – Tipo de Identificación	[Sistema.Almacenamiento.Almacén]. Tipo de Identificación	Tipo de Identificación esquematizado
TypePath – Tipo de Ruta	Tipo de identificación jerárquica	Uso interno únicamente (columna persistida)
EntityState –	[Sistema.Almacenamiento.Almacén].	Estado de la

Estado de la Entidad	Estado de la Entidad	Entidad udt que contiene meta información sobre el fragmento
SDId – Identificación de SD	Ent	Uso interno únicamente
TombstoneStatus – Estado Tombstone	Ent	Estado Tombstone
ObjectSize – Tamaño del Objeto	Ent Grande	Tamaño del Objeto de Fragmento de ítem (columna persistida)
ChangeInformatio n – Cambiar Información	[Sistema.Almacenamiento.Almacén]. Cambiar Información	Cambiar la información de rastreo
LastUpdateLocal TS - Última actualización del TS Local	Ent Grande	Uso interno únicamente (sólo utilizado para indexar). Esta columna se mapea a Cambiar Información sobre Última actualización del TS Local - ChangeInformatio

		n.LastUpdateLoca ITS. Se debe mapear para poder crearle un índice.
PathHandle - Manija de Ruta	[Sistema.Almacenamiento.Almacén]. Manija de Ruta de Bin	La Manija de Ruta al ítem propietario

Tabla 13

[0065] Los índices de la tabla de Fragmento de Ítem se describen a continuación:

Columnas	Único	Agrupado	Columnas Incluidas
TypePath – Tipo de Ruta, ItemId – Identificación de Ítem, SetId – Identificación del Conjunto, FragmentId – Identificación del Fragmento	Sí	Sí	
ItemId - Identificación de Ítem, SetId – Identificación del Conjunto, Fragmentad – Identificación del Fragmento	Sí	No	Identificación de SD, Estado Tombstone, Manija de Ruta
PathHandle - Manija de Ruta	Sí	No	Identificación de SD, Estado Tombstone
LastUpdateLocalTS - Última actualización del TS Local	No	No	Identificación de SD, Estado Tombstone, Manija de Ruta

CO.

Tabla 14

[0066] Las Figs. 9-10 ilustran metodologías de acuerdo con el sujeto del invento. Para simplicidad en la explicación, las metodologías se ilustran y se describen como una serie de actos. Se entiende que el sujeto del invento no se encuentra limitado por los actos ilustrados y/o por el orden de los mismos; por ejemplo, los actos pueden ocurrir en múltiple orden y/o de manera concurrente, junto con otros actos no descritos en este documento. Adicionalmente, no todos los actos ilustrados pueden ser requeridos para implementar las metodologías según el sujeto del invento. Además, aquéllos conocedores en la materia apreciarán que las metodologías pueden ser representados de manera alternativa como una serie de estados interrelacionados vía un diagrama de estado o eventos.

[0067] La Fig. 9 ilustra una metodología 900 que facilita el almacenamiento de una instancia de tipo que mapea a un modelo de información y/o suministra una búsqueda. El modelo de información puede ser un modelo de almacenamiento de información de archivos que permite el almacenamiento, la búsqueda y el reracionamiento de la información. Un tipo de información puede ser, pero no se limita a, un documento, una imagen, un video, un contacto, un mensaje, un correo electrónico, un clip de audio, etc. Estos tipos de información (ej., unidades de información) se pueden representar como instancias de tipos complejos que son parte de un sistema tipo que apoya la herencia, la cual permite a los objetos heredar atributos y/o comportamientos de otros objetos. También se debe tener en cuenta que dicha herencia puede ser considerada una estructura y/o formato jerárquico.

[0068] En el numeral de referencia 902, la información se puede recibir, la información puede incluir un tipo, un criterio, un esquema, un criterio de búsqueda, La instancia tipo se puede almacenar para mapear un modelo de información

(ej., un modelo de información de almacenamiento de archivos) en el numeral de referencia 904. Por ejemplo, el modelo puede mapear tipos descritos en el esquema a los tipos definidos por el usuario y objetos de la base de datos. El mapeo de almacenamiento puede describir los objetos de la base de datos que se generan basados en una definición de esquema y en la forma cómo las instancias tipo descritos en el esquema se almacenan y/o acceden. En un ejemplo, la estructura de la base de datos puede ser diseñada de manera tal que las instancias tipo y las reglas para mapear las declaraciones tipo en objetos de base de datos se puedan proporcionar.

[0069] En el numeral de referencia 906, se puede invocar una búsqueda para satisfacer por lo menos un criterio para encontrar los ítems, documentos y contactos de manera eficiente y efectiva. Por ejemplo, el almacenamiento de la instancia tipo que se relaciona con los tipos de información puede utilizar una búsqueda para suministrar por lo menos uno de entre estos: 1) encontrar por lo menos un ítem en el modelo de información de almacenamiento de archivos 102 que cumpla con ciertos criterios; 2) encontrar por lo menos un documento en modelo de información de almacenamiento de archivos 102 que cumpla con un criterio particular; y 3) encontrar por lo menos un contacto (ej., incluyendo un individuo, una organización y un grupo) que cumpla con ciertos criterios.

[0070] La Fig. 10 ilustra una metodología 1000 que facilita el almacenamiento de una instancia tipo y/o proporcionar una búsqueda. En el numeral de referencia 1002, la información se obtiene y/o recibe. Dicha información puede incluir un tipo, un esquema, un criterio de búsqueda, etc. En el 1004, un motor de base de datos se puede utilizar para suministrar por lo menos un mecanismo para almacenar las instancias tipo y/o para las búsquedas. Por ejemplo, un motor de base de datos relacional se puede utilizar para proporcionar un almacenamiento relacional y una capacidad de búsqueda de relación. El motor de la base de datos relacional puede utilizar múltiples ítems de información

organizados como un conjunto de tablas descritas de manera formal. La información de dichas tablas se puede acceder y/o reagrupar de múltiples maneras sin requerir la reorganización de tablas de bases de datos. Adicionalmente, la base de datos relacional se puede extender fácilmente, agregando nuevas categorías sin modificar las aplicaciones existentes y/o a la información.

[0071] En el numeral de referencia 1006, se almacena una instancia tipo que mapea a un modelo de información, por el cual dicho modelo de información puede ser un sistema de almacenamiento de archivo. El mapeo del almacenamiento describe los objetos de la base de datos que son creados basados en el esquema y en cómo las instancias tipo descritos en dicho esquema se almacenan y/o son accedidos. El mapeo puede ser de los tipos descritos en el esquema, por el que dicho mapeo es para tipos definidos por el usuario y objetos de la base de datos. En el numeral de referencia 1008, se puede realizar una búsqueda para encontrar por lo menos uno de entre un ítem, un documento y un contacto. La búsqueda se puede utilizar para realizar una búsqueda basada al menos en parte en unos criterios. Adicionalmente, en el numeral de referencia 1010, se puede emplear una vista para exhibir las instancias tipo. Por ejemplo, el tipo de sistema puede ser jerárquico en su estructura, y la vista puede ser generada para exhibir cualquier instancia del tipo particular. En otras palabras, dada la jerarquía, la vista está asociada con un tipo dado que proyecta un subconjunto de los tipos particulares de la vista asociada con el tipo base.

[0072] Para proporcionar contexto adicional para la implementación de múltiples aspectos del sujeto del invento, es el objetivo de las Figs. 11-12 y de la siguiente discusión proporcionar una descripción breve y general de un ambiente de cómputo adecuado en el cual los múltiples aspectos del sujeto del invento se pueda implementar. Aunque el invento ha sido previamente descrito en el contexto general de instrucciones ejecutables por el computador en un programa

de cómputo que opera en un computador local y/o remoto, aquéllos conocedores del tema apreciarán que el invento también puede ser implementado en conjunto con otros módulos de programa. En general, los módulos de programa incluyen rutinas, programas, componentes, estructuras de información, etc., que realizan tareas específicas y/o implementan tipos de información abstractos particulares.

[0073] Adicionalmente, aquéllos conocedores del tema apreciarán que los métodos inventivos se pueden realizar con otras configuraciones del sistema de cómputo, incluyendo sistemas de cómputo de procesador único o de procesador múltiple, mini-computadores, procesadores centrales, así como computadores personales, instrumentos manuales, artículos electrónicos a base de microprocesador y/o programables, entre otros, cada uno de los cuales puede comunicar de manera operativa con uno o más instrumentos asociados. Los aspectos ilustrados del invento también se pueden realizar en ambientes de cómputo distribuidos en los que ciertas tareas se efectúan por medio de instrumentos de procesamiento remoto que están conectados en una red de comunicaciones. Sin embargo, algunos, si no la totalidad de los aspectos del invento se pueden efectuar en computadores independientes. En un ambiente de cómputo distribuido, los módulos de programa se pueden ubicar en herramientas de almacenamiento de memoria ya sea locales y/o remotos.

[0074] La Fig. 11 es un diagrama esquemático de bloque de un ejemplo de ambiente de cómputo 1100 con el que el sujeto del invento puede interactuar. El sistema 1100 incluye uno o más cliente (s) 1110. Estos clientes 1110 pueden ser hardware y/o software (ej., secuencias, procesos, instrumentos de cómputo). El sistema 1100 también incluye uno o más servidores 1120. Dichos servidores 1120 pueden ser hardware y/o software (ej., secuencias, proceso, instrumentos de cómputo). Los servidores 1120 pueden albergar secuencias para realizar transformaciones utilizando el sujeto del invento, por ejemplo.

[0075] Una posible comunicación entre un cliente 1110 y un servidor 1120 puede ser en la forma de un paquete de información adaptado para ser transmitido entre dos o más procesos de cómputo. El sistema 1100 incluye una estructura de comunicación 1140 que puede ser utilizada para facilitar las comunicaciones entre los clientes 1110 y los servidores 1120. Los clientes 1110 se conectan de forma operativa con uno o más almacenes de información de cliente 1150 que pueden ser empleados para almacenar la información local de los clientes 1110. De manera similar, los servidores 1120 están conectados de manera operativa a uno o más almacenes de información del servidor 1130 que puede ser utilizados para almacenar información local a los servidores 1140.

[0076] En la Fig. 12, se hace referencia a un ejemplo de ambiente 1200 para implementar varios aspectos del invento, que incluye un computador 1212. Dicho computador 1212 incluye una unidad de procesamiento 1214, un sistema de memoria 1216, y un sistema bus 1218. Dicho sistema bus 1218 acopla componentes del sistema, incluyendo pero sin limitarse al sistema de memoria 1216 a la unidad de procesamiento 1214. Dicha unidad de procesamiento 1214 puede ser cualquiera de múltiples procesadores disponibles. Los microprocesadores duales y otras arquitecturas de multiprocesador también pueden ser empleados como unidad de procesamiento 1214.

[0077] El sistema bus 1218 puede ser cualquiera de los múltiples tipos de estructura incluyendo la memoria bus o controlador de memoria, un bus periférico o externo, y/o bus local utilizando cualquier variedad disponible de arquitectura bus, incluyendo pero sin limitarse a Arquitectura Industrial Estándar - Industrial Standard Architecture (ISA), Arquitectura de Micro-Canal - Micro-Channel Architecture (MSA), ISA Extensiva - Extended ISA (EISA), Instrumentos Electrónicos Inteligentes - Intelligent Drive Electronics (IDE), Bus Local VESA - VESA Local Bus (VLB), Componente Periférico de Interconexión - Peripheral Component Interconnect (PCI), Tarjeta Bus - Card Bus, Bus Serial Universal -

Universal Serial Bus (USB), Puerto Avanzado de Gráficos - Advanced Graphics Port (AGP), Asociación Bus de Tarjetas de Memoria de Computadores Personales - Personal Computer Memory Card International Association bus (PCMCIA), Firewire (IEEE 1394), e Interfase de Sistemas de Pequeños Computadores - Small Computer Systems Interface (SCSI).

[0078] El sistema de memoria 1216 incluye una memoria volátil 1220 y una memoria no volátil 1222. El sistema básico de entrada/salida - basic input/output system (BIOS), que contiene las rutinas básicas para transferir información entre elementos en el computador 1212, como durante el inicio, es almacenado en la memoria no volátil 1222. A manera de ilustración y no de limitante, la memoria no volátil 1222 puede incluir memoria de solo lectura - read only memory (ROM), ROM programable - programmable ROM (PROM), ROM programable electrónicamente - electrically programmable ROM (EPROM), ROM programable borrrable electrónicamente - electrically erasable programmable ROM (EEPROM), o memoria flash. La memoria volátil 1220 incluye memoria de acceso aleatoria - random access memory (RAM), la cual actúa como memoria caché externa. A manera de ilustración y no de limitante, la RAM está disponible en múltiples formas tales como RAM estática - static RAM (SRAM), RAM dinámica - dynamic RAM (DRAM), DRAM sincronizada - synchronous DRAM (SDRAM), SDRAM de velocidad de información doble - double data rate SDRAM (DDR SDRAM), SDRAM mejorada - enhanced SDRAM (ESDRAM), Synchlink DRAM (SLDRAM), Rambus direct RAM (RDRAM), RAM directo Rambus Dinámico - direct Rambus dynamic RAM (DRDRAM), y RAM Rambus dinámica - Rambus dynamic RAM (RDRAM).

[0079] El computador 1212 también incluye medios de almacenamiento de cómputo removibles/no-removibles, volátiles/no-volátiles. La Fig. 12 ilustra, por ejemplo, un almacenamiento en disco 1224. Dicho almacenamiento en disco 1224 incluye pero no se limita a instrumentos como la unidad de disco magnético,

66

unidad de disco flexible, unidad de cinta, unidad Jaz, unidad Zip, unidad LS-100, tarjeta de memoria flash, o tarjeta de memoria. Adicionalmente, el almacenamiento en disco 1224 puede incluir medios de almacenamiento de manera separada o en conjunto con otros medios de almacenamiento que incluyen, pero no se limitan a una unidad de disco óptico tales como una unidad de disco compacto ROM (CD-ROM), una unidad de disco en la que se puede grabar (CD-R Drive), una unidad de CD reescribible (CD-RW Drive) o una unidad de disco versátil digital ROM (DVD-ROM). Para facilitar la conexión de las unidades de almacenamiento en disco 1224 al sistema bus 1218, típicamente se utiliza una interfase removible o no-removible, como una interfase 1226.

[0080] Se puede apreciar que la Fig. 12 describe un software que actúa como un intermediario entre los usuarios y los recursos básicos de cómputo descritos en el ambiente de operación adecuado 1200. Este software incluye un sistema de operación 1228. Dicho sistema de operación 1228, el cual puede ser almacenado en un almacenamiento en disco 1224, actúa para controlar y asignar recursos del sistema de cómputo 1212. Las aplicaciones de sistema 1230 aprovechan la administración de recursos por el sistema de operación 1228 a través de módulos de programa 1232 y la información de programa 1234 almacenada ya sea en el sistema de memoria 1216 o en el almacenamiento en disco 1224. Se debe tener en cuenta que el sujeto del invento se puede implementar en múltiples sistemas de operación o en combinaciones de sistemas de operación.

[0081] Un usuario ingresa comandos o información al computador 1212 por los instrumentos de entrada 1236. Éstos incluyen pero no se limitan a punteros tales como el ratón, bola de mando, puntero indicador, tecla de tacto, teclado, micrófono, palanca de juegos, teclado de juegos, disco satelital, escáner, tarjeta de sintonización de TV, cámara digital, cámara digital de video, cámara web, entre otros. Estos y otros instrumentos de entrada se conectan a la unidad de

procesamiento 1214 por el sistema bus 1218 vía puertos de interfase 1238. Los puertos de interfase 1238 incluyen, por ejemplo, un puerto serial, un puerto paralelo, un puerto de juegos, y un bus serial universal - universal serial bus (USB). Los instrumentos de salida 1240 utilizan algunos de los mismos tipos de puertos que los instrumentos de entrada 1236. En consecuencia, por ejemplo, el puerto USB puede ser utilizado para proveer entrada al computador 1212, y para sacar información de dicho computador 1212 a un instrumento de salida 1240. Un adaptador de salida 1242 se proporciona para ilustrar que existen algunos instrumentos de salida 1240 tales como monitores, parlantes e impresoras, entre otros instrumentos de salida 1240, los cuales requieren de adaptadores especiales. Los adaptadores de salida 1242 incluyen, a manera de ilustración pero no de limitante, tarjetas de sonido y de vídeo que pueden proporcionar un medio de conexión entre el instrumento de salida 1240 y el sistema bus 1218. Se debe tener en cuenta que otros instrumentos y/o sistema de instrumentos proporcionan capacidades tanto de entrada como salida, tales como computadores remotos 1244.

[0082] El computador 1212 puede operar en un ambiente de red utilizando conexiones lógicas a uno o más computadores remotos, tales como los computadores remotos 1244. Éstos computadores remotos 1244 pueden ser un computador personal, un servidor, un enrutador, una red de PC, una estación de trabajo, un electrodoméstico basado en un microprocesador, un instrumento de par u otro nodo de red común, entre otros, y por lo general incluyen múltiples o la totalidad de los elementos descritos relativos al computador 1212. Para ser breves, sólo se ilustra un instrumento de almacenamiento de memoria 1246 con un computador remoto 1244. Dicho computador remoto 1244 está conectado lógicamente al computador 1212 a través de la interfase de red 1248 y luego están físicamente conectados vía conexión de comunicación 1250. La interfase de red 1248 incluye redes de comunicación alámbricas y/o inalámbricas, tales como redes de área local - local-area networks (LAN) y redes de área extensas - wide-

componentes deben corresponder, salvo que se indique lo contrario, a cualquier componente que realice la función especificada del componente descrito (ej., un equivalente funcional), a pesar de que no sea equivalente estructuralmente a la estructura mencionada, que realiza la función en los ejemplos aquí ilustrados como aspectos del invento. En cuanto a esto, también se debe tener en cuenta que el invento incluye un sistema y un medio reconocible por el computador con instrucciones ejecutables por el computador para la realización de las acciones y/o eventos de los múltiples métodos del invento.

[0086] Adicionalmente, aunque se haya mencionado una característica específica del invento con respecto a sólo una de múltiples implementaciones, dicha característica puede ser combinada con una o más características de las otras implementaciones como se desee y en la manera en que sea ventajoso para cualquier aplicación dada o específica. Adicionalmente, en la medida en que el término "incluye" e "incluyendo" y sus variaciones en la descripción detallada o en las reivindicaciones, es la intención que dicho término sea inclusive, de manera similar al término "comprende".

REIVINDICACIONES

Se reivindica lo siguiente:

1. Un sistema que facilita el mapeo de un modelo de información, el cual incluye:

Un modelo de información de almacenamiento de archivo que se utiliza para definir un esquema de mapeo; y

Un componente de almacenamiento tipo que mapea un objeto de esquema a un objeto de base de datos basado en el esquema de mapeo asociado, en el que se almacena una instancia de un tipo en el esquema de mapeo.

2. El sistema de la reivindicación 1, en el que la instancia tipo es por lo menos uno de: un documento; una imagen; música; un video; un contacto; un mensaje; una persona; una organización; un correo electrónico; un fax; una voz; y un clip de audio.

3. El sistema de la reivindicación 1, en el que el componente de almacenamiento tipo suministra una búsqueda para encontrar por lo menos uno de los siguientes: por lo menos un ítem en el sistema que satisface un criterio respectivo; por lo menos un documento en el sistema que satisface un criterio respectivo; y por lo menos un contacto que satisface un criterio respectivo.

4. El sistema de la reivindicación 3, en el que el contacto puede incluir por lo menos uno de los siguientes: un individuo; una organización; y un grupo.

5. El sistema de la reivindicación 1, que además incluye un componente relacional que utiliza por lo menos uno de entre un almacenamiento relacional y una capacidad de búsqueda relacional.

6. El sistema de la reivindicación 5, en el que el componente relacional utiliza una técnica de base de datos relacional que es una colección de ítems de información organizada como un conjunto de tablas descritas formalmente, en las cuales la información puede ser por lo menos uno accedido sin reorganización y reensamblada sin reorganización.
7. El sistema de la reivindicación 1, en el que el mapeo describe por lo menos uno de los objetos de la base de datos generados en el esquema y cómo la instancia de tipo descrita en dicho esquema se almacena y se accede.
8. El sistema de la reivindicación 1, en el que el tipo en el esquema mapea a una clase de tiempo de operación común - common language runtime (CLR) en un almacenamiento del modelo de información de almacenamiento de archivo.
9. El sistema de la reivindicación 8, en el que el tipo es por lo menos uno de los siguientes: un ítem; una extensión de ítem; un fragmento de ítem; y un enlace.
10. El sistema de la reivindicación 1, que además incluye una tabla que contiene por lo menos uno de los siguientes: 1) una columna que contiene por lo menos una instancia del tipo respectivo; y 2) una fila que contiene una representación serializada de la instancia de clase de CLR que representa la instancia tipo.
11. El sistema de la reivindicación 1, que además incluye un componente de vista que genera una vista que exhibe por lo menos una instancia del tipo, en la que la vista proyecta un subconjunto del respectivo tipo de vista asociada con el tipo base.
12. El sistema de la reivindicación 1, en el que el tipo contiene por lo menos uno de los siguientes: un tipo de jerarquía y un legado.

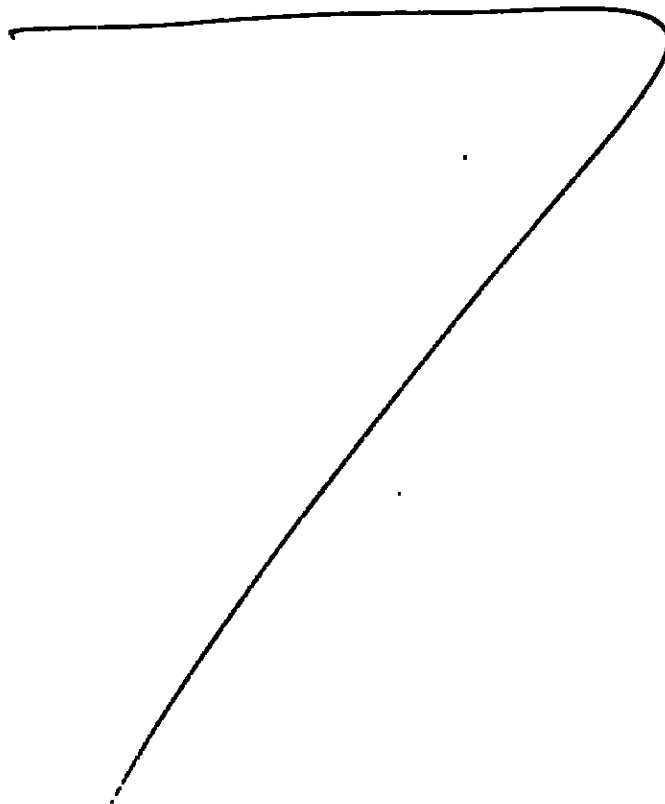
13. Un medio reconocible por el computador que ha almacenado los componentes del sistema de la reivindicación 1.
14. Un método implementado en el computador que facilita el mapeo de un modelo de información que incluye:
 - obtener un modelo de información de almacenamiento de archivo;
 - recibir por lo menos uno de entre un esquema, un tipo, un criterio y una información de búsqueda;
 - mapear un objeto de esquema a un objeto de base de datos basado en el esquema asociado al modelo de información de almacenamiento de archivo; y
 - almacenar una instancia de tipo del esquema.
15. El método de la reivindicación 14, que además incluye una búsqueda para encontrar por lo menos uno de los siguientes: por lo menos un ítem en el sistema que cumpla con ciertos criterios; por lo menos un documento en el sistema que cumpla con ciertos criterios; y por lo menos un contacto que cumpla con ciertos criterios.
16. El método de la reivindicación 14, que además incluye la utilización de un motor de base de datos relacional para proporcionar un almacenamiento relacional y una capacidad de búsqueda relacional.
17. El método de la reivindicación 14, que además incluye la generación de una vista que exhibe por lo menos una instancia de tipo, en la que la vista proyecta un subconjunto del tipo respectivo de la vista asociadas con el tipo base.
18. Un paquete de información que comunica entre un modelo de información de almacenamiento de archivo y un componente de almacenamiento tipo en el cual el paquete de información facilita el método de la reivindicación 14.

19. Un sistema implementado en el computador que facilita el mapeo de un modelo de información que incluye:

medios para recibir por lo menos uno de los siguientes: un esquema, un tipo, un criterio y un criterio de búsqueda; y

medios para mapear un objeto de esquema a un objeto de base de datos basado en un esquema asociado con el modelo de información de almacenamiento de archivo, en el que se almacena una instancia tipo.

20. El sistema de la reivindicación 11, en el que el componente de vista invoca a una interfase de usuario a interactuar entre un usuario y el componente de almacenamiento tipo.



COMPENDIO

El sujeto del invento proporciona un sistema y/o método que facilita mapear un modelo de base de datos a un objeto de base de datos. Un sistema de almacenamiento tipo puede utilizar un mapeo de almacenamiento de un modelo de información de almacenamiento de archivo. El mapeo puede describir el objeto de la base de datos creado basado al menos parcialmente en un esquema y en cómo las instancias de tipo descritas en el esquema se almacenan y/o se acceden. Adicionalmente, se puede proporcionar una búsqueda para encontrar por lo menos uno de entre un ítem, un documento, y/o un contacto que cumplan con por lo menos un criterio. El sistema de almacenamiento tipo puede recibir información, por el cual la información es por lo menos uno de entre un esquema, un modelo de información, un tipo, una búsqueda, un criterio de búsqueda vía una interfase para suministrar el almacenamiento y la búsqueda. Adicionalmente, el sistema de almacenamiento tipo puede generar una vista que exhibe por lo menos una instancia de tipo.



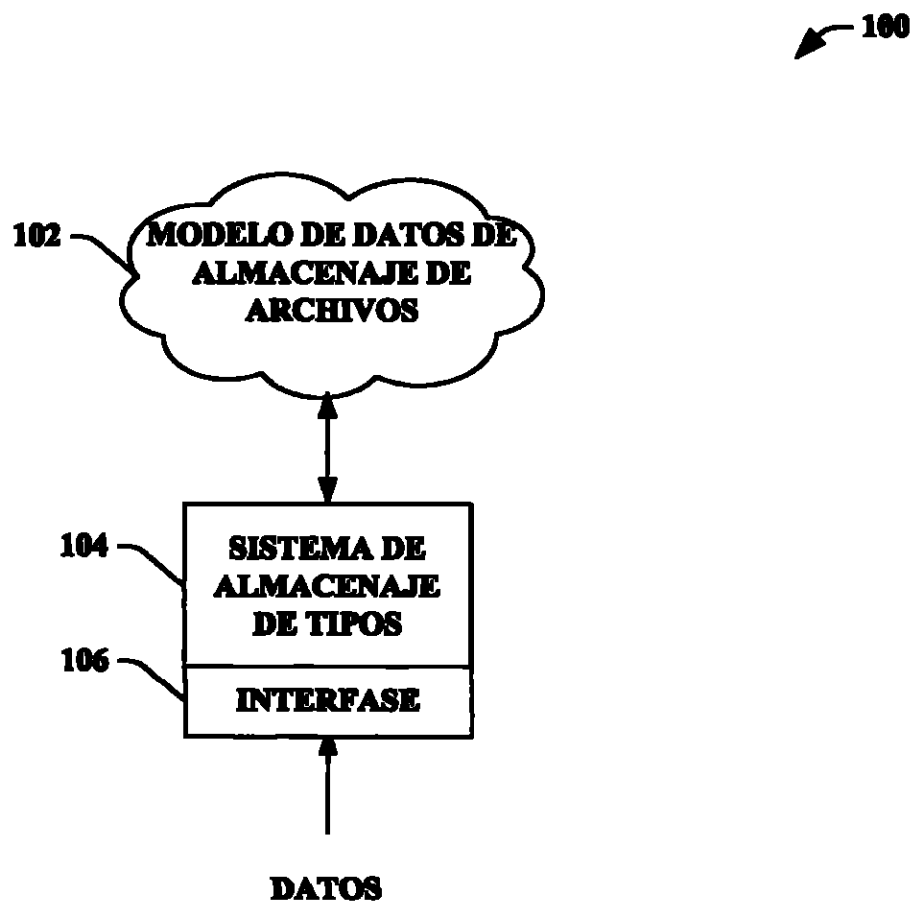


FIG. 1

76

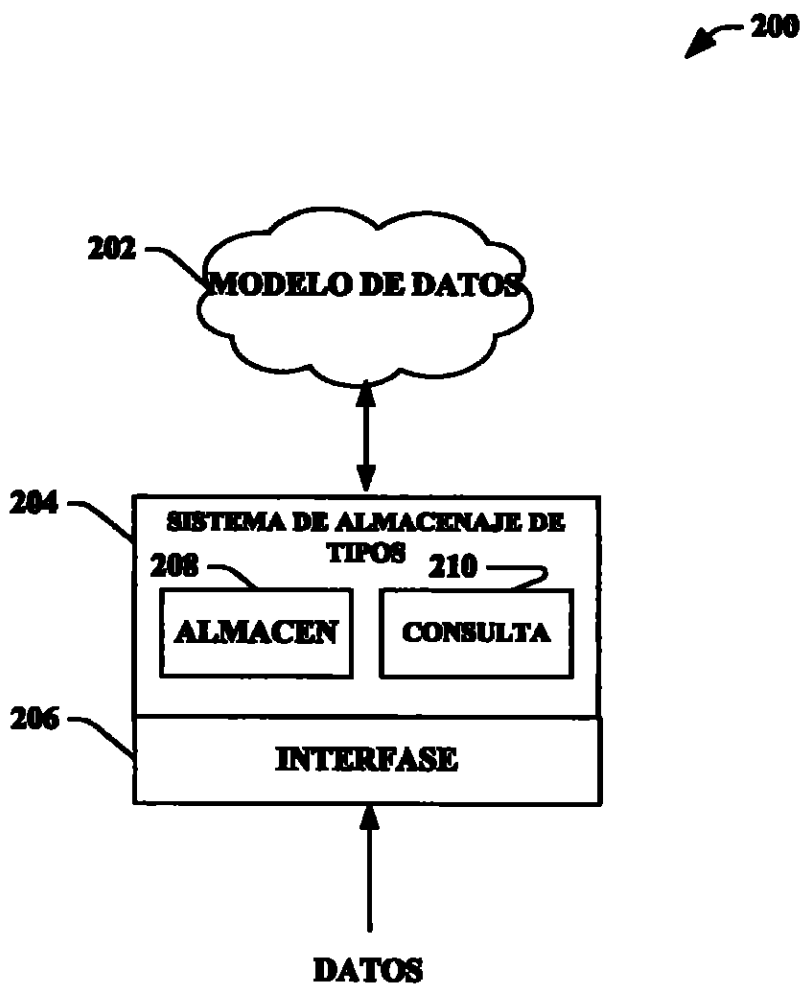


FIG. 2

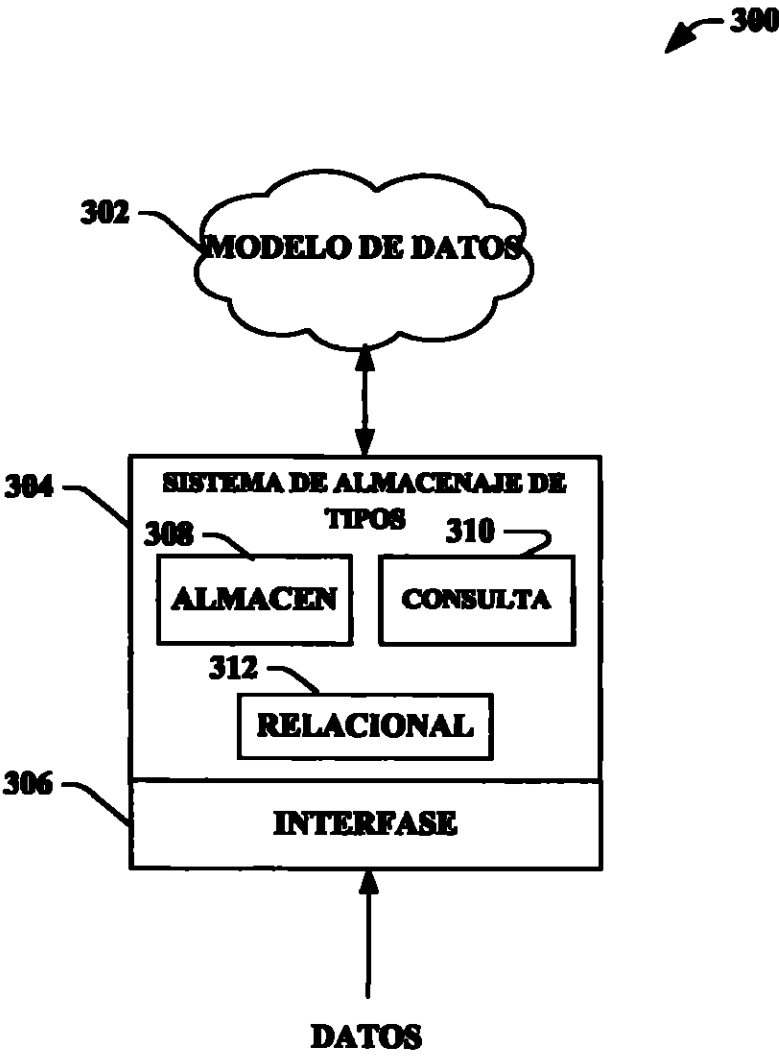


FIG. 3

78

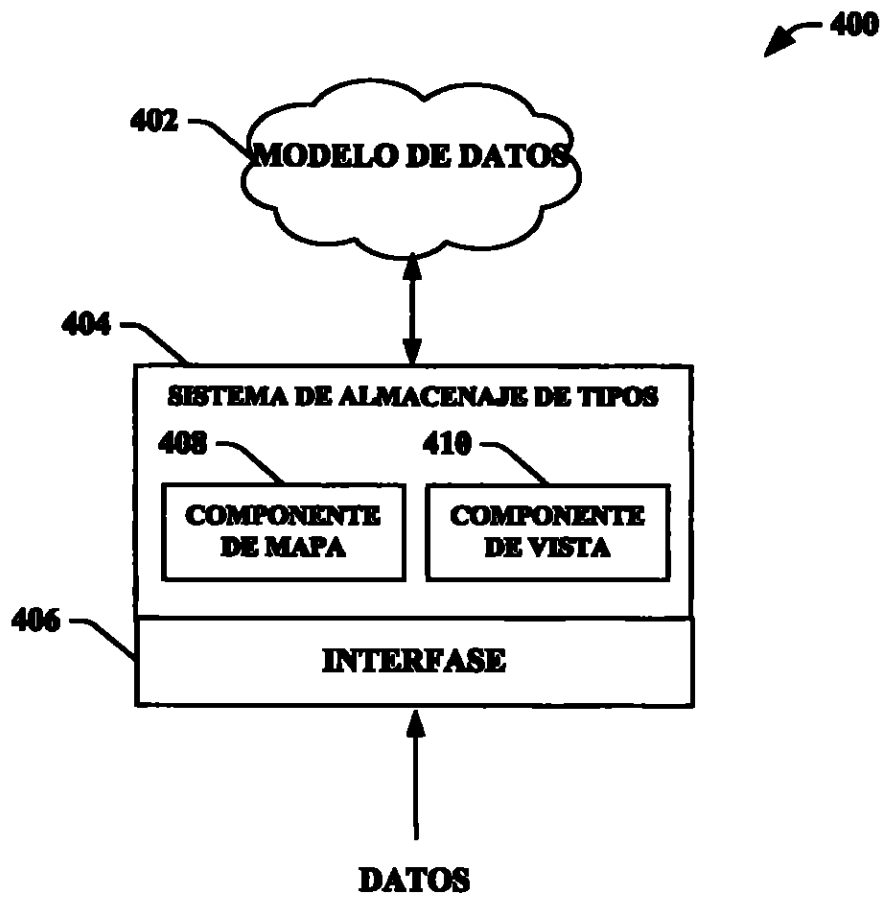


FIG. 4

79.

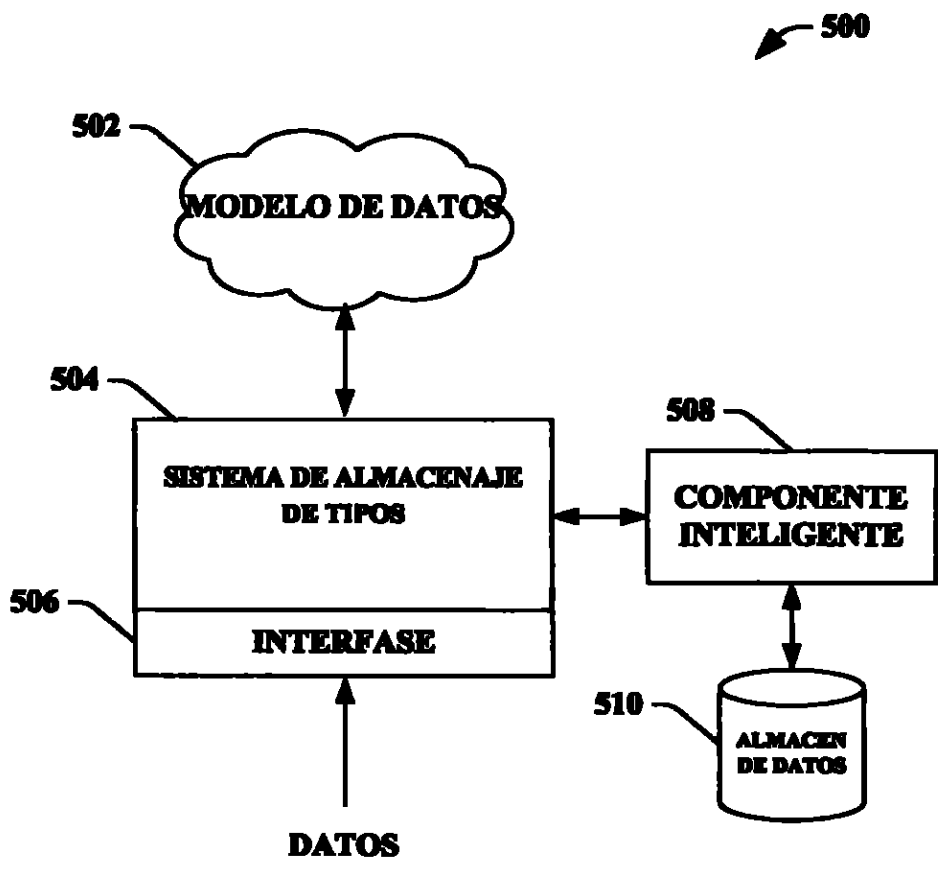


FIG. 5

80

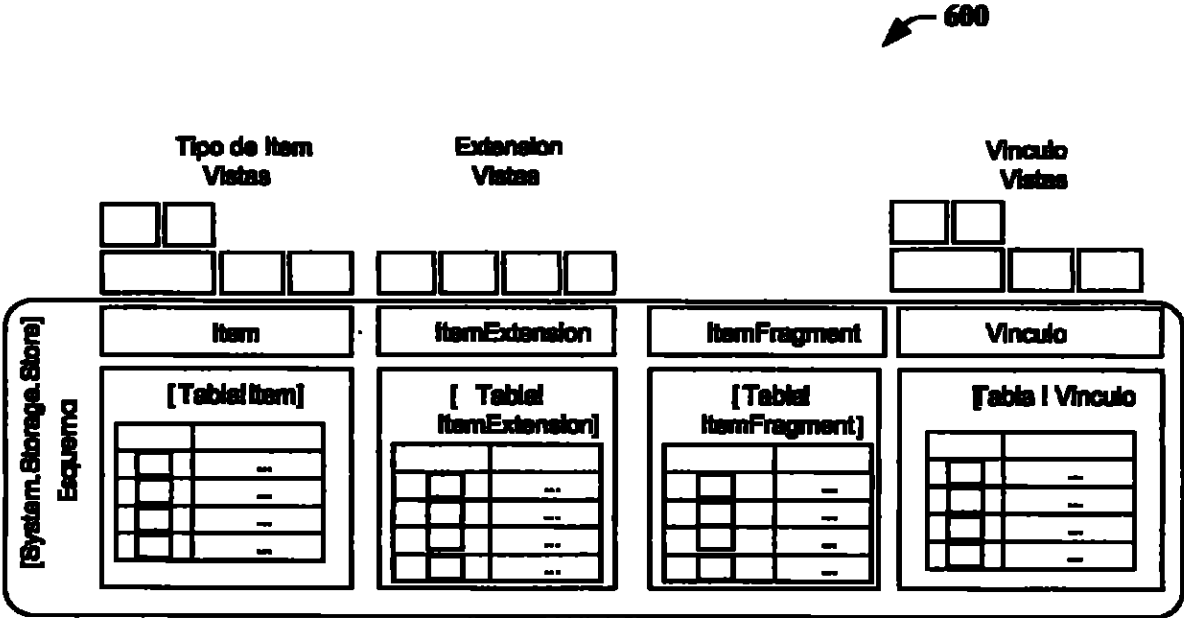


FIG. 6

34

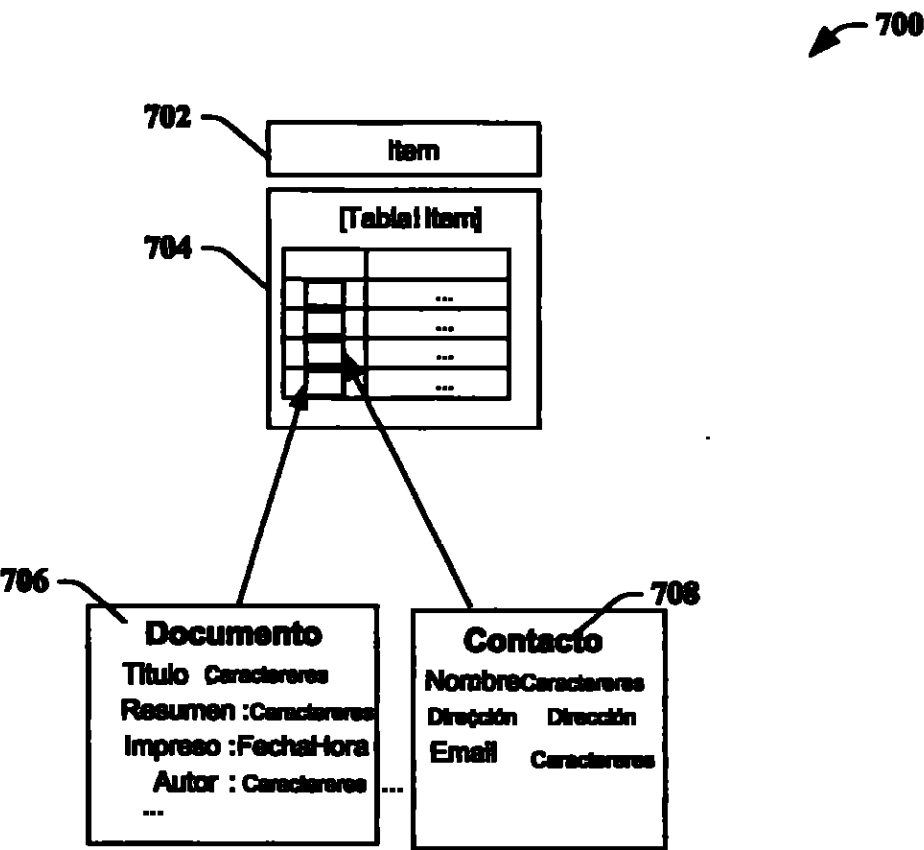


FIG. 7

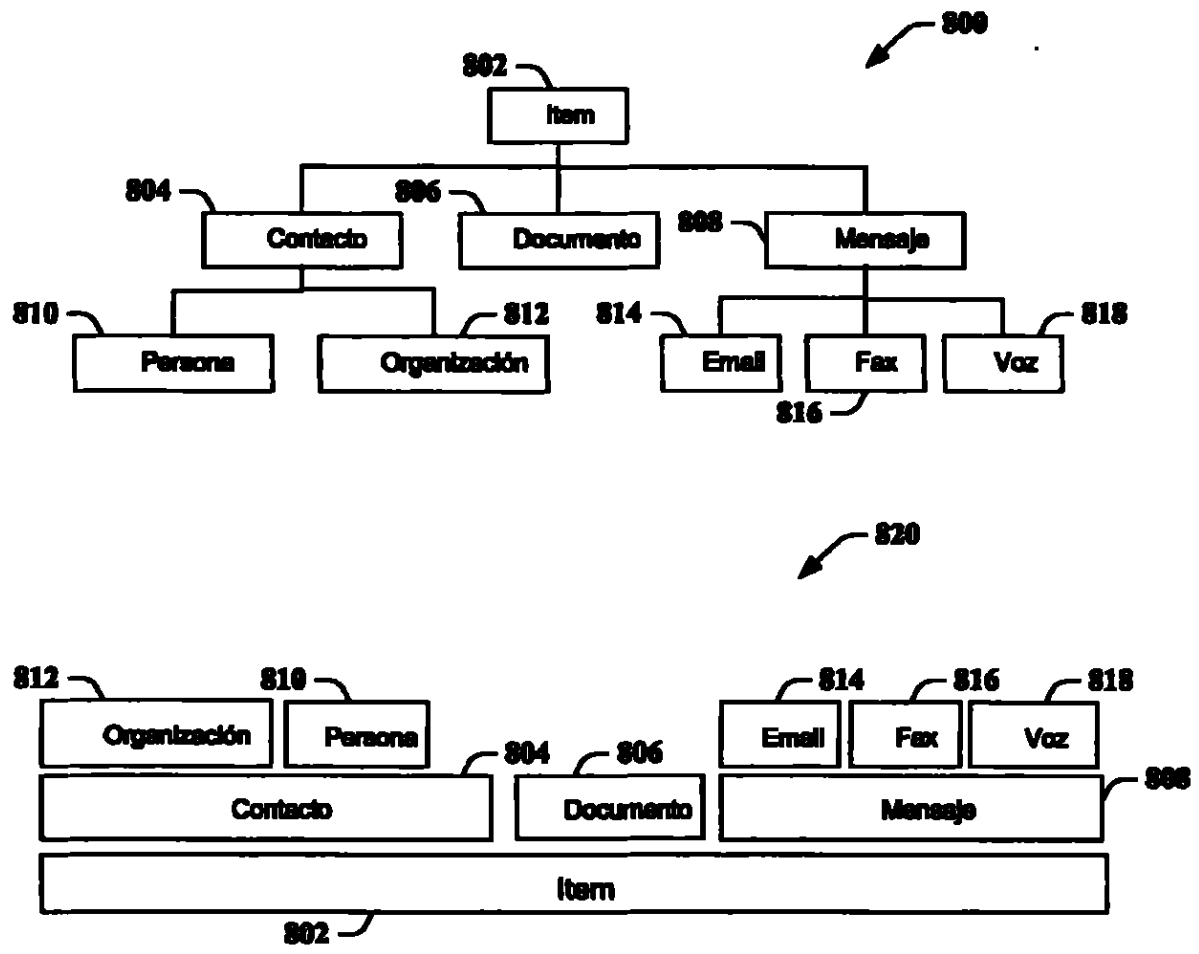


FIG. 8

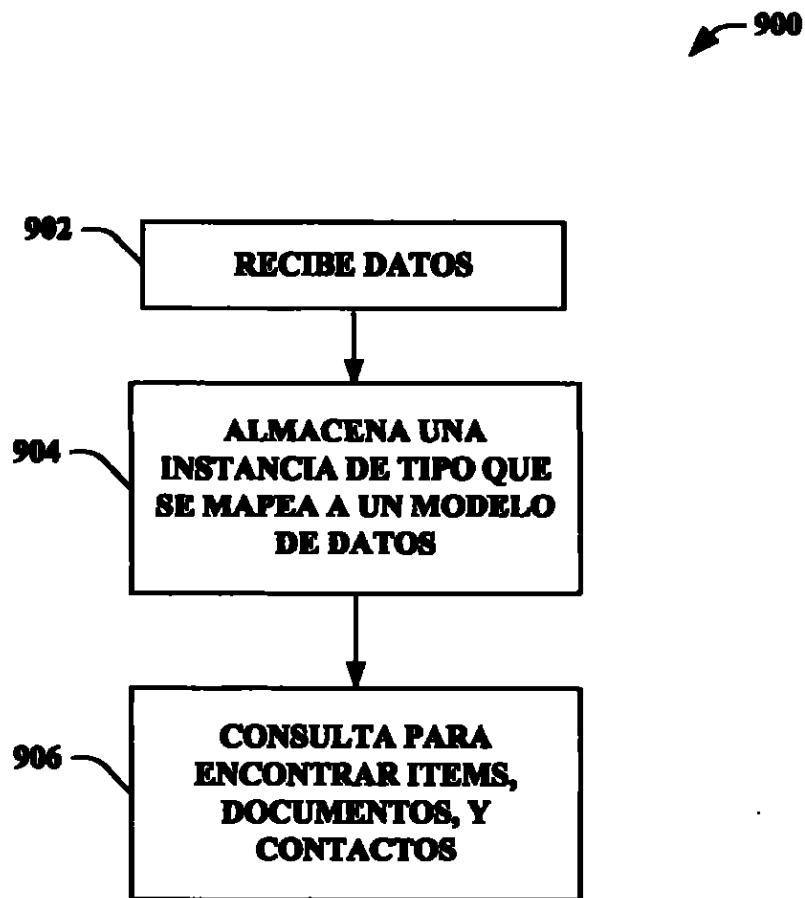


FIG. 9

1000

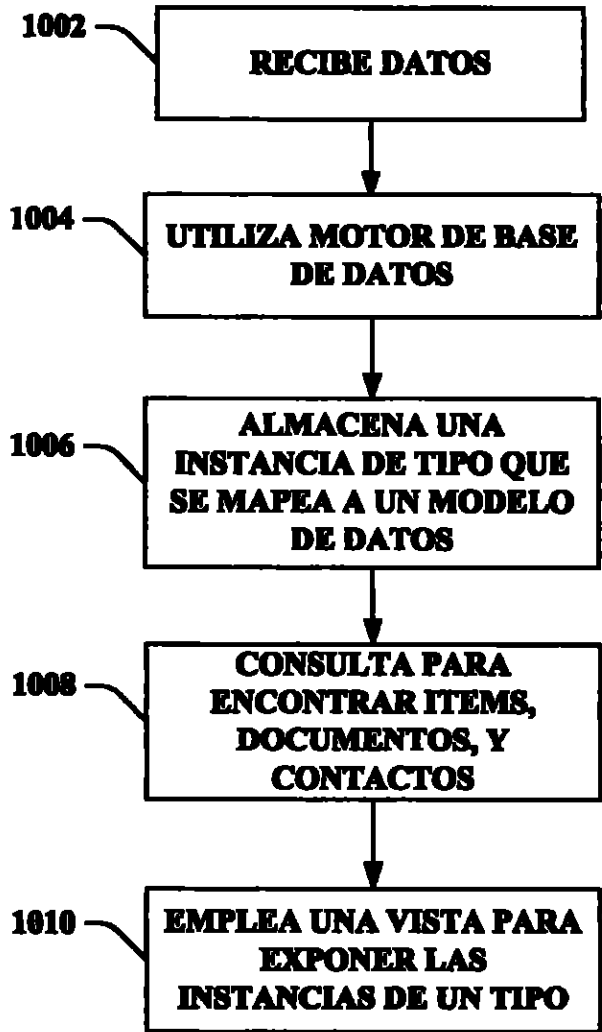


FIG. 10

85

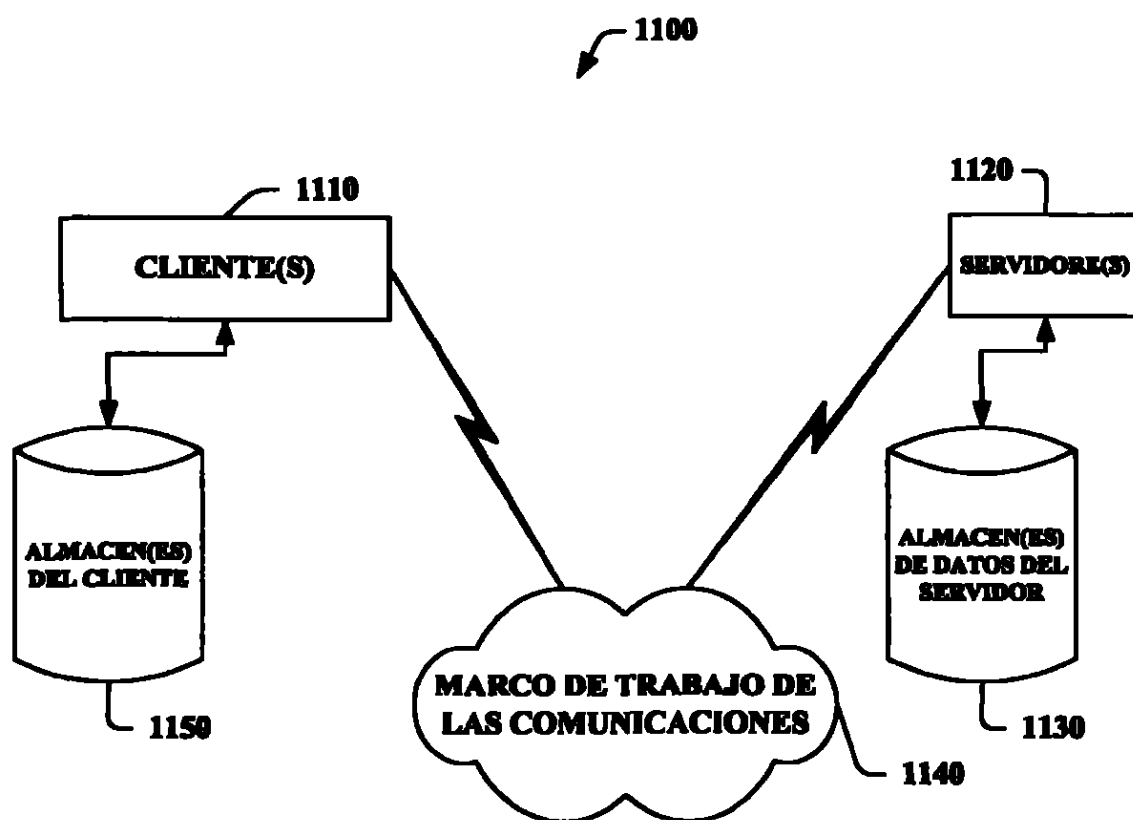


FIG. 11

3-6

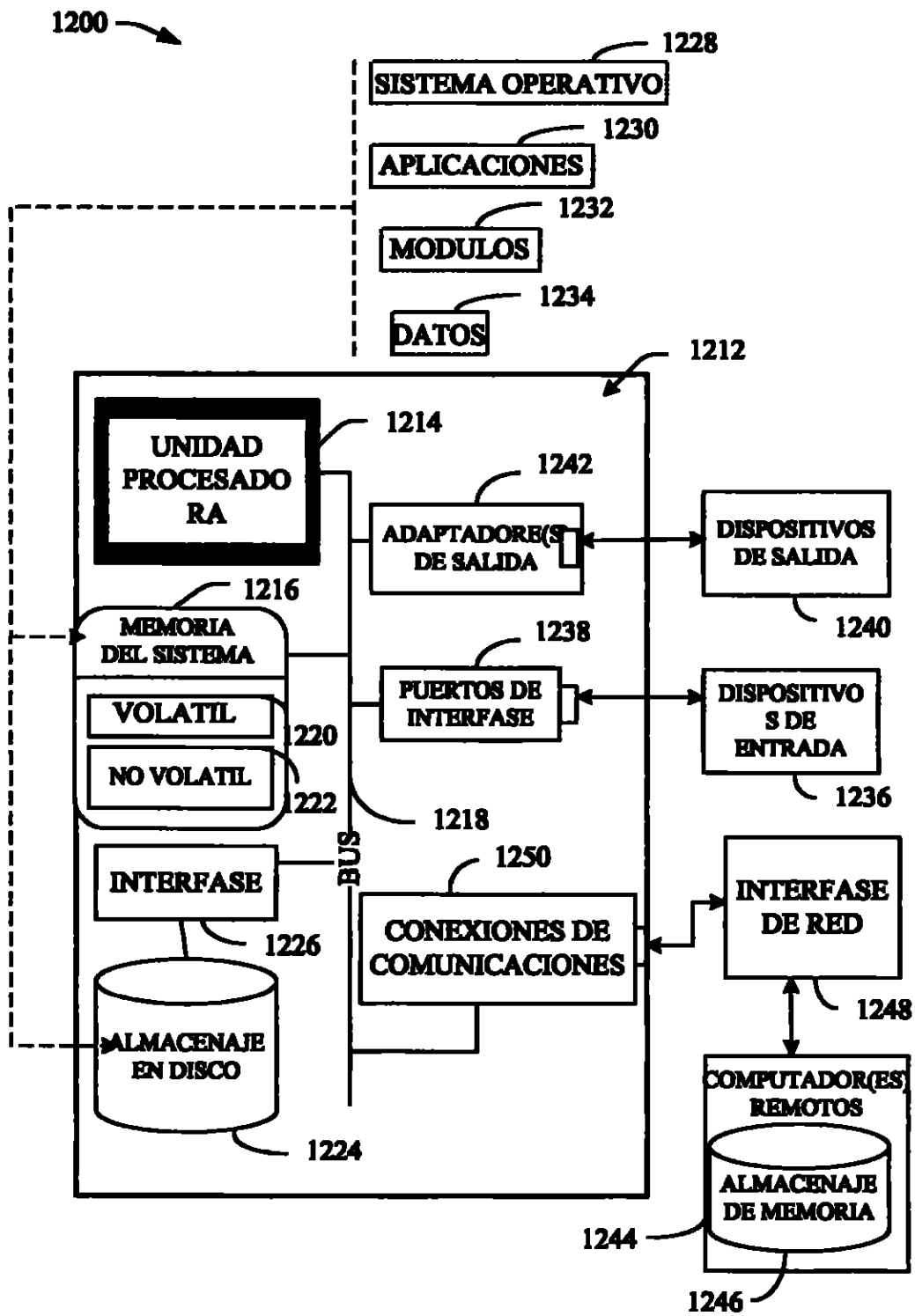


FIG. 12

87

Sheet of Particulars

Application Information

MS Main File Number: 311235

File #	Country
311235.02	EP
311235.03	JP
311235.04	CN
311235.05	KR
311235.06	IN
311235.07	CA
311235.08	AU
311235.09	BR
311235.10	RU
311235.11	MX
311235.12	ZA
311235.13	HK
311235.14	TH
311235.15	MY
311235.16	TW
311235.17	NO
311235.18	EG
311235.19	PH
311235.20	ID
311235.21	IL
311235.22	SG
311235.23	CL
311235.24	CO
311235.25	NZ

Title of Invention:

MAPPING OF A FILE SYSTEM MODEL TO A DATABASE OBJECT

Priority information:

Country: United States

Serial Number: 11/091,079

Priority Filing Date: March 28, 2005

Inventors:

Please use Microsoft Corporation's address as the address for the inventor(s) below.

Inventors	
Inventor Name	Citizenship
Amit Shukla	India
Anil Kumar Nori	United States
Bekim Demiroski	Macedonia
Gregory S. Friedman	United States

Jason T. Hunter	United States
Jeffrey T. Pearce	United States
Michael J Newman	United States
Nigel R. Ellis	United States
Srinivasmurthy P. Acharya	United States

Applicant: Microsoft is the only applicant unless additions are noted below as co-assignees.

Microsoft Corporation
 One Microsoft Way
 Redmond, WA 98052
 United States of America

State of Incorporation: Washington

Application document: Available on Web Site (unless otherwise stated)

Page counts:

Specification: 31
 Claims: 3 (No. of claims - 20)
 Drawings: 12 (No. of figures - 12)
 Abstract: 1

Total Pages: 47

Figure for publication: 1

X Priority application and enclosed application are identical **EXCEPT** the re-numbering of claims 14-20 to 13-19 and the addition of claim 20. Please see the Preliminary Amendment for complete details.

Instructions

Please provide a *soft* copy of the application documents filed in this case, as well as a copy of all correspondence filed with and issued by the patent office throughout prosecution of this application. If you require any additional documentation at this time, please let us know immediately. In the absence of instructions, please take all actions necessary to maintain this application. This application should not be abandoned nor allowed to lapse without our prior written consent.

When forwarding an Office Action issued by the Patent Office, we ask that you send us a suitable draft response along with it for our review as we will be relying on your expertise and knowledge of local patent rules and laws.

Thank you for your assistance with this matter. Please promptly acknowledge receipt of our instructions by electronic mail to sender with a copy to msintl@microsoft.com.

29

Additional Information

The following information is provided as additional information for you to use when assigning your internal resources and to help you understand the strategy/relationships that this invention has within Microsoft. We do not have specific instructions as to the use of this information.

US Law Firm: This firm/attorney handled the original drafting of the application.

OC Resource		
Law firm	Attorney	OC Reference
Amin	Carlos Garritano	MSFTP852US



UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
ASSISTANT SECRETARY AND COMMISSIONER
OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

AUGUST 25, 2005

PTAS

500046958A

500046958A

HIMANSHU S. AMIN
24TH FLOOR, 1900 E. 9TH STREET
CLEVELAND, OH 44114

UNITED STATES PATENT AND TRADEMARK OFFICE
NOTICE OF RECORDATION OF ASSIGNMENT DOCUMENT

THE ENCLOSED DOCUMENT HAS BEEN RECORDED BY THE ASSIGNMENT DIVISION OF THE U.S. PATENT AND TRADEMARK OFFICE. A COMPLETE MICROFILM COPY IS AVAILABLE AT THE ASSIGNMENT SEARCH ROOM ON THE REEL AND FRAME NUMBER REFERENCED BELOW.

PLEASE REVIEW ALL INFORMATION CONTAINED ON THIS NOTICE. THE INFORMATION CONTAINED ON THIS RECORDATION NOTICE REFLECTS THE DATA PRESENT IN THE PATENT AND TRADEMARK ASSIGNMENT SYSTEM. IF YOU SHOULD FIND ANY ERRORS OR HAVE QUESTIONS CONCERNING THIS NOTICE, YOU MAY CONTACT THE EMPLOYEE WHOSE NAME APPEARS ON THIS NOTICE AT 703-308-9723. PLEASE SEND REQUEST FOR CORRECTION TO: U.S. PATENT AND TRADEMARK OFFICE, MAIL STOP: ASSIGNMENT SERVICES DIVISION, P.O. BOX 1450, ALEXANDRIA, VA 22313.

RECORDATION DATE: 08/25/2005

REEL/FRAME: 016451/0719
NUMBER OF PAGES: 4

BRIEF: ASSIGNMENT OF ASSIGNOR'S INTEREST (SEE DOCUMENT FOR DETAILS).

ASSIGNOR:

ELLIS, NIGEL R.

DOC DATE: 03/24/2005

ASSIGNOR:

FRIEDMAN, GREGORY S.

DOC DATE: 03/24/2005

ASSIGNOR:

DEMIROSKI, BEKIM

DOC DATE: 03/24/2005

ASSIGNOR:

NORI, ANIL KUMAR

DOC DATE: 03/24/2005

ASSIGNOR:

SHUKLA, AMIT

DOC DATE: 03/24/2005

ASSIGNOR:

ACHARYA, SRINIVASMURTHY P.

DOC DATE: 03/24/2005

at

92

016451/0719 PAGE 2

ASSIGNOR:
PEARCE, JEFFREY T.

DOC DATE: 03/24/2005

ASSIGNOR:
NEWMAN, MICHAEL J.

DOC DATE: 03/24/2005

ASSIGNOR:
HUNTER, JASON T.

DOC DATE: 03/24/2005

ASSIGNEE:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WASHINGTON 98052

SERIAL NUMBER: 11091079

FILING DATE: 03/28/2005

PATENT NUMBER:

ISSUE DATE:

TITLE: MAPPING OF A FILE SYSTEM MODEL TO A DATABASE OBJECT

DOROTHY RILEY, PARALEGAL
ASSIGNMENT DIVISION
OFFICE OF PUBLIC RECORDS

~~92~~

PATENT ASSIGNMENT

Electronic Version v1.1
Stylesheet Version v1.1

08/25/2005
500046958

SUBMISSION TYPE:	NEW ASSIGNMENT																				
NATURE OF CONVEYANCE:	ASSIGNMENT																				
CONVEYING PARTY DATA																					
<table border="1"><thead><tr><th>Name</th><th>Execution Date</th></tr></thead><tbody><tr><td>Nigel R. Ellis</td><td>03/24/2005</td></tr><tr><td>Gregory S. Friedman</td><td>03/24/2005</td></tr><tr><td>Baldm Demiroeld</td><td>03/24/2005</td></tr><tr><td>Anil Kumar Nool</td><td>03/24/2005</td></tr><tr><td>Amit Shulda</td><td>03/24/2005</td></tr><tr><td>Srinivasamurthy P. Acharya</td><td>03/24/2005</td></tr><tr><td>Jeffrey T. Pearce</td><td>03/24/2005</td></tr><tr><td>Michael J. Newman</td><td>03/24/2005</td></tr><tr><td>Jason T. Hunter</td><td>03/24/2005</td></tr></tbody></table>		Name	Execution Date	Nigel R. Ellis	03/24/2005	Gregory S. Friedman	03/24/2005	Baldm Demiroeld	03/24/2005	Anil Kumar Nool	03/24/2005	Amit Shulda	03/24/2005	Srinivasamurthy P. Acharya	03/24/2005	Jeffrey T. Pearce	03/24/2005	Michael J. Newman	03/24/2005	Jason T. Hunter	03/24/2005
Name	Execution Date																				
Nigel R. Ellis	03/24/2005																				
Gregory S. Friedman	03/24/2005																				
Baldm Demiroeld	03/24/2005																				
Anil Kumar Nool	03/24/2005																				
Amit Shulda	03/24/2005																				
Srinivasamurthy P. Acharya	03/24/2005																				
Jeffrey T. Pearce	03/24/2005																				
Michael J. Newman	03/24/2005																				
Jason T. Hunter	03/24/2005																				
RECEIVING PARTY DATA																					
<table border="1"><tr><td>Name:</td><td>Microsoft Corporation</td></tr><tr><td>Street Address:</td><td>One Microsoft Way</td></tr><tr><td>City:</td><td>Redmond</td></tr><tr><td>State/Country:</td><td>WASHINGTON</td></tr><tr><td>Postal Code:</td><td>98052</td></tr></table>		Name:	Microsoft Corporation	Street Address:	One Microsoft Way	City:	Redmond	State/Country:	WASHINGTON	Postal Code:	98052										
Name:	Microsoft Corporation																				
Street Address:	One Microsoft Way																				
City:	Redmond																				
State/Country:	WASHINGTON																				
Postal Code:	98052																				
PROPERTY NUMBERS Total: 1																					
<table border="1"><thead><tr><th>Property Type</th><th>Number</th></tr></thead><tbody><tr><td>Application Number:</td><td>11091079</td></tr></tbody></table>		Property Type	Number	Application Number:	11091079																
Property Type	Number																				
Application Number:	11091079																				
CORRESPONDENCE DATA																					
Fax Number: (216)898-8731 <i>Correspondence will be sent via US Mail when the fax attempt is unsuccessful.</i>																					
Phone: 216-898-8730																					
Email: volozeweld@thepatentattorneys.com																					
Correspondent Name: Himanshu S. Amin																					
Address Line 1: 24th Floor, 1800 E. 8th Street																					
Address Line 4: Cleveland, OHIO 44114																					

OP: \$10.00 11091079

93

NAME OF SUBMITTER:

Himanshu S. Arora

Total Attachments: 2

source=M8311235.01.AssignmentPage1.tif

source=M8311235.01.AssignmentPage2.tif

PATENT ASSIGNMENT

We, Nigel R. Ellis, Gregory S. Friedman, Bekim Demiroski, Anil Kumar Nori, Amit Shukla, Srinivasmurthy P. Acharya, Jeffrey T. Pearce, Michael J. Newman, and Jason T. Hunter ("ASSIGNORS"), have invented subject matter ("INVENTION") disclosed and/or claimed in a patent application entitled " MAPPING OF A FILE SYSTEM MODEL TO A DATABASE OBJECT " ("APPLICATION"), which:

- ☒ will be filed without this executed PATENT ASSIGNMENT. ASSIGNORS hereby authorize and request, ASSIGNEE'S legal representatives, of Amin and Turocy, LLC, 1900 E. 9th Street, Cleveland, Ohio 44114, who are associated with customer number 27195, to insert here in parenthesis (U.S. Serial No. 11/091,079, filed 3/28/05) this APPLICATION's U.S. Serial Number and filing date, when known;
- ☐ was filed on and was given U.S. Serial No.;
- ☐ is filed concurrently herewith;

Microsoft Corporation, a Washington Corporation, on behalf of itself and its successors and assigns ("ASSIGNEE"), is entitled to, and is desirous of acquiring, the entire and exclusive rights, title and interest in the INVENTION and the APPLICATION (and all other applications and patents derived therefrom, such as continuing applications, in and for the United States, its territories, and all foreign countries ("APPLICATION DERIVATIVES"));

For good and valuable consideration, the receipt of which is hereby acknowledged by the ASSIGNORS, the ASSIGNORS hereby sell, assign and transfer to the ASSIGNEE, the entire and exclusive rights, title and interest in the INVENTION and the APPLICATION (and APPLICATION DERIVATIVES);

ASSIGNORS agree to execute all instruments and documents required for the making and prosecution of the APPLICATION (and APPLICATION DERIVATIVES), for litigation regarding letters patent derived therefrom, and for the purpose of protecting and perfecting title to the APPLICATION (and APPLICATION DERIVATIVES).

3/24/2005
DATE

3/24/2005
DATE

3/24/2005
DATE

Nigel R. Ellis
Nigel R. Ellis

Gregory S. Friedman
Gregory S. Friedman

Bekim Demiroski
Bekim Demiroski

[Handwritten mark]

3/24/2005
DATE

3/24/2005
DATE

3/24/2005
DATE

3/24/2005
DATE

3/24/05
DATE

3/24/05
DATE

Anil Kumar Nori
Anil Kumar Nori

Amit Shukla
Amit Shukla

Srinivasmurthy P. Acharya
Srinivasmurthy P. Acharya

Jeffrey T. Pearce
Jeffrey T. Pearce

Michael J. Newman
Michael J. Newman

Jason T. Hunter
Jason T. Hunter

PATENT

MS311235.01/MSFTP852US

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being fixed to 703-672-9306 on the date shown below to Mail Stop Non-Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Date: 5-25-05


Himanshu S. Amin

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Applicant(s): Nigel Ellis, *et al.*

Serial No: 11/091,079

Filing Date: March 28, 2005

Examiner: Unknown

Art Unit: Unknown

Title: MAPPING OF A FILE SYSTEM MODEL TO A DATABASE OBJECT

Mail Stop Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

PRELIMINARY AMENDMENT

Dear Sir:

Entry of this preliminary amendment is respectfully requested prior to performing substantive examination of the above-identified patent application.



AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS

1. (Original) A system that facilitates mapping a data model, comprising:
 - a file storage data model that is utilized to define a mapping schema; and
 - a type storage component that maps a schema object to a database object based on the associated mapping schema, wherein an instance of a type in the mapping schema is stored.
2. (Original) The system of claim 1, the instance of the type is at least one of: a document; an image; music; a video; a contact; a message; a person; an organization; an email; a fax; a voice; and an audio clip.
3. (Original) The system of claim 1, the type storage component further provides a query to find at least one of the following: at least one item in the system that satisfies a respective criteria; at least one document in the system that satisfies a respective criteria; and at least one contact that satisfies a respective criteria.
4. (Original) The system of claim 3, the contact can include at least one of the following: an individual; an organization; and a group.
5. (Original) The system of claim 1, further comprising a relational component that utilizes at least one of a relational storage and a relational query capability.
6. (Original) The system of claim 5, the relational component utilizes a relational database technique that is a collection of data items organized as a set of formally described tables, wherein data can be at least one of accessed without reorganization and reassembled without reorganization.

7. (Original) The system of claim 1, the mapping describes at least one of the database objects created based on the schema and how the instance of the type described in the schema is stored and accessed.
8. (Original) The system of claim 1, the type in the schema maps to a common language runtime (CLR) class in a storage of the file storage data model.
9. (Original) The system of claim 8, the type is at least one of the following: an item; an item extension; an item fragment; and a link.
10. (Original) The system of claim 1, further comprising a table that has at least one of the following: 1) a column that contains at least one instance of the respective type; and 2) a row that contains a serialized representation of the CLR class instance that represents the type instance.
11. (Original) The system of claim 1, further comprising a view component that generates a view that exposes at least one instance of the type, wherein the view projects a subset of the respective type of the view associated with a base type.
12. (Original) The system of claim 1, the type contains at least one of the following a type hierarchy and an inheritance.
- ~~14.13.~~ (Currently Amended) A computer readable medium having stored thereon the components of the system of claim 1.

15.14. (Currently Amended) A computer-implemented method that facilitates mapping a data model, comprising:

- obtaining a file storage data model;
- receiving at least one of a schema, a type, a criteria, and a query information;
- mapping a schema object to a database object based on the schema associated to the file storage data model; and
- storing an instance of the type from the schema.

16.15. (Currently Amended) The method of claim 14-15, further comprising querying to find at least one of the following: at least one item in the system that satisfies a respective criteria; at least one document in the system that satisfies a respective criteria; and at least one contact that satisfies a respective criteria.

17.16. (Currently Amended) The method of claim 14-15, further comprising utilizing a relational database engine to provide a relational storage and a relational query capability.

18.17. (Currently Amended) The method of claim 14-15, further comprising generating a view that exposes at least one instance of the type, wherein the view projects a subset of the respective type of the view associated with a base type.

19.18. (Currently Amended) A data packet that communicates between a file storage data model and a type storage component, the data packet facilitates the method of claim 14-15.

20.19. (Currently Amended) A computer-implemented system that facilitates mapping a data model, comprising:

- means for receiving at least one of the following a schema, a type, a criteria, and a query criteria; and
- means for mapping a schema object to a database object based on a schema associated to a file storage data model, wherein an instance of a type in the schema is stored.

20. (New) The system of claim 11, the view component invokes a user interface to interact between a user and the type storage component.

REMARKS

Please add new claim 20 to the subject application in accordance herewith. Please note that originally filed claims 14-20 have been renumbered to 13-19 and the addition of claim 20 does not require any additional fees. A complete listing of the claims can be found at pages 2-5. Applicants' representative submits that no new matter has been entered into the present application by this preliminary amendment.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number listed below.

In the event any additional fees are due in connection with the filing of this document, including any additional extensions of time that may be required, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,

AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731

Title: MAPPING OF A FILE SYSTEM MODEL TO A DATABASE OBJECT**TECHNICAL FIELD**

[0001] The present invention generally relates to databases, and more particularly to systems and/or methods that facilitate storing a type instance and/or querying data.

BACKGROUND OF THE INVENTION

[0002] Advances in computer technology (e.g., microprocessor speed, memory capacity, data transfer bandwidth, software functionality, and the like) have generally contributed to increased computer application in various industries. Ever more powerful server systems, which are often configured as an array of servers, are commonly provided to service requests originating from external sources such as the World Wide Web, for example.

[0003] As the amount of available electronic data grows, it becomes more important to store such data in a manageable manner that facilitates user friendly and quick data searches and retrieval. Today, a common approach is to store electronic data in one or more databases. In general, a typical database can be referred to as an organized collection of information with data structured such that a computer program can quickly search and select desired pieces of data, for example. Commonly, data within a database is organized *via* one or more tables. Such tables can be arranged with rows and columns.

[0004] The tables can comprise one or more records, wherein a record can include a set of fields. Records are commonly indexed as rows within a table, and the record fields are typically indexed as columns, such that a row/column pair of indices can reference a particular datum within a table. For example, a row may store a complete data record relating to a sales transaction, a person, or a project. Likewise, columns of the table can define discrete portions of the rows that have the same general data format, wherein the columns can define fields of the records.

[0005] Each individual piece of data, standing alone, is generally not very informative. Database applications make data more useful because they help users organize and process the data. The database application allows the user to compare, sort,

order, merge, separate and interconnect the data, so that useful information can be generated from the data. Yet, the capacity and versatility of databases have grown to an incredible amount to allow a virtually endless storage capacity utilizing databases. Moreover, typical database systems offer limited query-ability based upon time, file extension, location, and size. For example, in order to search the vast amounts of data associated to a database, a typical search is limited to a file name, a file size, a date of creation, wherein such techniques are deficient and inept.

[0006] With a continuing and increasing creation of data from end-users, the problems and difficulties surrounding finding, relating, and storing such data is reaching its peak. End-users write documents, store photos, rip music from compact discs, receive email, retain copies of sent email, *etc.* For example, in the simple process of creating a music compact disc, the end-user can create megabytes of data. Ripping the music from the compact disc, converting the file to a suitable format, creating a jewel case cover, designing a compact disc label, all require the creation of data.

[0007] Not only are the complications surrounding users, developers have similar issues with data. Developers create and write myriad of applications varying from personal applications to highly developed enterprise applications. While creating and/or developing, developers frequently, if not always, gather data. While obtaining such data, the data needs to be stored. In other words, the problems and difficulties surrounding finding, relating, and storing data affect both the developer and the end user. In view of the above, there is a need to improve upon and/or provide systems and/or methods that mitigate deficiencies associated with conventional systems and databases.

SUMMARY OF THE INVENTION

[0008] The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is intended to neither identify key or critical elements of the invention nor delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed description that is presented later.

[0009] The subject invention relates to systems and/or methods that facilitate mapping of a data model to a database object. A type storage system can provide storing of a type instance and/or querying. The storing of the type can describe the database object and how instances of types are stored and/or accessed. The querying can find at least one of an item, a document, and a contact that each satisfies a certain criteria. Moreover, the type storage system can utilize an interface to receive data, wherein the data can include a schema, a type, a criteria, a query criteria, *etc.*

[0010] In accordance with one aspect of the subject invention, the type storage system can include a store component that stores at least one type instance. The storage can be a mapping of the data model, wherein the data model can represent a file storage system. Furthermore, the type storage system can also include a query component. The query component can provide at least one query in the file storage system data model to satisfy a criterion. In accordance with another aspect of the subject invention, the type storage system can include a relational component that can utilize a relational storage and/or a relational query capability. The relational component can invoke a database engine to provide such relational techniques, wherein these techniques can facilitate storing the type instance and/or querying.

[0011] Moreover, the type storage system can further include a view component that can provide a view that exposes all instances of a given type. The type can be a hierarchical structure and/or an inheritance structure. The view can be related to a particular type that projects a subset of the types with a base type. In other aspects of the subject invention, methods are provided that facilitate mapping of a data model to a database object.

[0012] The following description and the annexed drawings set forth in detail certain illustrative aspects of the invention. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention may be employed and the subject invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention will become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0013] Fig. 1 illustrates a block diagram of an exemplary system that facilitates storing a type instance associated with a data model.
- [0014] Fig. 2 illustrates a block diagram of an exemplary system that facilitates storing a type instance and/or querying to find at least one of an item, a document, and a contact.
- [0015] Fig. 3 illustrates a block diagram of an exemplary system that facilitates utilizing a relational storage and/or relational query capabilities.
- [0016] Fig. 4 illustrates a block diagram of an exemplary system that facilitates mapping and/or viewing in conjunction with a type storage system.
- [0017] Fig. 5 illustrates a block diagram of an exemplary system that facilitates storing a type instance associated with a data model.
- [0018] Fig. 6 illustrates a block diagram of a high-level structure of a store within a file storage data model.
- [0019] Fig. 7 illustrates a block diagram of a type instance with associated tables.
- [0020] Fig. 8 illustrates block diagram of a type hierarchy and a corresponding view projection.
- [0021] Fig. 9 illustrates an exemplary methodology for storing a type instance that maps to a data model and/or providing a query.
- [0022] Fig. 10 illustrates an exemplary methodology for storing a type instance that maps to a data model and/or providing a query.
- [0023] Fig. 11 illustrates an exemplary networking environment, wherein the novel aspects of the subject invention can be employed.
- [0024] Fig. 12 illustrates an exemplary operating environment that can be employed in accordance with the subject invention.

DESCRIPTION OF THE INVENTION

- [0025] As utilized in this application, terms “component,” “system,” “interface,” “schema,” and the like are intended to refer to a computer-related entity, either hardware, software (e.g., in execution), and/or firmware. For example, a component can be a process running on a processor, a processor, an object, an executable, a program, and/or a

computer. By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and a component can be localized on one computer and/or distributed between two or more computers.

[0026] The subject invention is described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the subject invention. It may be evident, however, that the subject invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to facilitate describing the subject invention.

[0027] Now turning to the figures, Fig. 1 illustrates a system 100 that facilitates storing a type instance associated with a data model. The data model can be a file storage data model 102 that enables storing, finding, and relating information. For example, an information type can be, but is not limited to, a document, an image, a video, a contact, a message, an email, an audio clip, *etc.* The informational types can be considered units of information, which can be represented as instances of complex types that are part of a type system that supports inheritance. Inheritance can be defined as a situation in which certain characteristics are passed on from one context to another. In particular to object-oriented programming, objects inherit attributes and/or behaviors from other objects. It is to be appreciated that inheritance can be considered a hierarchical structure and/or format.

[0028] A type storage system 104 can store the type instance and query to efficiently and effectively find at least one of items, documents, and/or contacts. The type storage system 104 can receive data, wherein the data can include a type, a criteria, a schema, a query criteria, Specifically, storing the type instance that relates to an informational type (*e.g.*, a document, an image, a video, a contact, a message, an email, an audio clip) can provide at least one of the following: 1) find at least one item in the file storage data model 102 that satisfy a certain criteria; 2) find at least one document in the file storage data model 102 that satisfy a particular criteria; and 3) find at least one

contact (*e.g.*, including an individual, organization, and group) that satisfy a certain criteria.

[0029] The type storage system 104 can utilize a relational database technique associated to a relational storage and a relational query. It is to be appreciated that such capabilities can be provided by a database engine (not shown). The relational database techniques can be associated to relational databases, wherein a relational database is a collection of data items organized as a set of formally-described tables. The data within the tables can be accessed and/or reassembled in various ways without the requirement of reorganizing the database tables. Furthermore, the relational database can be easily extended, such as adding new categories without modifications to existing applications and/or data. It is to be appreciated that the subject invention is not limited to relational databases and/or associated techniques and that any suitable technique can be utilized.

[0030] The system 100 further includes an interface component 106, which provides various adapters, connectors, channels, communication paths, *etc.* to integrate the type storage system 104 into virtually any operating system. In addition, the interface component 106 can provide various adapters, connectors, channels, communication paths, *etc.* that provide for interaction with data and the type storage system 104. It is to be appreciated that although the interface component 106 is incorporated into the type storage system 104, such implementation is not so limited. For instance, the interface component 106 can be a stand-alone component to receive or transmit the data in relation to the system 100.

[0031] Fig. 2 illustrates a system 200 that facilitates storing a type instance and/or querying to find an item, a document, and a contact associated to a data model 202. The data model 202 can be a model representation of a file storage system that utilizes a hierarchical characteristic and/or inheritance. The type can include a document, an image, a video, a contact, a message, an email, an audio clip, *etc.* However, it is to be appreciated that a type can be a typical information type stored in a system represented by the data model 202. A type storage system 204 can store the type instance and provide querying that can efficiently find at least one of the following: an item in the data model 202, a document in the data model 202, and a contact in the data model 202. The type storage system 204 can receive data, wherein the data can be a query criteria, a

schema, a criteria, a schema definition, a data model, a type, The system 200 can further employ an interface 206 to facilitate providing various adapters, connectors, channels, communication paths, *etc.* to integrate the type storage system 204 into virtually any operating system and facilitate communication.

[0032] The type storage system 204 can further include a store component 208 ("store 208") that can store the type instances. The storage can be a mapping of the data model 202, wherein the storage mapping can describe the database object that is created based on a schema definition and how instances of types described in the schema are stored and/or accessed. In other words, the store 208 can store the instances of the type and at least one rule associated to mapping a type declaration into the database object.

[0033] The type storage system 204 can include a query component 210 ("query 210") that provides a querying of data. The query 210 can obtain at least one of the following: an item in the system represented by data model 202; a document in the system represented by data model 202; and a contact (*e.g.*, including an individual, an organization, and a group) in the system represented by the data model 202. It is to be appreciated and understood that the query can be based at least upon a certain criteria and/or query criteria obtained *via* the interface 206. Furthermore, the query 210 is not so limited to the item, document, and contact, and any suitable information type stored in the system represented by the data model 202 can be utilized.

[0034] Fig. 3 illustrates a system 300 that facilitates utilizing a relational storage and relational query capabilities. A type storage system 304 can store a type instance and query to efficiently and effectively find items, documents, and contacts, wherein such elements are associated to a file system represented by a data model 302. It is to be appreciated that the file system can be a storage file system that utilizes a hierarchical structure and/or an inheritance characteristic. The type storage system 304 can be substantially similar to the type storage system 104, 204, as depicted in Figs 1 and 2 respectively. The type storage system 304 can invoke an interface 306 to facilitate communication and/or receiving data to be further utilized in accordance with the subject invention.

[0035] The type storage system 304 can include a store component 308 (store 308) and a query component 310 ("query 310"). The store 308 can provide any suitable

storing technique to store the type instance. The query 310 can provide a query that can efficiently and effectively obtain at least one of the following: at least one item in a system that meets a criteria; at least one document in a system that satisfies a criteria; and at least one contact that meets a criteria. It is to be appreciated and understood that the store 308 and the query 310 can be substantially similar to the store 208, and the query 210 as depicted in Fig. 2.

[0036] The type storage system 304 can further include a relational component 312 ("relational 312"). The relational 312 utilizes database techniques (e.g., utilizing a database engine) to build a relational storage and/or provide a relational query capability to facilitate storing type instances and/or querying. The relational 312 can incorporate techniques associated to a relational database, wherein the relational database is a collection of data items organized as a set of formally-described tables as described in detail above. It is to be appreciated that the subject invention is not limited to relational database and associated techniques and that any suitable technique can be utilized.

[0037] Fig. 4 illustrates a system 400 that facilitates mapping and/or viewing in conjunction with a data model 402. The data model 402 can represent a file storage system that enables storing, finding, and relating information. A typical information type that can be stored in the system can include a document, an image, music, a video, a contact, a message, *etc.* The information type can be represented as instances of complex types that are part of a type system that supports inheritance. A type storage system 404 can store the type instances and provide a query to find items, documents, and contacts that satisfy a certain criteria. The system 400 can further employ an interface 406 to facilitate communication and/or receiving data that can include criteria, types, schemas, models, and query criteria.

[0038] The type storage system 404 can further include a map component 408 to facilitate storing the type instance. The map component 408 provides mapping of types described in a schema to defined types and database objects. The map component 408 can be a storage mapping that describes at least one database object, wherein the database object can be created based on a schema definition and/or how instances of types described in the schema are stored and/or accessed. In other words, the instances of types can be stored and rules can be utilized to map a type declaration into the database object.

Each type in the schema maps to a class (e.g., common language runtime (CLR)) in a storage.

[0039] The type storage system 404 can include a view component 410 to provide a view projection. The view projection can expose instances of the type that is viewed. It is to be appreciated that the types can be in a hierarchical structure utilizing at least inheritance. In other words, each type is a part of a type hierarchy. The view can be associated with a given type and can project a subset of the respective types of the view associated with its base type. The view can project instances associated to the particular type. For example, for a type "message," only the instances of which message is a parent can be viewed based at least upon the hierarchy structure.

[0040] The view component 410 can provide various types of user interfaces to facilitate interaction between a user and any component coupled to type storage system 404. As depicted, the view component 410 is a separate entity that is incorporated into the type storage system 404. However, it is to be appreciated that the view component 410 and/or similar view components can be a separate component from the type storage system 404 and/or a stand-alone unit. The view component 410 can provide one or more graphical user interfaces (GUIs), command line interfaces, and the like. For example, a GUI can be rendered that provides a user with a region or means to load, import, read, *etc.* data, and can include a region to present the results of such. These regions can comprise known text and/or graphic regions comprising dialogue boxes, static controls, drop-down-menus, list boxes, pop-up menus, as edit controls, combo boxes, radio buttons, check boxes, push buttons, and graphic boxes. In addition, utilities to facilitate the presentation such vertical and/or horizontal scroll bars for navigation and toolbar buttons to determine whether a region will be viewable can be employed. For example, the user can interact with one or more of the components coupled to the type storage system 404.

[0041] The user can also interact with the regions to select and provide information *via* various devices such as a mouse, a roller ball, a keypad, a keyboard, a pen and/or voice activation, for example. Typically, a mechanism such as a push button or the enter key on the keyboard can be employed subsequent entering the information in order to initiate the search. However, it is to be appreciated that the invention is not so

limited. For example, merely highlighting a check box can initiate information conveyance. In another example, a command line interface can be employed. For example, the command line interface can prompt (e.g., *via* a text message on a display and an audio tone) the user for information *via* providing a text message. The user can then provide suitable information, such as alpha-numeric input corresponding to an option provided in the interface prompt or an answer to a question posed in the prompt. It is to be appreciated that the command line interface can be employed in connection with a GUI and/or API. In addition, the command line interface can be employed in connection with hardware (e.g., video cards) and/or displays (e.g., black and white, and EGA) with limited graphic support, and/or low bandwidth communication channels.

[0042] Fig. 5 illustrates a system 500 that employs intelligence to facilitate storing a type instance associated with a data model 502. The system 500 includes a type storage system 504, an interface 506, and the data model 502, which can be substantially similar to the components depicted in earlier figures. The interface 506 can facilitate communication associated to data, which can include a criteria, a type, a schema, a data model, and a query criteria. The system 500 can provide storing a type, querying, and/or providing a view. It is to be appreciated and understood that a database engine can provide a relational storage and a relational query to the system 500.

[0043] The system 500 further includes an intelligent component 508. The intelligent component 508 can be utilized by the type storage system 504 to facilitate storing and/or querying for the system 500. For example, the intelligent component 510 can be utilized to facilitate determining a user defined type to be stored. Historic data in conjunction with user profiles can allow the intelligent component 508 to determine storing the type and/or querying with certain criteria.

[0044] It is to be understood that the intelligent component 508 can provide for reasoning about or infer states of the system, environment, and/or user from a set of observations as captured *via* events and/or data. Inference can be employed to identify a specific context or action, or can generate a probability distribution over states, for example. The inference can be probabilistic - that is, the computation of a probability distribution over states of interest based on a consideration of data and events. Inference can also refer to techniques employed for composing higher-level events from a set of



events and/or data. Such inference results in the construction of new events or actions from a set of observed events and/or stored event data, whether or not the events are correlated in close temporal proximity, and whether the events and data come from one or several event and data sources. Various classification (explicitly and/or implicitly trained) schemes and/or systems (*e.g.*, support vector machines, neural networks, expert systems, Bayesian belief networks, fuzzy logic, data fusion engines...) can be employed in connection with performing automatic and/or inferred action in connection with the subject invention.

[0045] A classifier is a function that maps an input attribute vector, $x = (x_1, x_2, x_3, x_4, x_n)$, to a confidence that the input belongs to a class, that is, $f(x) = \text{confidence}(\text{class})$. Such classification can employ a probabilistic and/or statistical-based analysis (*e.g.*, factoring into the analysis utilities and costs) to prognose or infer an action that a user desires to be automatically performed. A support vector machine (SVM) is an example of a classifier that can be employed. The SVM operates by finding a hypersurface in the space of possible inputs, which hypersurface attempts to split the triggering criteria from the non-triggering events. Intuitively, this makes the classification correct for testing data that is near, but not identical to training data. Other directed and undirected model classification approaches include, *e.g.*, naïve Bayes, Bayesian networks, decision trees, neural networks, fuzzy logic models, and probabilistic classification models providing different patterns of independence can be employed. Classification as used herein also is inclusive of statistical regression that is utilized to develop models of priority.

[0046] Furthermore, the intelligent component 508 can utilize a data store 510 to store user profiles and/or historic data. The data store 510 can be, for example, either volatile memory or nonvolatile memory, or can include both volatile and nonvolatile memory. By way of illustration, and not limitation, nonvolatile memory can include read only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), or flash memory. Volatile memory can include random access memory (RAM), which acts as external cache memory. By way of illustration and not limitation, RAM is available in many forms such as static RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM

(SDRAM), double data rate SDRAM (DDR SDRAM), enhanced SDRAM (ESDRAM), Synchlink DRAM (SLDRAM), Rambus direct RAM (RDRAM), direct Rambus dynamic RAM (DRDRAM), and Rambus dynamic RAM (RDRAM). The data store 510 of the subject systems and methods is intended to comprise, without being limited to, these and any other suitable types of memory. In addition, it is to be appreciated that the data store 510 can be a server and/or database.

[0047] Fig. 6 illustrates a high-level structure of a store. A schema can be provided, wherein various instances of types can be stored in tables dependent upon their type. The types can be, but are not limited to, an item, an item extension, an item fragment, and a link. Each type can have a corresponding table that can include columns having object instances therein. For instance, one column in the item table can contain all instances of items in the store. For each row, the column can contain a serialized representation of the CLR class instance that represents the item type instance. It is to be appreciated that for the item extension, item fragment and the link type instances can be represented in a similar structure. Inline type instances can be stored inside a parent object instance rather than storing in separate tables and/or columns. For each item, item extension, item fragment, and link a view can be generated that exposes all the instances of that type. Each type is a part of a type hierarchy. For instance, the item view projects all items in the store.

[0048] Briefly turning to Fig. 7, an item 702 is depicted. A table 704 of the item 702 can contain all instances of the items in the store (not shown). The table 704 can contain an object instance 706 ("Document") and an object instance 708 ("Contact"). It is to be appreciated that each type in a schema maps to a CLR class in the storage. The object instance 706 can include a title, an abstract, a printed, an author, and various other metadata associated with the object instance 706. Similarly, the object instance 708 can include various metadata such as name, address, email, *etc.*

[0049] Fig. 8 illustrates a type hierarchy 800 and a corresponding view projection 820. The type hierarchy 800 can include an item 802, a contact 804, a document 806, a message 808, a person 810, an organization 812, an email 814, a fax 816, and a voice 818. It is to be appreciated that the type hierarchy 800 is an example and any suitable hierarchy and/or types can be utilized in accordance with the subject invention. As

illustrated, item 802 is considered a parent of every type included therein. Thus, contact 804 is the parent for person 810 and organization 812, while message 808 is the parent for email 814, fax 816, and voice 818. Furthermore, the corresponding view projection 820 can reflect the type hierarchy 800. The view associated with a given type can project a subset of the items of the view associated with its base type. For example, the item view projects all the items in the store. The contact view projects only the items that are of type Contact. The person view projects only the contacts that have the most derived type as person.

[0050] Turning back to Fig. 6, type mapping can be provided utilizing an algorithm for mapping the type to the type used to describe the corresponding storage structure in the store. It is to be appreciated that the mapping can be associated to any file storage system (*e.g.*, a data model based on a system that utilizes complex instances of types to describe and/or represent a unit of information). Each type declared in a schema is mapped to a CLR class that supports the SQL UDT contract. It is to be appreciated that although SQL is utilized in the following examples, any suitable database management system can be employed. The types belong to a namespace whose name corresponds to the namespace of the schema with “.Store” suffix appended. The types in a given namespace are compiled into a single assembly that is utilized as a unit of deployment of the schema. The name of the CLR type is equal to the name declared in the schema. For each declared property of the type, the following is added to the corresponding CLR type: 1) a private field with the name equal to the name specified in the schema, pre-fixed by “m_”. The field is attributed with the UDT specific attribute: `System.Data.SqlTypes.SqlUdtField`; 2) a public property with the name equal to the name specified in the schema and corresponding get/set statements. The property is attributed with the UDT specific attribute `System.Data.SqlTypes.SqlUdtProperty`; and 3) the type of the field and the property is the CLR type that corresponds to the type declared in the schema. If the type is one of the scalar types that it is mapped to one of the scalar SQL types (discussed *infra*).

[0051] The following table describes the mapping of the file storage system scalar types to the corresponding SQL managed types:

File Storage System Type	Managed SQL Type	Description
String	SqlString	Variable-length Unicode data with a maximum length of length of 2^{31} characters. Length can be fixed from 1 – 4000 characters or unconstrained using the “max” keyword.
Binary	SqlBinary	Variable-length binary data with a maximum length of length of 2^{32} bytes. Length can be fixed from 1 – 8000 bytes or unconstrained using the “max” keyword.
Boolean	SqlBoolean	Null-able Boolean value
Byte	SqlByte	A single unsigned byte
Int16	SqlInt16	Integer data from -2^{15} (-32,768) through $2^{15} - 1$ (32,767).
Int32	SqlInt32	Integer (whole number) data from -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647).
Int64	SqlInt64	Integer (whole number) data from -2^{63} (-9223372036854775808) through $2^{63} - 1$ (9223372036854775807).
Single	SqlSingle	Floating precision number data from $-3.40E + 38$ through $3.40E + 38$
Double	SqlDouble	Floating precision number data from $-1.79E + 308$ through $1.79E + 308$
Decimal	SqlDecimal	Numeric data types with fixed precision and scale. Precision and Scale attributes apply for properties of type Decimal. Precision defines the maximum number of decimal digits that decimal value. This includes the digits to the left and to the right of the decimal point. The value of the Precision attribute is an integer between 1 and 28. Scale defines the maximum number of decimal digits to the right of the decimal point. The value of the Scale attribute must be between 0 and the value of the Precision attribute. Note for milestone B: these attributes are not supported in this milestone. Precision is fixed to 28 and Scale is fixed to 14.
DateTime	SqlDateTime	Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of one second or 3.33 milliseconds.
Guid	SqlGuid	A globally unique identifier (GUID).
Xml	SqlString	XML is currently stored as a string field. Native Xml

		type support is being considered for post-Beta1.
Stream	TBD	Support planned for post-Beta1. If supported it may replace string(max) and binary(max) declarations.

Table 1

[0052] A database object created in a file storage system store can be stored in a SQL schema name derived from the file storage system schema name. The suffix “.Store” is appended to the file storage system schema name to produce the SQL schema name. For example, the file storage system.storage schema produces objects in the “[System.Storage.Store]” SQL schema such as “[System.Storage.Store].Item.”

[0053] Content in the file storage system can be accessed through views. The views depicted below are read-only, but the subject invention is not so limited since the views can be write-enabled. Each item type can be mapped to a typed view. Each typed item view can be identified in a file storage system store using the naming convention [<schema name>.Store].<Item type name>. A type view for type T can return all items that are type T and all types that derive from T. The view that corresponds to the System.Storage.Item type is [System.Storage.Store].Item. This view can return all the items in a file storage system store. The following table describes the columns of an item type view:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the item
TypeId	[System.Storage.Store].TypeId	Schematized type id
NamespaceName	nvarchar(255)	The unique name of the item
ContainerId	[System.Storage.Store].ItemId	The id of the container item
Item	Item type	Instance of the item type
EntityState	[System.Storage.Store].EntityState	The state information for the item
ObjectSize	Bigint	Size of the item object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information

ItemSyncMetadata	[System.Storage.Store].ItemSyncMetadata	Sync metadata
PathHandle	[System.Storage.Store].BinPathHandle	The handle of the path to the item
PromotionStatus	Int	The status of the promotion for the item

Table 2

[0054] Each link type is mapped to a typed view. Each typed link view is identified in a file storage system store using the naming convention [**<schema name>.Store].<link type name>**]. The view that corresponds to the **System.Storage.Link** type is **[System.Storage.Store].[Link]**. This view can contain all the links in a file storage system store. The following table describes the columns of a link type view:

Column name	Type	Description
SourceRef	[System.Storage.Store].ItemId	ItemId of the source item
LinkId	[System.Storage.Store].LinkId	Id of the link
TargetRef	[System.Storage.Store].ItemId	ItemId of the target item
TypeId	[System.Storage.Store].TypeId	TypeId of the most derived type of the link instance
Link	Link type	Instance of the link type
EntityState	[System.Storage.Store].EntityState	The state information for the link
ObjectSize	Bigint	Size of the link object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
LinkSyncMetadata	[System.Storage.Store].ItemSyncMetadata	Sync metadata
PathHandle	[System.Storage.Store].BinPathHandle	The handle of the path to the item that is the source of the link

Table 3

[0055] All item fragments can be accessible through single view [System.Storage.Store].[ItemFragment]. The following table describes the columns of global item fragment view:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the owning item
SetId	[System.Storage.Store].SetId	The usage of the item fragment
FragmentId	[System.Storage.Store].FragmentId	The Id of the fragment instance
TypeId	[System.Storage.Store].TypeId	TypeId of the most derived type of the ItemFragment instance
ItemFragment	[System.Storage.Store].[ItemFragment]	Instance of the ItemFragment type
EntityState	[System.Storage.Store].EntityState	The state information for the link
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
PathHandle	[System.Storage.Store].BinPathHandle	The handle of the path to the item that is the source of the link

Table 4

[0056] Each store provides a global extension view named [System.Storage.Store].[ItemExtension]. Instances of all extension types are accessible through this view. The following table describes the columns in the global extension view:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the extension
TypeId	[System.Storage.Store].TypeId	Schematized type id
ItemExtension	[System.Storage.Store].ItemExtension	Extension type instance
EntityState	[System.Storage.Store].EntityState	The state

119

		information for the extension
ObjectSize	Bigint	Size of the extension object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
PathHandle	[System.Storage.Store].BinPathHandle	The handle of the path to the item that owns the extension

Table 5

[0057] Each extension type is mapped to a typed view. Each typed extension view is identified in a file storage system store using the naming convention [**<schema name>.Store**].**<extension type name >**]. The following table describes the columns of an extension type view:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the extension
TypeId	[System.Storage.Store].TypeId	Schematized type id
ItemExtension	Extension type	Extension type instance
ObjectSize	Bigint	Size of the extension object (persisted column)
EntityState	[System.Storage.Store].EntityState	The state information for the extension
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
PathHandle	[System.Storage.Store].BinPathHandle	The handle of the path to the item that owns the extension

Table 6

In the file storage system, an inline type instance can be stored within entity type instances. It is to be appreciated and understood that they are accessed by querying the appropriate search view.

psd

[0058] All Items are stored in a single item table, called [System.Storage.Store].[TableItem]. The unique key in the following table is ItemId:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the item
TypeId	[System.Storage.Store].TypeId	Schematized type id
NamespaceName	Nvarchar(255)	The unique name of the item
ContainerId	[System.Storage.Store].ItemId	The id of the container item
Item	Item type	The item
EntityState	[System.Storage.Store].EntityState	EntityState udt which contains meta information about the item.
SDId	Int	Internal use only (used by security sub-system)
SDLastUpdateLocalTS	Bigint	The timestamp of the last change of the SDId. Used for change tracking.
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
ItemSyncMetadata	[System.Storage.Store].ItemSyncMetadata	Sync metadata for global Ids
TombstoneStatus	Int	Tombstone status
LastUpdateLocalTS	Bigint	Internal use only (used for indexing). This column is actually mapped to ChangeInformation.LastUpdateLocalTS. We need to map it in order to create an index on it.
MaxOrd	int	Internal use only (used to compute

tot

		PathHandle for new items added to this container item)
TypePath	Hierarchical_Type_Id	Internal use only (persisted column)
ObjectSize	Bigint	Size of the item object (persisted column)
PathHandle	[System.Storage.Store].BinPathHandle	The path handle to this item
PromotionStatus	Int	The promotion status for the item
LastAccessTime	DateTime	The last access time for the file stream associated with the item.
StreamSize	Bigint	The size of the stream associated with the item
AllocationSize	Bigint	The allocated size for the stream associated with the item

Table 7

[0059] The indexes on the item table are described in the following table:

Columns	Unique	Clustered	Included Columns
TypePath, ItemId	Yes	Yes	
ItemId	Yes	No	TombstoneStatus, TypePath, SDId
PathHandle	Yes	No	SDId, TombstoneStatus
ContainerId, NamespaceName	Yes	No	SDId, TombstoneStatus
LastUpdateLocalTS	No	No	SDId, TombstoneStatus, PathHandle

Table 8

122

[0060] All links will be stored in the link table, which is named [System.Storage.Store].[Table!Link]. The unique key in the following table is ItemId, LinkId:

Column name	Type	Description
SourceRef	[System.Storage.Store].ItemId	ItemId of the source item
LinkId	[System.Storage.Store].LinkId	Id of the link
TargetRef	[System.Storage.Store].ItemId	ItemId of the target item
TypeId	[System.Storage.Store].TypeId	Schematized type id
TypePath	Hierarchical_Type_Id	Internal use only (persisted column)
Link	Link type	The link object
EntityState	[System.Storage.Store].EntityState	State information about the link
SDId	Int	Internal use only
TombstoneStatus	Int	Tombstone status
ObjectSize	Bigint	Size of the link object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
LinkSyncMetadata	[System.Storage.Store].LinkSyncMetadata	Sync metadata
LastUpdateLocalTS	Bigint	Internal use only (used for indexing). This column is actually mapped to ChangeInformation. LastUpdateLocalTS. We need to map it in order to create an index on it.
PathHandle	[System.Storage.Store].BinPathHandle	The path handle to the item that is the source of the link

Table 9

[0061] The indexes on the Link table are described in the following table:

TAB

Columns	Unique	Clustered	Included Columns
SourceRef, LinkId	Yes	Yes	
TypePath	No	No	SDId, TombstoneStatus
SourceRef, TypePath	No	No	SDId, TombstoneStatus
PathHandle	No	No	SDId, TombstoneStatus
TargetRef	No	No	SDId, TombstoneStatus
LastUpdateLocalTS	No	No	SDId, TombstoneStatus, PathHandle

Table 10

[0062] All EntityExtensions are stored in a single table called:
[System.Storage.Store].[TableItemExtension]. The following table describes item extension table:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the extension
TypeId	[System.Storage.Store].TypeId	Schematized type id
TypePath	Hierarchical_Type_Id	Internal use only (persisted column)
ItemExtension	[System.Storage.Store].ItemExtension	The ItemExtension object
EntityState	[System.Storage.Store].EntityState	EntityState udt which contains meta information about the EntityExtension.
SDId	Int	Internal use only
TombstoneStatus	Int	Tombstone status
ObjectSize	Bigint	Size of the EntityExtension object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
LastUpdateLocalTS	Bigint	Internal use only (used for indexing). This column is actually mapped to ChangeInformation. LastUpdateLocalTS. We need to map it in order to create an

124

		index on it.
PathHandle	[System.Storage.Store].BinPathHandle	The path handle to the item owning this EntityExtension

Table 11

[0063] The indexes on the ItemExtension table are described in the following table:

Columns	Unique	Clustered	Included Columns
TypePath, ItemId, TypeId	Yes	Yes	
ItemId, TypeId	No	No	SDId, TombstoneStatus, PathHandle
PathHandle	Yes	No	SDId, TombstoneStatus
LastUpdateLocalTS	No	No	SDId, TombstoneStatus, PathHandle

Table 12

[0064] All ItemFragment type instances are stored in a single table named [System.Storage.Store].[Table!ItemFragment]. The following table depicts the ItemFragment type:

Column name	Type	Description
ItemId	[System.Storage.Store].ItemId	ItemId of the owning item
SetId	[System.Storage.Store].SetId	The usage of the item fragment
FragmentId	[System.Storage.Store].FragmentId	The Id of the fragment instance
ItemFragment	[System.Storage.Store].ItemFragment	The ItemFragment object
TypeId	[System.Storage.Store].TypeId	Schematized type id
TypePath	Hierarchical_Type_Id	Internal use only (persisted column)
EntityState	[System.Storage.Store].EntityState	EntityState udt which contains meta information about the fragment.
SDId	int	Internal use only
TombstoneStatus	int	Tombstone status

125

ObjectSize	bigint	Size of the ItemFragment object (persisted column)
ChangeInformation	[System.Storage.Store].ChangeInformation	Change tracking information
LastUpdateLocalTS	bigint	Internal use only (used for indexing). This column is actually mapped to ChangeInformation. LastUpdateLocalTS. We need to map it in order to create an index on it.
PathHandle	[System.Storage.Store].BinPathHandle	The path handle to the owning item

Table 13

[0065] The indexes on the ItemFragment table are described in the following table:

Column name	Unique	Clustered	Included Columns
TypePath, ItemId, SetId, FragmentId	Yes	Yes	
ItemId, SetId, FragmentId	Yes	No	SDId, TombstoneStatus, PathHandle
PathHandle	Yes	No	SDId, TombstoneStatus
LastUpdateLocalTS	No	No	SDId, TombstoneStatus, PathHandle

Table 14

[0066] Figs. 9-10 illustrate methodologies in accordance with the subject invention. For simplicity of explanation, the methodologies are depicted and described as a series of acts. It is to be understood and appreciated that the subject invention is not limited by the acts illustrated and/or by the order of acts, for example acts can occur in various orders and/or concurrently, and with other acts not presented and described herein. Furthermore, not all illustrated acts may be required to implement the methodologies in accordance with the subject invention. In addition, those skilled in the

art will understand and appreciate that the methodologies could alternatively be represented as a series of interrelated states *via* a state diagram or events.

[0067] Fig. 9 illustrates a methodology 900 that facilitates storing a type instance that maps to a data model and/or provides a query. The data model can be a file storage data model that enables storing, finding, and relating information. An information type can be, but is not limited to, a document, an image, a video, a contact, a message, an email, an audio clip, *etc.* These informational types (*e.g.*, units of information) can be represented as instances of complex types that are part of a type system that supports inheritance, wherein inheritance allows objects to inherit attributes and/or behaviors from other objects. It is to be appreciated that inheritance can be considered a hierarchical structure and/or format.

[0068] At reference numeral 902, data can be received, wherein the data can include a type, a criteria, a schema, a query criteria, The type instance can be stored to map to a data model (*e.g.*, a file storage data model) at reference numeral 904. For instance, the model can map types described in the schema to user defined types and database objects. The storage mapping can describe the database objects that are created based on a schema definition and how instances of types described in the schema are stored and/or accessed. In one example, a database structure can be designed such that the instances of types and the rules for mapping of type declarations into database objects can be provided.

[0069] At reference numeral 906, a query can be invoked to satisfy at least one criterion to efficiently and effectively find items, documents, and contacts. For instance, storing the type instance that relates to information types can utilize the query to provide at least one of the following: 1) find at least one item in the file storage data model 102 that satisfy a certain criteria; 2) find at least one document in the file storage data model 102 that satisfy a particular criteria; and 3) find at least one contact (*e.g.*, including an individual, organization, and group) that satisfy a certain criteria.

[0070] Fig. 10 illustrates a methodology 1000 that facilitates storing a type instance and/or providing a query. At reference numeral 1002, data is obtained and/or received. The data can include a type, a criteria, a schema, a query criteria, *etc.* At 1004, a database engine can be utilized to provide at least one mechanism for storing the type

instances and/or for querying. For instance, a relational database engine can be utilized to provide a relational storage and a relation query capability. The relational database engine can utilize a collection of data items organized as a set of formally described tables. The data within the tables can be accessed and/or reassembled in various ways without the requirement of reorganizing the database tables. Furthermore, the relational database can be easily extended, such as adding new categories without modifications to existing applications and/or data.

[0071] At reference numeral 1006, a type instance is stored that maps to a data model, wherein the data model can be a file storage system. The storage mapping describes the database objects that are created based on the schema and how instances of types described in the schema are stored and/or accessed. The mapping can be of types described in the schema, wherein such mapping is to user defined types and database objects. At reference numeral 1008, a query can be provided to find at least one of an item, a document, and a contact. The query can be utilized to search based at least in part upon a criterion. Moreover, at reference numeral 1010, a view can be employed to expose the instances of the type. For instance, the type system can be hierarchical in structure, wherein a view can be generated to expose any instance of the particular type. In other words, because of the hierarchy, the view is associated with a given type that projects a subset of the particular types of the view associated with its base type.

[0072] In order to provide additional context for implementing various aspects of the subject invention, Figs. 11-12 and the following discussion is intended to provide a brief, general description of a suitable computing environment in which the various aspects of the subject invention may be implemented. While the invention has been described above in the general context of computer-executable instructions of a computer program that runs on a local computer and/or remote computer, those skilled in the art will recognize that the invention also may be implemented in combination with other program modules. Generally, program modules include routines, programs, components, data structures, *etc.*, that perform particular tasks and/or implement particular abstract data types.

[0073] Moreover, those skilled in the art will appreciate that the inventive methods may be practiced with other computer system configurations, including single-

processor or multi-processor computer systems, minicomputers, mainframe computers, as well as personal computers, hand-held computing devices, microprocessor-based and/or programmable consumer electronics, and the like, each of which may operatively communicate with one or more associated devices. The illustrated aspects of the invention may also be practiced in distributed computing environments where certain tasks are performed by remote processing devices that are linked through a communications network. However, some, if not all, aspects of the invention may be practiced on stand-alone computers. In a distributed computing environment, program modules may be located in local and/or remote memory storage devices.

[0074] Fig. 11 is a schematic block diagram of a sample-computing environment 1100 with which the subject invention can interact. The system 1100 includes one or more client(s) 1110. The client(s) 1110 can be hardware and/or software (*e.g.*, threads, processes, computing devices). The system 1100 also includes one or more server(s) 1120. The server(s) 1120 can be hardware and/or software (*e.g.*, threads, processes, computing devices). The servers 1120 can house threads to perform transformations by employing the subject invention, for example.

[0075] One possible communication between a client 1110 and a server 1120 can be in the form of a data packet adapted to be transmitted between two or more computer processes. The system 1100 includes a communication framework 1140 that can be employed to facilitate communications between the client(s) 1110 and the server(s) 1120. The client(s) 1110 are operably connected to one or more client data store(s) 1150 that can be employed to store information local to the client(s) 1110. Similarly, the server(s) 1120 are operably connected to one or more server data store(s) 1130 that can be employed to store information local to the servers 1140.

[0076] With reference to Fig. 12, an exemplary environment 1200 for implementing various aspects of the invention includes a computer 1212. The computer 1212 includes a processing unit 1214, a system memory 1216, and a system bus 1218. The system bus 1218 couples system components including, but not limited to, the system memory 1216 to the processing unit 1214. The processing unit 1214 can be any of various available processors. Dual microprocessors and other multiprocessor architectures also can be employed as the processing unit 1214.

[0077] The system bus 1218 can be any of several types of bus structure(s) including the memory bus or memory controller, a peripheral bus or external bus, and/or a local bus using any variety of available bus architectures including, but not limited to, Industrial Standard Architecture (ISA), Micro-Channel Architecture (MSA), Extended ISA (EISA), Intelligent Drive Electronics (IDE), VESA Local Bus (VLB), Peripheral Component Interconnect (PCI), Card Bus, Universal Serial Bus (USB), Advanced Graphics Port (AGP), Personal Computer Memory Card International Association bus (PCMCIA), Firewire (IEEE 1394), and Small Computer Systems Interface (SCSI).

[0078] The system memory 1216 includes volatile memory 1220 and nonvolatile memory 1222. The basic input/output system (BIOS), containing the basic routines to transfer information between elements within the computer 1212, such as during start-up, is stored in nonvolatile memory 1222. By way of illustration, and not limitation, nonvolatile memory 1222 can include read only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), or flash memory. Volatile memory 1220 includes random access memory (RAM), which acts as external cache memory. By way of illustration and not limitation, RAM is available in many forms such as static RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), double data rate SDRAM (DDR SDRAM), enhanced SDRAM (ESDRAM), Synchlink DRAM (SLDRAM), Rambus direct RAM (RDRAM), direct Rambus dynamic RAM (DRDRAM), and Rambus dynamic RAM (RDRAM).

[0079] Computer 1212 also includes removable/non-removable, volatile/non-volatile computer storage media. Fig. 12 illustrates, for example a disk storage 1224. Disk storage 1224 includes, but is not limited to, devices like a magnetic disk drive, floppy disk drive, tape drive, Jaz drive, Zip drive, LS-100 drive, flash memory card, or memory stick. In addition, disk storage 1224 can include storage media separately or in combination with other storage media including, but not limited to, an optical disk drive such as a compact disk ROM device (CD-ROM), CD recordable drive (CD-R Drive), CD rewritable drive (CD-RW Drive) or a digital versatile disk ROM drive (DVD-ROM). To facilitate connection of the disk storage devices 1224 to the system bus 1218, a removable or non-removable interface is typically used such as interface 1226.

[0080] It is to be appreciated that Fig. 12 describes software that acts as an intermediary between users and the basic computer resources described in the suitable operating environment 1200. Such software includes an operating system 1228. Operating system 1228, which can be stored on disk storage 1224, acts to control and allocate resources of the computer system 1212. System applications 1230 take advantage of the management of resources by operating system 1228 through program modules 1232 and program data 1234 stored either in system memory 1216 or on disk storage 1224. It is to be appreciated that the subject invention can be implemented with various operating systems or combinations of operating systems.

[0081] A user enters commands or information into the computer 1212 through input device(s) 1236. Input devices 1236 include, but are not limited to, a pointing device such as a mouse, trackball, stylus, touch pad, keyboard, microphone, joystick, game pad, satellite dish, scanner, TV tuner card, digital camera, digital video camera, web camera, and the like. These and other input devices connect to the processing unit 1214 through the system bus 1218 via interface port(s) 1238. Interface port(s) 1238 include, for example, a serial port, a parallel port, a game port, and a universal serial bus (USB). Output device(s) 1240 use some of the same type of ports as input device(s) 1236. Thus, for example, a USB port may be used to provide input to computer 1212, and to output information from computer 1212 to an output device 1240. Output adapter 1242 is provided to illustrate that there are some output devices 1240 like monitors, speakers, and printers, among other output devices 1240, which require special adapters. The output adapters 1242 include, by way of illustration and not limitation, video and sound cards that provide a means of connection between the output device 1240 and the system bus 1218. It should be noted that other devices and/or systems of devices provide both input and output capabilities such as remote computer(s) 1244.

[0082] Computer 1212 can operate in a networked environment using logical connections to one or more remote computers, such as remote computer(s) 1244. The remote computer(s) 1244 can be a personal computer, a server, a router, a network PC, a workstation, a microprocessor based appliance, a peer device or other common network node and the like, and typically includes many or all of the elements described relative to computer 1212. For purposes of brevity, only a memory storage device 1246 is

illustrated with remote computer(s) 1244. Remote computer(s) 1244 is logically connected to computer 1212 through a network interface 1248 and then physically connected *via* communication connection 1250. Network interface 1248 encompasses wire and/or wireless communication networks such as local-area networks (LAN) and wide-area networks (WAN). LAN technologies include Fiber Distributed Data Interface (FDDI), Copper Distributed Data Interface (CDDI), Ethernet, Token Ring and the like. WAN technologies include, but are not limited to, point-to-point links, circuit switching networks like Integrated Services Digital Networks (ISDN) and variations thereon, packet switching networks, and Digital Subscriber Lines (DSL).

[0083] Communication connection(s) 1250 refers to the hardware/software employed to connect the network interface 1248 to the bus 1218. While communication connection 1250 is shown for illustrative clarity inside computer 1212, it can also be external to computer 1212. The hardware/software necessary for connection to the network interface 1248 includes, for exemplary purposes only, internal and external technologies such as, modems including regular telephone grade modems, cable modems and DSL modems, ISDN adapters, and Ethernet cards.

[0084] What has been described above includes examples of the subject invention. It is, of course, not possible to describe every conceivable combination of components or methodologies for purposes of describing the subject invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the subject invention are possible. Accordingly, the subject invention is intended to embrace all such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

[0085] In particular and in regard to the various functions performed by the above described components, devices, circuits, systems and the like, the terms (including a reference to a "means") used to describe such components are intended to correspond, unless otherwise indicated, to any component which performs the specified function of the described component (*e.g.*, a functional equivalent), even though not structurally equivalent to the disclosed structure, which performs the function in the herein illustrated exemplary aspects of the invention. In this regard, it will also be recognized that the invention includes a system as well as a computer-readable medium having computer-

executable instructions for performing the acts and/or events of the various methods of the invention.

[0086] In addition, while a particular feature of the invention may have been disclosed with respect to only one of several implementations, such feature may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application. Furthermore, to the extent that the terms “includes,” and “including” and variants thereof are used in either the detailed description or the claims, these terms are intended to be inclusive in a manner similar to the term “comprising.”

CLAIMS

What is claimed is:

- 1. A system that facilitates mapping a data model, comprising:
a file storage data model that is utilized to define a mapping schema; and
a type storage component that maps a schema object to a database object based on the associated mapping schema, wherein an instance of a type in the mapping schema is stored.**
- 2. The system of claim 1, the instance of the type is at least one of: a document; an image; music; a video; a contact; a message; a person; an organization; an email; a fax; a voice; and an audio clip.**
- 3. The system of claim 1, the type storage component further provides a query to find at least one of the following: at least one item in the system that satisfies a respective criteria; at least one document in the system that satisfies a respective criteria; and at least one contact that satisfies a respective criteria.**
- 4. The system of claim 3, the contact can include at least one of the following: an individual; an organization; and a group.**
- 5. The system of claim 1, further comprising a relational component that utilizes at least one of a relational storage and a relational query capability.**
- 6. The system of claim 5, the relational component utilizes a relational database technique that is a collection of data items organized as a set of formally described tables, wherein data can be at least one of accessed without reorganization and reassembled without reorganization.**

7. The system of claim 1, the mapping describes at least one of the database objects created based on the schema and how the instance of the type described in the schema is stored and accessed.
8. The system of claim 1, the type in the schema maps to a common language runtime (CLR) class in a storage of the file storage data model.
9. The system of claim 8, the type is at least one of the following: an item; an item extension; an item fragment; and a link.
10. The system of claim 1, further comprising a table that has at least one of the following: 1) a column that contains at least one instance of the respective type; and 2) a row that contains a serialized representation of the CLR class instance that represents the type instance.
11. The system of claim 1, further comprising a view component that generates a view that exposes at least one instance of the type, wherein the view projects a subset of the respective type of the view associated with a base type.
12. The system of claim 1, the type contains at least one of the following a type hierarchy and an inheritance.
13. A computer readable medium having stored thereon the components of the system of claim 1.
14. A computer-implemented method that facilitates mapping a data model, comprising:
 - obtaining a file storage data model;
 - receiving at least one of a schema, a type, a criteria, and a query information;
 - mapping a schema object to a database object based on the schema associated to the file storage data model; and
 - storing an instance of the type from the schema.

15. The method of claim 14, further comprising querying to find at least one of the following: at least one item in the system that satisfies a respective criteria; at least one document in the system that satisfies a respective criteria; and at least one contact that satisfies a respective criteria.

16. The method of claim 14, further comprising utilizing a relational database engine to provide a relational storage and a relational query capability.

17. The method of claim 14, further comprising generating a view that exposes at least one instance of the type, wherein the view projects a subset of the respective type of the view associated with a base type.

18. A data packet that communicates between a file storage data model and a type storage component, the data packet facilitates the method of claim 14.

19. A computer-implemented system that facilitates mapping a data model, comprising:

means for receiving at least one of the following a schema, a type, a criteria, and a query criteria; and

means for mapping a schema object to a database object based on a schema associated to a file storage data model, wherein an instance of a type in the schema is stored.

20. The system of claim 11, the view component invokes a user interface to interact between a user and the type storage component.

ABSTRACT

The subject invention provides a system and/or a method that facilitates mapping a data base model to a database object. A type storage system can utilize a storage mapping of a file storage data model. The mapping can describe the database object created based at least in part upon a schema and how instances of the type described in the schema are stored and/or accessed. Furthermore, a query can be provided to find at least one of an item, a document, and/or a contact that satisfy at least one criterion. The type storage system can receive data, wherein the data is at least one of a schema, a data model, a type, a query, and query criteria *via* an interface to provide the storing and querying. Additionally, the type storage system can generate a view that exposes at least one instance of the type.

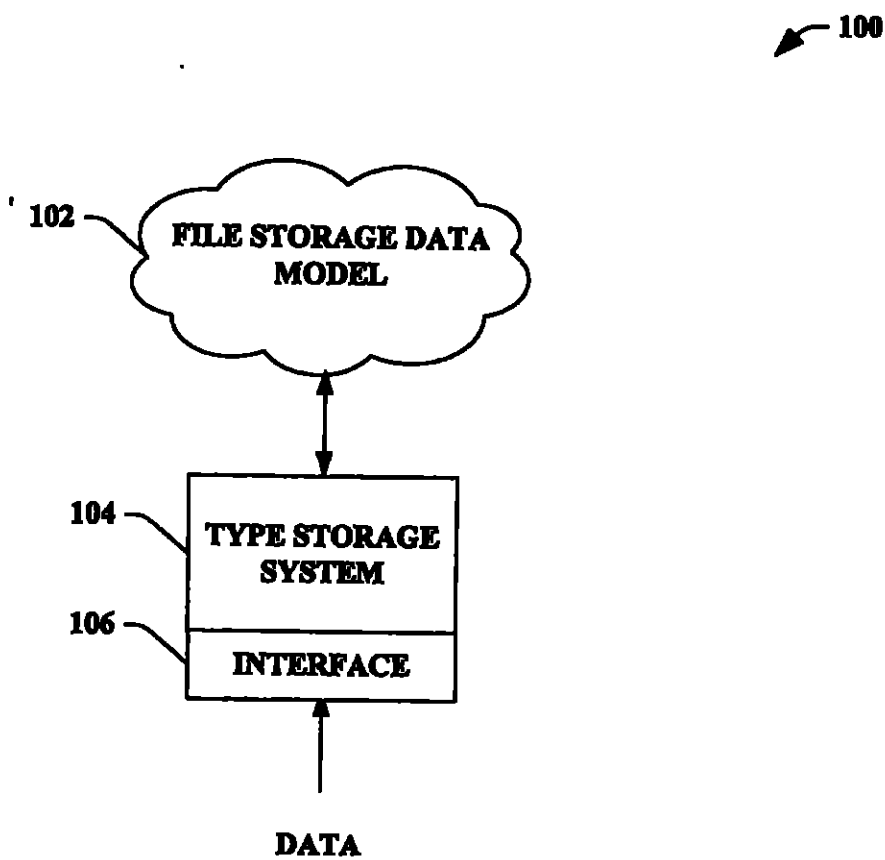


FIG. 1

138

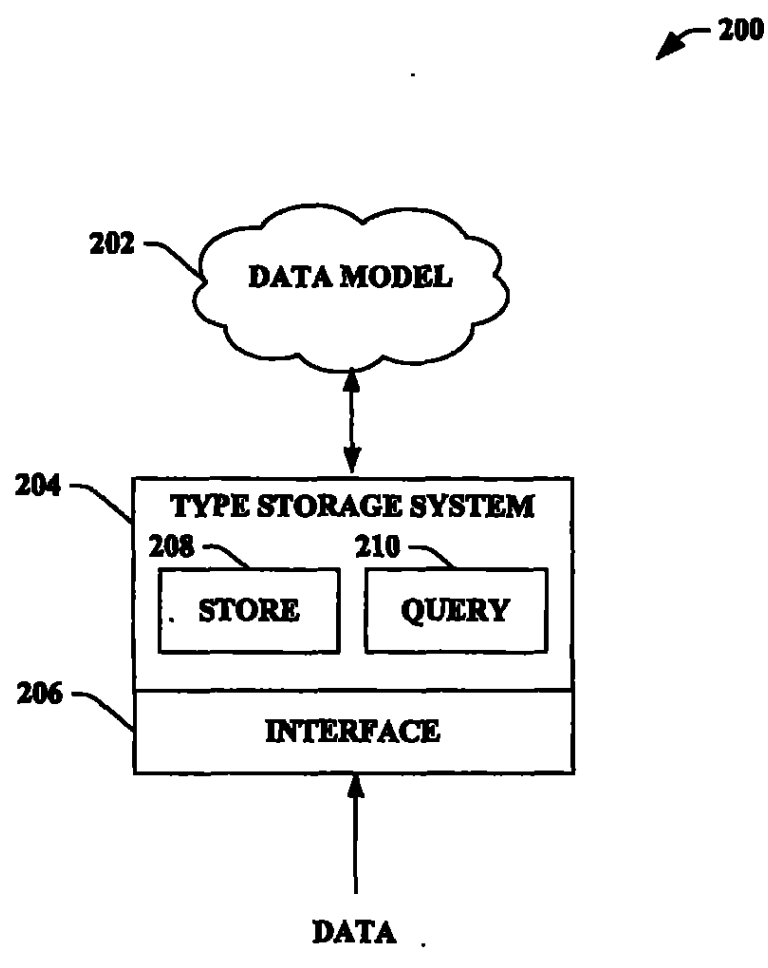


FIG. 2

157

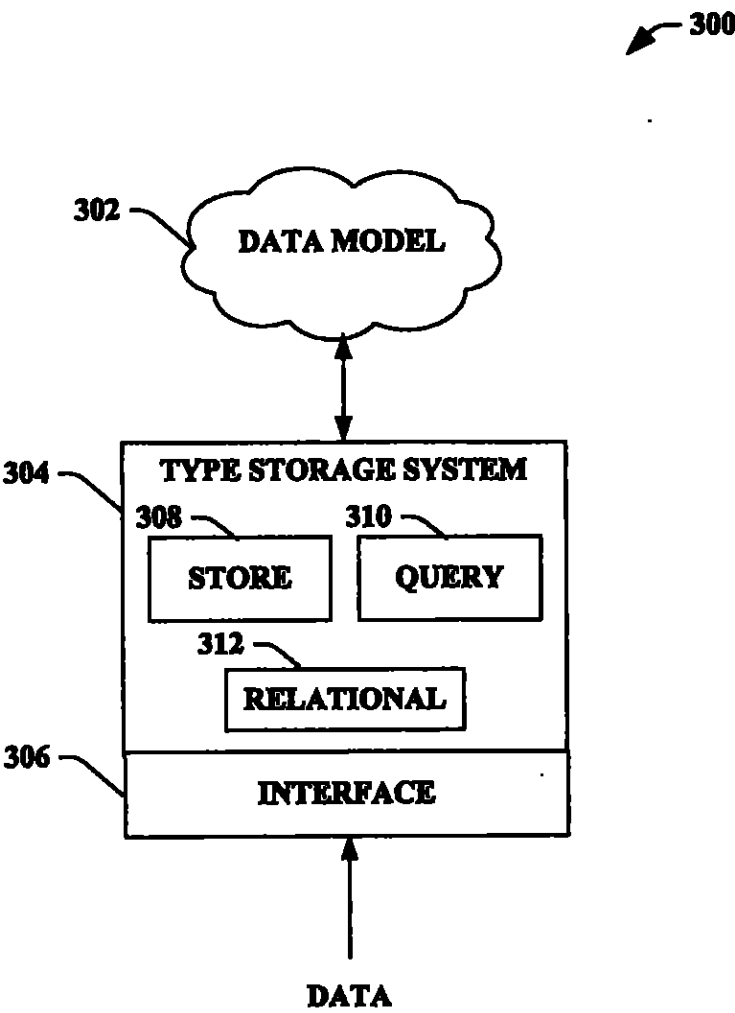


FIG. 3

140

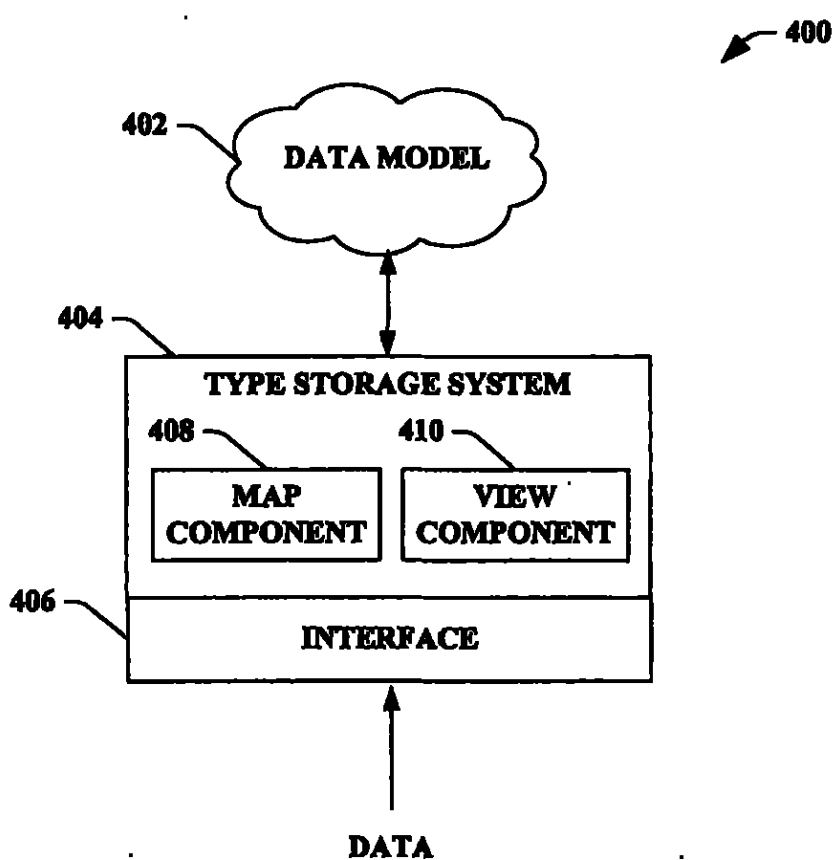


FIG. 4

[Handwritten signature]

5/12

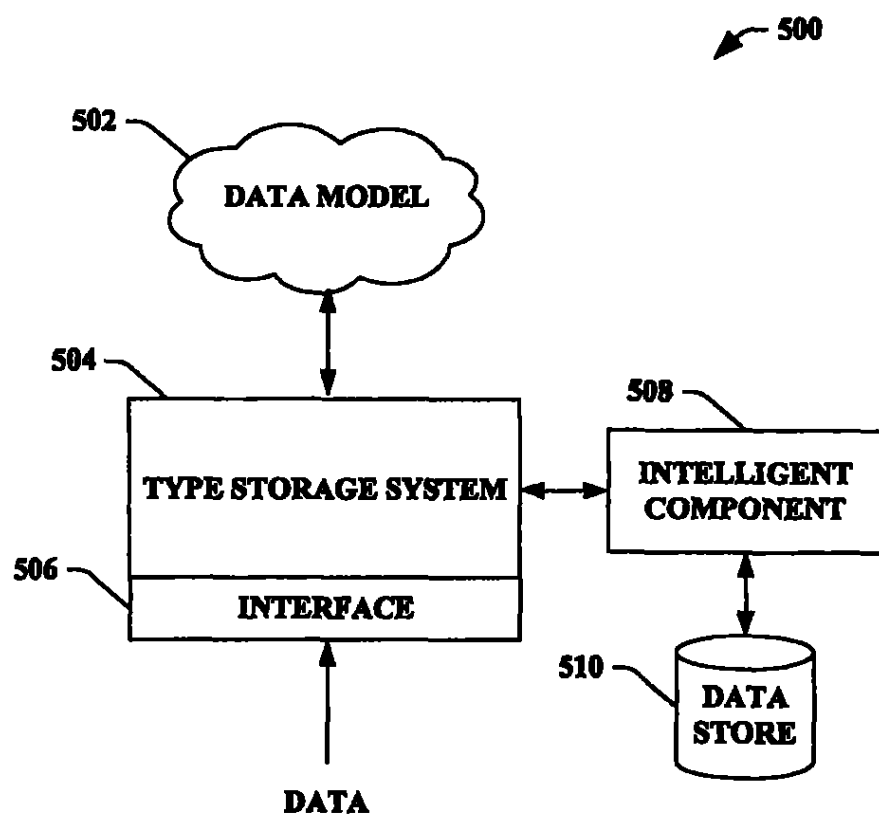


FIG. 5

142

600

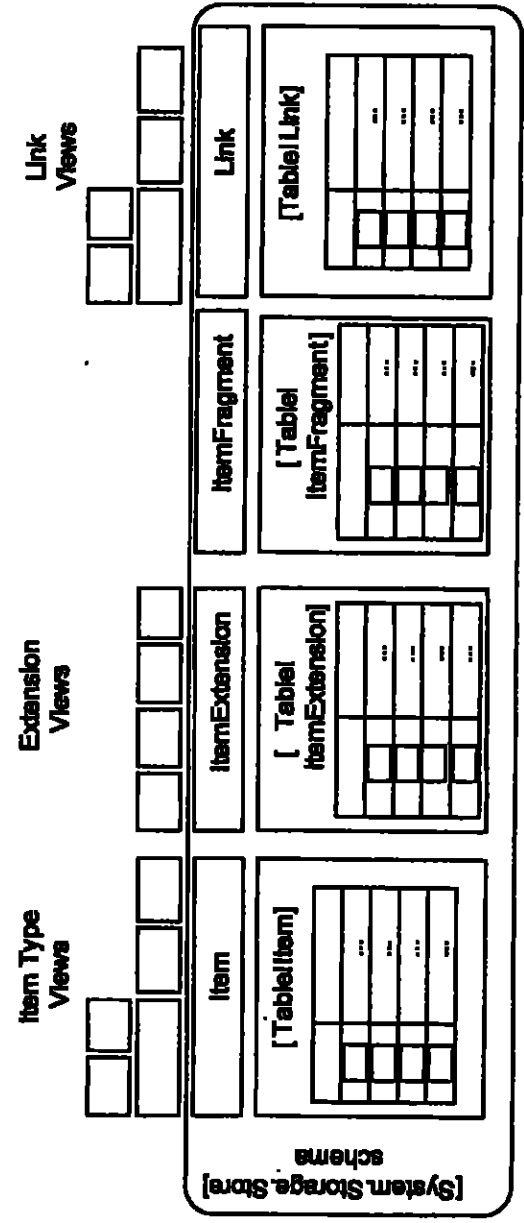


FIG. 6

7-13

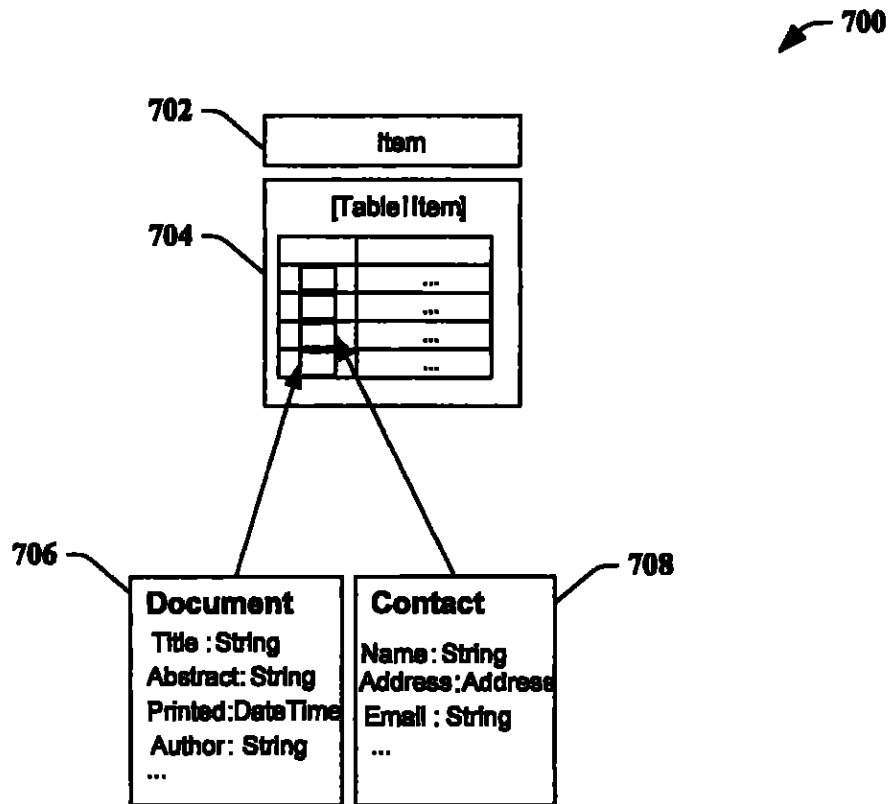


FIG. 7

7/12

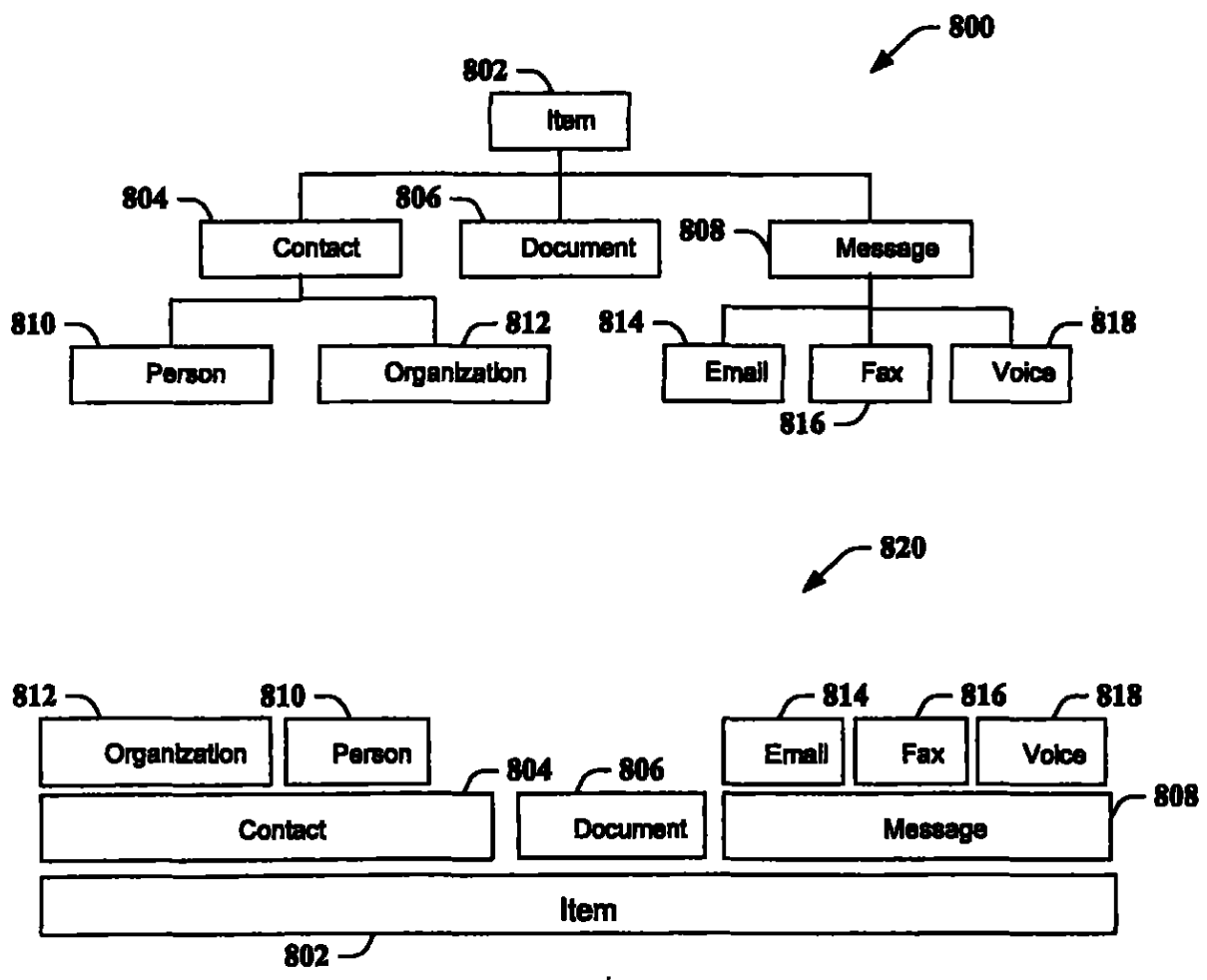
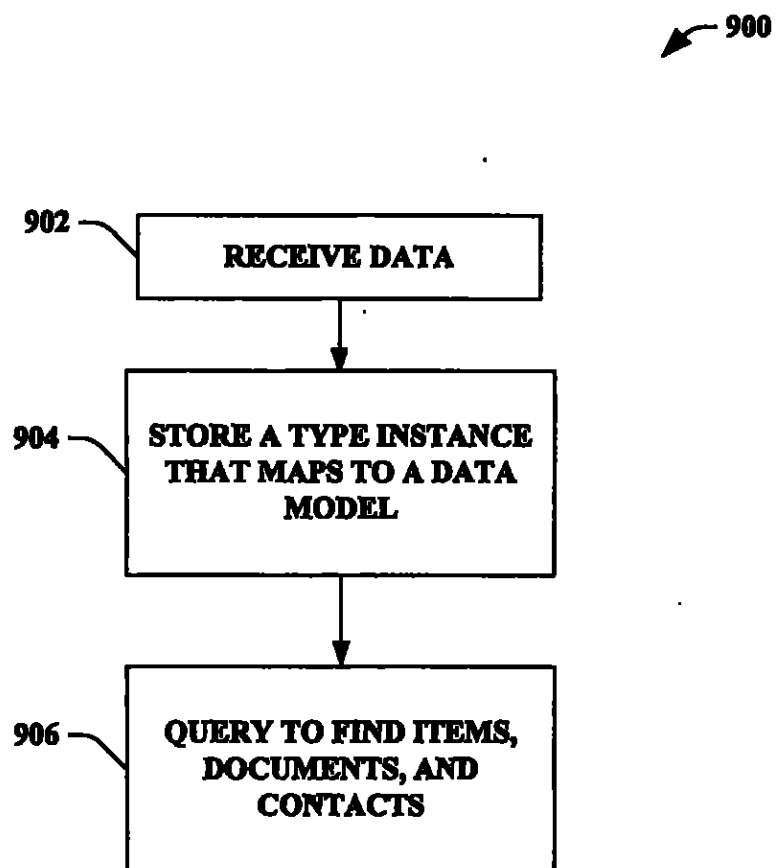


FIG. 8

145-

9/12

**FIG. 9**

748

10/12

1000

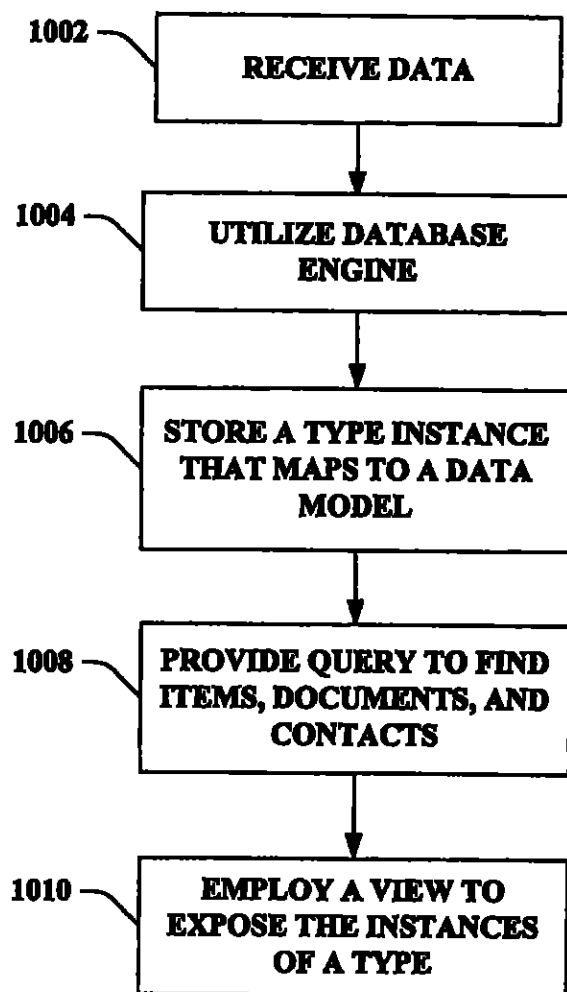


FIG. 10

190

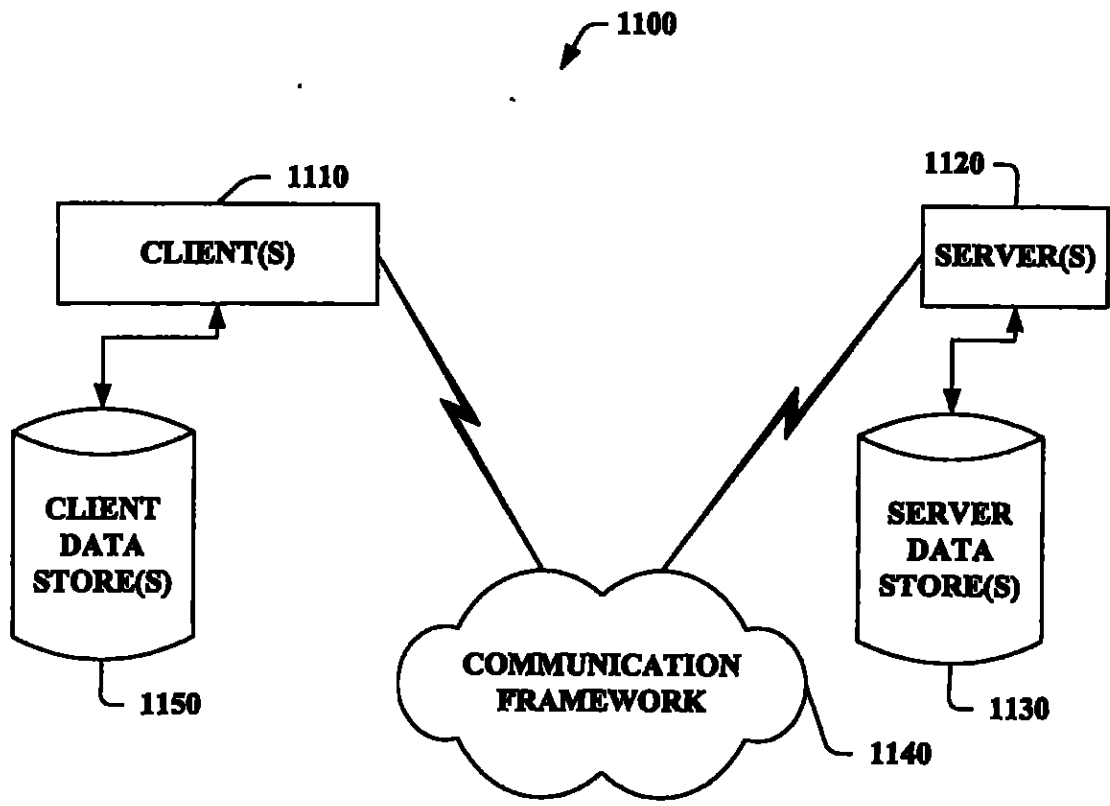


FIG. 11

748

12/12

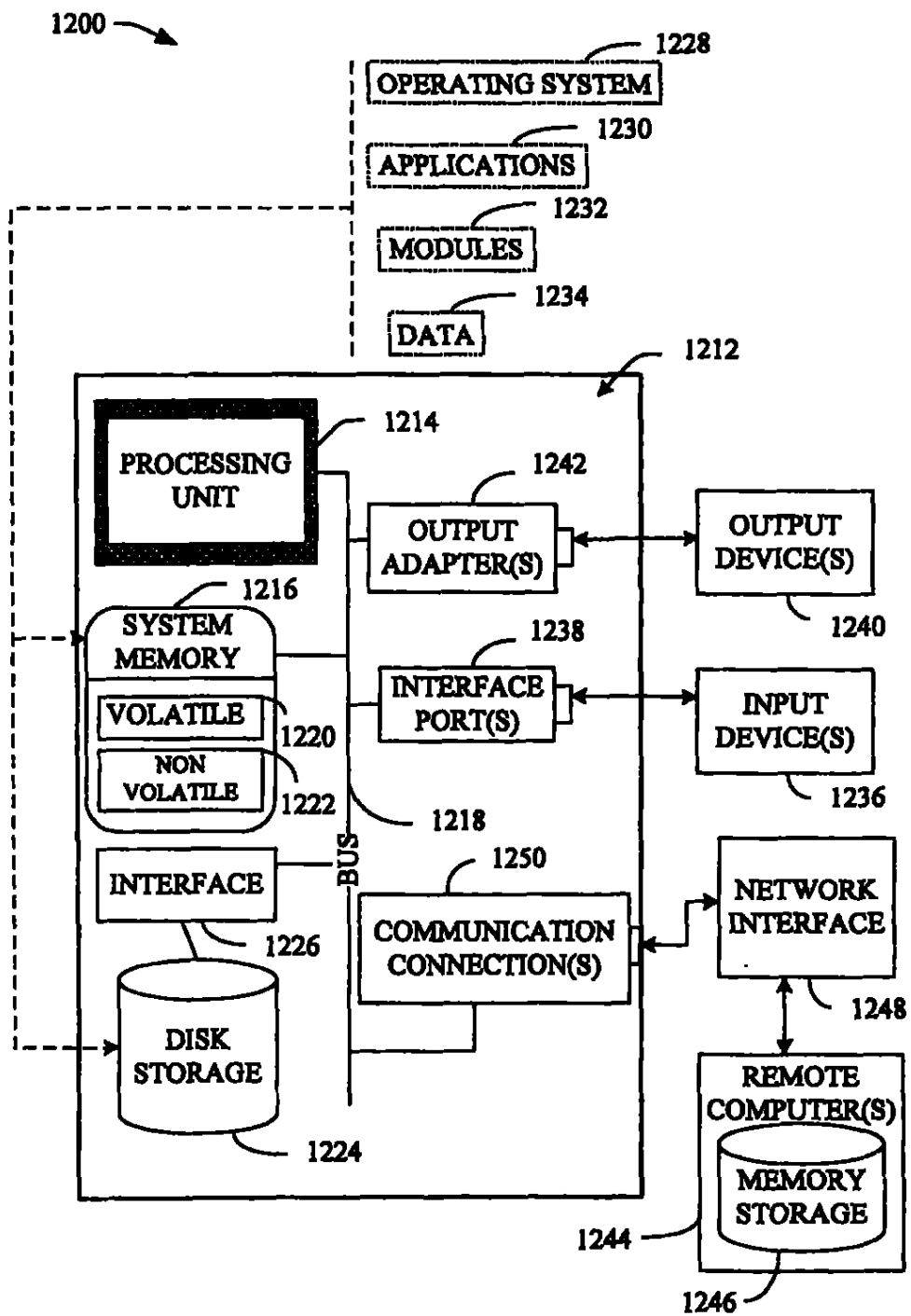


FIG. 12

149



Industria y Comercio SUPERINTENDENCIA

Delegatura de Propiedad Industrial División de Nuevas Creaciones

REQUISITOS NECESARIOS PARA PRESENTACIÓN Y AGILIZACIÓN DEL TRAMITE

PATENTE DE INVENCION (✓)	MODELO DE UTILIDAD ()	USUARIO ()	RADICADOR ()
-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. NC?	()	()	(X)
-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 se presenta la solicitud.	()	()
-Representación gráfica y fotográfica del Diseño Industrial o muestra del Material que incorpora el diseño.	()	()
-Comprobante de Pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	()

ESQUEMA DE TRAZADO ()

-Indicación de que se solicita el registro de Diseño Industrial.	()	()
-Datos de identificación del solicitante o de la persona que se presenta la solicitud.	()	()
-Representación gráfica del Esquema Trazado.	()	()
-Comprobante de Pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	()

Firma Responsable:

Fecha:

febrero 28 de 2006

151

PROTOCOLO

Expediente 06 - 19553

**SUPERINDUSTRIA Y COMERCIO
SISTEMA DE REGISTRO**

MANTENIMIENTO DE PROTOCOLOS

Numero : 16089

Fecha : 04/08/1995

Conferido: MICROSOFT CORPORATION

Pais : US ESTADOS UNIDOS DE AMERICA

Ciudad : 234 REDMOND, WASHINGTON

Represent: S Sustituir: S Desistir: S

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

Isencia	Codigo	Ctr	Nombre
1	77	A	CARDENAS NAVAS DARIO
2	3653		CARDENAS CABALLERO EDUARDO
3	5048		CARDENAS MARTINEZ BERNARDO

Encontrados (1/1) registro(s)

**Al contestar favor indique el número de
radicación consignado en el sticker**

**Sede Centro: Carrera 13 No. 27-00 Pisos 2, 5, 7 y 10
Commutador: 3 820840 Fax: 3 505220
Sede CAN: Tr. 40A No. 38-50 3153265 al 69
Fax: 3 505220. Línea 9800-910 165
Web: www.sic.gov.co e-mail: info@sic.gov.co
Bogotá D.C. - Colombia**

152

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Bogotá,
2020

Doctor(a)
CARDENAS NAVAS DARIO
Apoderado(a) y/o Representante de
MICROSOFT CORPORATION

REFERENCIA	Radicación No. 6 19553
	Trámite 2
	Evento 1
	Actuación 430
	Oficio No. 2854

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

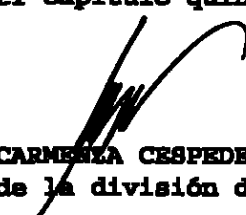
1. Debe allegar copia del documento en que consta la cesión del derecho a la patente otorgado por el(los) inventor(s) al(los) solicitante(s) o a su(s) causante(s) (art. 26-k, Dec 486/2000).

2. De la figura característica enviar por duplicado arte final en tamaño 12 x 12 cm.

3. Para efectos de la publicación en la gaceta se recomienda allegar el extracto diligenciado.

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

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


ALIX CARMEN CESPEDES DE VERGEL
Jefe de la división de Nuevas Creaciones

El anterior requerimiento fue notificado por fijación en lista de fecha 09 MAR. 2006 adpai

Se recuerda al interesado que debe presentar, dentro de los dieciséis meses siguientes contados a partir de la fecha de presentación de la solicitud cuya prioridad se invoca, copia de la misma certificada por la autoridad que la expidió

Continúa en la página siguiente...

REPUBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Radicación No.:6 19553

y, . en los casos en que haya lugar, la traducción simple del capítulo reivindicatorio. El incumplimiento de éste plazo, acarreará la pérdida de la prioridad invocada.

✱