

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 702 611**

51 Int. Cl.:

G06F 17/30 (2006.01)

G06F 7/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **20.11.2007 PCT/US2007/085202**

87 Fecha y número de publicación internacional: **29.05.2008 WO08064207**

96 Fecha de presentación y número de la solicitud europea: **20.11.2007 E 07864644 (5)**

97 Fecha y número de publicación de la concesión europea: **17.10.2018 EP 2084597**

54 Título: **Crear datos en un almacenamiento de datos usando una ontología dinámica**

30 Prioridad:

20.11.2006 US 602626

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

04.03.2019

73 Titular/es:

**PALANTIR TECHNOLOGIES, INC. (100.0%)
1530 Page Mill Road
Palo Alto, CA 94304, US**

72 Inventor/es:

**JAIN, AKASH;
MCGREW, ROBERT J. y
GETTINGS, NATHAN**

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 702 611 T3

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

DESCRIPCIÓN

Crear datos en un almacenamiento de datos usando una ontología dinámica

5 Campo de la invención

La presente divulgación se refiere en general a técnicas para crear datos en un almacenamiento de datos.

Antecedentes

10

Los enfoques descritos en esta sección son enfoques que podrían perseguirse, pero no necesariamente enfoques que han sido anteriormente concebidos o perseguidos. Por lo tanto, a no ser que se indique lo contrario, no debería suponerse que ninguno de los enfoques descritos en esta sección se califican como técnica anterior meramente mediante su inclusión en esta sección.

15

Sistemas de base de datos basados en ordenador, tal como sistemas de gestión de base de datos relacionales, habitualmente organizan datos de acuerdo con una estructura fija de tablas y relaciones. La estructura puede describirse usando una ontología, incorporada en un esquema de base de datos, que comprende un modelo de datos que se usa para representar la estructura y razón de objetos en la estructura.

20

Una ontología de una base de datos normalmente se fija en el momento en que se crea la base de datos. Cualquier cambio en la ontología representada por el esquema habitualmente es extremadamente disruptivo al sistema de base de datos y requiere un administrador de base de datos para modificar tablas o relaciones, o crear nuevas tablas o relaciones.

25

La rigidez de la ontología de base de datos típica es un inconveniente grave para organizaciones que requieren técnicas de procesamiento de datos flexibles y dinámicas de acuerdo con cambios en los datos que se recopilan. Por ejemplo, el análisis de inteligencia se adecúa pobremente a sistemas de ontología fijos convencionales.

30

El documento US2006278163 divulga un método y sistema para definir el esquema de una base de datos.

35

El documento "Combining a flexible data-model and phase schema translation in data model reverse engineering", Ingeniería Inversa 1996, Acta de la tercera conferencia de trabajo en Monterey, CA, Estados Unidos, Sociedad Informática de IEEE, 8 de noviembre 1996, páginas 141-151, XP010201977 divulga analizadores para transformar documentos XML un modelo propietario de base de datos Netmark de Oracle.

Sumario

40

En una realización, un método comprende crear y almacenar una ontología para un almacenamiento de datos en respuesta a recibir una primera entrada de usuario que define la ontología, en el que la ontología comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto; crear una o más definiciones de analizador en respuesta a recibir una segunda entrada de usuario que define las definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar los primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto; almacenar cada una de la una o más definiciones de analizador en asociación con uno de la pluralidad de tipos de propiedad de objeto; en el que la operación ejecutada por máquina es al menos una de (a) enviar dichas instrucciones a través de medios de transmisión, (b) recibir dichas instrucciones a través de medios de transmisión, (c) almacenar dichas instrucciones en un medio de almacenamiento legible por máquina y (d) ejecutar las instrucciones.

50

En una característica, el método comprende adicionalmente recibir los primeros datos de entrada; determinar si los primeros datos de entrada coinciden con una de las subdefiniciones de analizador; usar una coincidente de las subdefiniciones de analizador, creando y almacenando los datos de entrada modificados; almacenar los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.

55

En otra característica, crear y almacenar una o más definiciones de analizador comprende crear y almacenar uno o más módulos de código de programa, en la que cada uno de los módulos de código comprende código de programa informático que cuando se ejecuta provoca la transformación de los primeros datos de entrada en los datos de entrada modificados.

60

En otra característica, crear y almacenar una o más definiciones de analizador comprende crear y almacenar una o más expresiones de transformación, en la que cada una de las expresiones de transformación comprende uno o más patrones sintácticos y un identificador de tipo de propiedad asociado con cada uno de los patrones sintácticos.

65

5 En aún otra característica, crear y almacenar una o más definiciones de analizador comprende crear y almacenar una o más expresiones de transformación, en la que cada una de las expresiones de transformación comprende uno o más patrones sintácticos y un identificador de tipo de propiedad asociado con cada uno de los patrones sintácticos, y el método comprende adicionalmente recibir los primeros datos de entrada; determinar si los primeros datos de entrada coinciden con uno de los patrones sintácticos; usar uno coincidente de los patrones sintácticos, creando y almacenando datos de entrada modificados; almacenar los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica mediante el identificador de tipo de propiedad asociado con el coincidente de los patrones sintácticos.

10 En aún otra característica, crear una o más definiciones de analizador comprende crear una o más definiciones de analizador que comprenden una restricción sobre qué datos de entrada modificados son aceptables para la creación de una propiedad de uno de los tipos de propiedad de objeto. En una característica adicional, crear una o más definiciones de analizador comprende crear una o más definiciones de analizador que comprenden un valor por defecto a sustituir por un componente de los datos de entrada modificados.

15 En otra característica, el método comprende adicionalmente recibir los primeros datos de entrada; determinar si los primeros datos de entrada coinciden con diferentes sucesivas de las subdefiniciones de analizador hasta que se identifica una subdefinición de analizador coincidente; usar una coincidente de las subdefiniciones de analizador, creando y almacenando los datos de entrada modificados; almacenar los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.

20 De acuerdo con otra realización, un sistema de almacenamiento de datos comprende un almacenamiento de datos; una ontología acoplada al almacenamiento de datos y que comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto; un analizador acoplado a la ontología y configurado para recibir datos de entrada y transformar los datos de entrada en datos modificados a almacenar en una propiedad de uno de los tipos de propiedad de acuerdo con una o más definiciones de analizador; en el que cada uno de los tipos de propiedad de objeto comprende una o más de las definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar los primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto.

25 De acuerdo con otra realización, un aparato comprende medios para crear y almacenar una ontología para un almacenamiento de datos en respuesta a recibir una primera entrada de usuario que define la ontología, en el que la ontología comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto; medios para crear una o más definiciones de analizador en respuesta a recibir una segunda entrada de usuario que define las definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar los primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto; y medios para almacenar cada una de la una o más definiciones de analizador en asociación con uno de la pluralidad de tipos de propiedad de objeto.

30 En otro ejemplo una interfaz gráfica de usuario comprende un campo de patrón de expresiones configurado para aceptar una entrada de usuario que especifica un patrón de expresiones de transformación que especifica cómo transformar los primeros datos de entrada en datos de entrada modificados; una o más subdefiniciones de analizador comprendiendo cada una: una porción del patrón de expresiones de transformación; un cuadro combinado configurado para aceptar una entrada de usuario que especifica uno de una pluralidad de tipos de componentes de propiedad de objeto de una ontología de un almacenamiento de datos; en el que cada una de las subdefiniciones de analizador especifica cómo transformar una porción de los primeros datos de entrada en una porción de entrada modificada que puede almacenarse el en componente especificado de uno de la pluralidad de tipos de propiedad de objeto.

35 En una característica, la una o más subdefiniciones de analizador comprenden una restricción sobre cómo transformar la porción de los primeros datos de entrada en la porción de datos de entrada modificados que es compatible con uno de los tipos de propiedad de objeto. En aún otra característica, la una o más subdefiniciones de analizador comprenden un valor por defecto a sustituir por los datos de entrada modificados si están vacíos.

55 Breve descripción de los dibujos

La presente invención se ilustra a modo de ejemplo, y no a modo de limitación, en las figuras de los dibujos adjuntos y en los que números de referencia similares se refieren a elementos similares y en los que:

60 la Figura 1 ilustra un sistema para crear datos en un almacenamiento de datos usando una ontología dinámica; la Figura 2 ilustra la definición de una ontología dinámica para uso en la creación de datos en un almacenamiento de datos; la Figura 3 ilustra un método de transformación de datos y creación de los datos en un almacenamiento de datos usando una ontología dinámica;

65 la Figura 4 ilustra un editor de tipo de objeto de ejemplo; la Figura 5A ilustra un editor de analizador de ejemplo;

la Figura 5B ilustra un asistente de edición de propiedad de ejemplo en el que se han creado múltiples analizadores para una propiedad particular; y la Figura 6 ilustra un sistema informático con el que puede implementarse una realización.

5 Descripción de las realizaciones de ejemplo

En la siguiente descripción, para los propósitos de explicación, se exponen numerosos detalles específicos para proporcionar un completo entendimiento de la presente invención. Será evidente, sin embargo, que la presente invención puede practicarse sin estos detalles específicos. En otros casos, estructuras bien conocidas y dispositivos se muestran en forma de diagrama de bloques para evitar la obstaculización innecesaria la presente invención. En una realización, un usuario de un sistema de base de datos especifica una ontología de la base de datos en términos de tipos de objeto y tipos de propiedad para propiedades de los objetos. El usuario adicionalmente especifica cómo analizar datos de entrada para la base de datos y cómo correlacionar los datos analizados con elementos de base de datos tal como objetos o propiedades de objeto. La base de datos se elige como una realización de ejemplo, también podrían considerarse otras realizaciones tal como archivos planos o índices de búsqueda.

I. SISTEMA DE BASE DE DATOS DE ONTOLOGÍA DINÁMICA

20 La Figura 1 ilustra un sistema para crear datos en una base de datos usando una ontología dinámica. Un analizador 102 se acopla a una ontología 106, que se acopla a una base de datos 108. En una realización, la ontología 106 comprende información almacenada que proporciona un modelo de datos de datos almacenados en la base de datos 108, y la ontología se define mediante uno o más tipos de objeto 110 y uno o más tipos de propiedad 116. Uno o más objetos 112 en la base de datos 108 pueden instanciarse basándose en los tipos de objeto 110, y cada uno de los objetos tiene una o más propiedades 114A, 114B que se instancian basándose en los tipos de propiedad 116. Cada uno de los tipos de propiedad 116 puede comprender uno o más componentes 118, tal como una cadena, número, etc. Los tipos de propiedad 116 puede instanciarse basándose en un tipo base 120. Por ejemplo, un tipo base 120 puede ser "Ubicaciones" y un tipo de propiedad 116 puede ser "Casa."

30 En un ejemplo un usuario del sistema usa un editor de tipo objeto 124 para crear los tipos de objeto 110 y definir atributos de los tipos de objeto. En una realización, un usuario del sistema usa un editor de tipo de propiedad 126 para crear los tipos de propiedad 116 y definir atributos de los tipos de propiedad.

En un ejemplo crear un tipo de propiedad 116 usando el editor de tipo de propiedad 126 implica definir al menos una definición de analizador usando un editor de analizador 122. Una definición de analizador comprende metadatos que informan al analizador 102 cómo analizar los datos de entrada 100 para determinar si valores en los datos de entrada pueden asignarse al tipo de propiedad 116 que se asocia con la definición de analizador. En un ejemplo cada definición de analizador puede comprender un analizador de expresiones regulares 104A o un analizador de módulo de código 104B. En otros ejemplos, pueden proporcionarse otras clases de definiciones de analizador usando rutinas u otros elementos programáticos. Los elementos de un analizador de expresiones regulares 104A y un analizador de módulo de código 104B se describen adicionalmente en secciones posteriores. Una vez definidos, tanto el analizador de expresiones regulares 104A como el analizador de módulo de código 104B pueden proporcionar una entrada al analizador 102 para controlar el análisis de datos de entrada 100.

45 En una realización de uso del sistema de la Figura 1, se proporcionan datos de entrada 100 al analizador 102. Una propiedad de objeto que correlaciona los datos de entrada 100 habilita que el analizador determine qué tipo de objeto 110 debería recibir datos desde una fila de los datos de entrada y qué tipos de propiedad 116 deberían recibir datos desde valores de campo individuales en los datos de entrada. Basándose en la correlación de propiedad de objeto 101, el analizador 102 selecciona una de las definiciones de analizador que se asocia con un tipo de propiedad en los datos de entrada. El analizador analiza un campo de datos de entrada usando la definición de analizador seleccionada, resultando en la creación de los datos modificados 103. Los datos modificados 103 se añaden a la base de datos 108 de acuerdo con la ontología 106 almacenando valores de los datos modificados en una propiedad del tipo de propiedad especificado. Como resultado, los datos de entrada 100 que tienen formato o sintaxis variable pueden crearse en la base de datos 108. La ontología 106 puede modificarse en cualquier momento usando el editor de tipo de objeto 124 y editor de tipo de propiedad 126. El editor de analizador 122 habilita la creación de múltiples definiciones de analizador que pueden analizar satisfactoriamente los datos de entrada 100 que tienen formato o sintaxis variable y determinar qué tipos de propiedad deberían usarse para transformar los datos de entrada 100 en datos de entrada modificados 103.

60 II. DEFINIR UNA ONTOLOGÍA DINÁMICA

La Figura 2 ilustra la definición de una ontología dinámica para uso en la creación de datos en una base de datos. Para propósitos de ilustración de un ejemplo claro, se describen las etapas 202-209 de la Figura 2 primero en un nivel alto y detalles de una implementación de ejemplo siguen a la descripción de nivel alto.

65

En la etapa 202, se crean uno o más tipos de objeto para una ontología de base de datos. En la etapa 206, se crean uno o más tipos de propiedad para cada tipo de objeto. Como se indica en la etapa 204, los atributos de tipos de objeto o tipos de propiedad de la ontología pueden editarse o modificarse en cualquier momento.

5 En la etapa 208, se crea al menos una definición de analizador para cada tipo de propiedad. En la etapa 209, los atributos de una definición de analizador pueden editarse o modificarse en cualquier momento. En una realización, cada tipo de propiedad se declara que es representativa de uno o más tipos de objeto. Un tipo de propiedad es representativo de un tipo de objeto cuando el tipo de propiedad se asocia intuitivamente con el tipo de objeto. Por ejemplo, un tipo de propiedad de "Número de Seguridad Social" puede ser representativo de un tipo de objeto "Persona" pero no representativo de un tipo de objeto "Negocio."

10 En una realización, cada tipo de propiedad tiene uno o más componentes y un tipo base. En una realización, un tipo de propiedad puede comprender una cadena, una fecha, un número o un tipo compuesto consistente de dos o más elementos de cadena, fecha o número. Por lo tanto, los tipos de propiedad son extensibles y pueden representar estructuras de datos complejas. Además, una definición de analizador puede referenciar un componente de un tipo de propiedad complejo como una unidad o testigo.

15 Un ejemplo de una propiedad que tiene múltiples componentes es una propiedad de Nombre (*Name*) que tiene un componente de Apellido (*Last Name*) y un componente de Nombre de pila (*First Name*). Un ejemplo de datos de entrada brutos es "Smith, Jane". Una definición de analizador de ejemplo especifica una asociación de datos de entrada a componentes de propiedad de objeto como se indica a continuación: {LAST_NAME}, {FIRST_NAME}→Name:Last, Name:First. En una realización, la asociación {LAST_NAME}, {FIRST_NAME} se define en una definición de analizador usando simbología de expresiones regulares. La asociación {LAST_NAME}, {FIRST_NAME} indica que una cadena de apellido seguida por una cadena de nombre de pila comprende datos de entrada válidos para una propiedad de tipo Nombre. En contraste, los datos de entrada de "Smith Jane" no serían válidos para la definición de analizador especificada, pero un usuario podría crear una segunda definición de analizador que sí coincide con los datos de entrada de "Smith Jane". La definición Name:Last, Name:First especifica que valores de datos de entrada coincidentes se correlacionan con componentes llamados "Last" y "First" de la propiedad Nombre.

20 Como resultado, el análisis de los datos de entrada usando la definición de analizador resulta en la asignación del valor "Smith" al componente Name:Last de la propiedad Nombre, y el valor "Jane" al componente Name:First de la propiedad Nombre.

25 En un ejemplo usuarios administrativos usan un editor administrativo para crear o editar tipos de objeto y tipos de propiedad. En un ejemplo usuarios usan el editor administrativo para especificar definiciones de analizador y para asociar expresiones regulares, módulos de código o rutinas con las definiciones de analizador. En el editor administrativo, un usuario puede especificar atributos y componentes de un tipo de propiedad. Por ejemplo, en un ejemplo un usuario especifica un icono de interfaz gráfica de usuario que se asocia con el tipo de propiedad y visualiza en una interfaz de usuario para seleccionar el tipo de propiedad. El usuario adicionalmente especifica una definición de analizador que se asocia con el tipo de propiedad y que puede analizar datos de entrada y correlacionar los datos de entrada con propiedades que corresponden al tipo de propiedad. El usuario adicionalmente especifica un formato de visualización para el tipo de propiedad que indica cómo verán los usuarios las propiedades de ese tipo de propiedad.

30 La Figura 4 ilustra un editor de tipo de objeto de ejemplo. En un ejemplo un panel de editor de tipo objeto 402 comprende botones gráficos 404 para seleccionar las funciones de añadir, borrar y editar, y una o más filas 406 que identifican tipos de objeto y un resumen de atributos seleccionados de los tipos de objeto. Atributos seleccionados de ejemplo que pueden visualizarse en el panel de editor de objeto 402 incluyen un nombre de tipo de objeto 408 (por ejemplo, "Negocio"), un identificador de recurso uniforme (URI) 410 que especifica una ubicación de información que define el tipo de objeto (por ejemplo, "com.palantir.object.business") y un tipo base 412 del tipo de objeto, también expresado en formato de URI (por ejemplo, "com.palantir.object.entity"). Cada URI también puede incluir un icono gráfico 414.

35 En un ejemplo un usuario interactúa con un ordenador para realizar las siguientes etapas para definir un tipo de objeto. Supóngase para propósitos de un ejemplo que el nuevo tipo de objeto es Vehículo. Usando el editor de tipo objeto, el usuario selecciona el botón "Añadir Tipo de Objeto" 404 y el ordenador genera y visualiza un panel que impulsa al usuario a introducir valores para un nuevo tipo de objeto. El usuario selecciona un tipo de objeto base de Entidad, que puede comprender cualquier persona, lugar o cosa. El usuario asigna un icono gráfico al tipo de objeto Vehículo. El usuario asigna un nombre de visualización de "Vehículo" al tipo de objeto.

40 En un ejemplo un usuario interactúa con el ordenador para definir un tipo de propiedad de manera similar. El usuario especifica un nombre para el tipo de propiedad, un nombre de visualización y un icono. El usuario puede especificar uno o más validadores para un tipo de propiedad. Cada validador puede comprender una expresión regular con la que deben coincidir los datos de entrada modificados mediante un analizador para constituir datos válidos para ese tipo de propiedad. En un ejemplo cada validador se aplica a datos de entrada antes de que un proceso pueda

almacenar los datos de entrada modificados en una propiedad de objeto del tipo de propiedad asociado. Validadores se aplican después del análisis y antes de que se permita que los datos de entrada se almacenen en una propiedad de objeto.

5 En diversos ejemplos, los validadores pueden comprender expresiones regulares, un conjunto de valores fijos o un módulo de código. Por ejemplo, un tipo de propiedad que es un número puede tener un validador que comprende una expresión regular que coincide con dígitos 0 a 9. Como otro ejemplo, un tipo de propiedad que es un estado de Estados Unidos puede tener un validador que comprende el conjunto {AK, AL, CA ... VA} de abreviaturas de dos letras válidas para estados. Los conjuntos de validadores pueden ser extensibles para permitir que un usuario añada
10 valores adicionales. Un tipo de propiedad puede tener elementos de componente y cada elemento de componente puede tener un validador diferente. Por ejemplo, un tipo de propiedad de "Dirección" puede comprender como componentes "Ciudad", "Estado" y "CP", cada uno de los cuales puede tener un validador diferente.

15 En un ejemplo que define un tipo de propiedad incluye identificar una o más palabras asociadas para el tipo de propiedad. Las palabras asociadas soportan funciones de búsqueda en grandes sistemas de base de datos. Por ejemplo, un tipo de propiedad de "Dirección" puede tener una palabra asociada de "casa" de modo que una búsqueda en el sistema para propiedades "casa" ofrecerá "Dirección" como un resultado.

20 En un ejemplo que define un tipo de propiedad incluye identificar un formateador de visualización para el tipo de propiedad. Un formateador de visualización especifica cómo imprimir o visualizar un valor de un tipo de propiedad.

25 En un ejemplo cada una de las definiciones de analizador incluye una expresión regular que coincide con una entrada válida, y el analizador usa un módulo de procesamiento de expresión regular. Por ejemplo, procesadores de lenguaje Java convencionales habitualmente tienen módulos de procesamiento de expresión regular incorporados. En un ejemplo definiciones de analizador que comprenden expresiones regulares pueden encadenarse juntas. En otro ejemplo una o más de las definiciones de analizador incluyen cada una un módulo de código que contiene lógica para analizar datos de entrada y determinar si los datos de entrada coinciden con una sintaxis específica o modelo de datos. El módulo de código puede escribirse en Java, JavaScript o cualquier otro lenguaje fuente
30 adecuado.

35 En un ejemplo puede existir cualquier número de definiciones de analizador y subdefiniciones. El número de definiciones de analizador no es importante porque los datos de entrada se aplican sucesivamente a cada definición de analizador hasta que se produce una coincidencia. Cuando se produce una coincidencia, los datos de entrada se correlacionan usando las subdefiniciones de analizador con uno o más componentes de una instancia de una propiedad de objeto. Como resultado, datos de entrada pueden variar sintácticamente de una sintaxis deseada pero los valores de datos correctos se correlacionan con valores de propiedad de objeto correctos en una base de datos.

40 Por consiguiente, haciendo referencia de nuevo a la Figura 2, crear una definición de analizador para un tipo de propiedad en la etapa 208 puede comprender seleccionar un tipo de analizador tal como una expresión regular, módulo de código u otro tipo de analizador. Cuando el tipo de analizador es "módulo de código," a continuación un usuario especifica el nombre de un módulo de código particular, rutina u otro elemento funcional que puede realizar análisis para el tipo de propiedad asociado.

45 En un ejemplo que define un tipo de propiedad incluye crear una definición de un analizador para el tipo de propiedad usando un editor de analizador. La Figura 5A ilustra un visualizador en pantalla de interfaz de usuario de editor de analizador de ejemplo. En un ejemplo el visualizador en pantalla 502 comprende un cuadro combinado de tipo de analizador 504 que puede recibir una selección de usuario de un tipo de analizador, tal como "Expresión regular" o "Módulo de código." El visualizador en pantalla 502 comprende además un cuadro de entrada de texto de Nombre 506 que puede recibir un nombre especificado por usuario para la definición de analizador.
50

55 Cuando el tipo de analizador es "expresión regular," se realizan etapas las 214-220. En la etapa 214, se especifica texto de expresión regular. Por ejemplo, cuando el valor de tipo de analizador de cuadro combinado 504 es "Expresión regular," el visualizador en pantalla 502 comprende un cuadro de texto de patrón de expresiones 508 que puede recibir una entrada de usuario de texto de patrón de expresión regular.

60 En la etapa 216, se especifica un componente de tipo de propiedad y un texto de subdefinición de expresión regular coincidente. Por ejemplo, el visualizador en pantalla 502 comprende además una o más coincidencias de componente de tipo de propiedad 510. Cada coincidencia de componente de tipo de propiedad asocia una subdefinición del texto de patrón de expresión regular con el componente de tipo de propiedad que se muestra en un cuadro combinado 512. Un usuario especifica un componente de tipo de propiedad seleccionando un componente de tipo de propiedad usando el cuadro combinado 512 para una subdefinición asociada 513. Como se muestra en etapa 218, especificar un componente de tipo de propiedad y texto de subdefinición de expresión regular puede repetirse para todos los otros componentes de tipo de propiedad de un tipo de propiedad particular. Como se ve en el ejemplo de la Figura 5A, se han creado seis (6) coincidencias de componente de tipo de propiedad 510 para
65 diferentes tipos de propiedad (DIRECCIÓN1, DIRECCIÓN2, DIRECCIÓN3, CIUDAD, ESTADO, CP).

En la etapa 220, un usuario puede especificar una o más restricciones, valores por defecto y/o otros atributos de una definición de analizador. En el ejemplo de la Figura 5A, el usuario también puede especificar que no se requiere una coincidencia con un componente de tipo de propiedad particular comprobando un cuadro de comprobación de "No requerido" 514. El visualizador en pantalla 502 puede comprender además un cuadro de texto de valor por defecto 514 que puede recibir una entrada de usuario para un valor por defecto para el componente de tipo de propiedad. Si se especifica un valor por defecto, entonces el tipo de propiedad asociado recibe ese valor si no se produce una coincidencia para agrupamiento asociado de la expresión regular. En realizaciones alternativas, pueden especificarse otras restricciones.

En la etapa 222, la definición de analizador se almacena en asociación con un tipo de propiedad. Por ejemplo, seleccionar el botón GUARDAR 520 de la Figura 5A provoca el almacenamiento de una definición de analizador basándose en los valores introducidos en el visualizador en pantalla 502. Definiciones de analizador pueden almacenarse en la base de datos 108.

Para propósitos de ilustración de un ejemplo claro, se ha descrito el enfoque de la Figura 2 con referencia a la Figura 5A. Sin embargo, el enfoque de la Figura 2 puede implementarse usando otros mecanismos para crear y especificar los valores y elementos identificados en la Figura 2, y la GUI particular de la Figura 5A no se requiere.

襪

III. CREAR DATOS EN UNA BASE DE DATOS USANDO UNA ONTOLOGÍA DINÁMICA

La Figura 3 ilustra un método de transformación de datos y creación de los datos en una base de datos usando una ontología dinámica. Para propósitos de ilustración de un ejemplo claro, el enfoque de la Figura 3 se describe en este documento con referencia a la Figura 1. Sin embargo, el enfoque de la Figura 3 puede implementarse usando otros mecanismos para realizar las etapas funcionales de la Figura 3, y el sistema particular de la Figura 1 no se requiere.

En la etapa 302, se reciben datos de entrada. En una realización, se recibe un archivo de datos de entrada. El archivo de datos de entrada puede comprender un archivo de valores separados por comas (CSV), una hoja de cálculo, XML u otros formatos de archivos de datos de entrada. Los datos de entrada 100 de la Figura 1 pueden representar tales formatos de archivo o cualquier otra forma de datos de entrada.

En la etapa 304, se identifica un tipo de objeto asociado con filas de datos de entrada de los datos de entrada y se identifican uno o más tipos de propiedad asociados con campos de datos de entrada de los datos de entrada. Por ejemplo, la correlación de propiedad de objeto 101 de la Figura 1 especifica que los datos de entrada 100 comprenden filas que corresponden a tipo de objeto PERSONA y campos que corresponden a componentes de tipo de propiedad LAST_NAME, FIRST_NAME de tipo de propiedad NOMBRE. La correlación de propiedad de objeto 101 puede integrarse en los datos de entrada 100 o puede almacenarse como metadatos en asociación con una herramienta de entrada de datos.

En la etapa 306, una fila de datos se lee desde los datos de entrada y se identifican uno o más valores de campo basándose en delimitadores u otros identificadores de campos en los datos de entrada.

En la etapa 308, se selecciona un conjunto de definiciones de analizador asociado con el tipo de propiedad de un campo de datos de entrada particular. Por ejemplo, metadatos almacenados como parte de creación de un tipo de propiedad específica un conjunto de definiciones de analizador, como se ha descrito anteriormente en conexión con la Figura 5A.

En la etapa 310, se aplica la siguiente definición de analizador a un valor de campo de datos de entrada. Por lo tanto, datos campos se leen desde cada fila del fichero y emparejan con cada analizador que se ha definido para los correspondientes tipos de propiedad. Por ejemplo, supóngase que la coincidencia indica que un archivo CSV de datos de entrada comprende valores (Apellido, Nombre de Pila) para propiedades de Nombre de objetos de Persona. Datos campos se leen desde el archivo CSV de datos de entrada y comparan con cada uno de los analizadores que se han definido para el tipo de propiedad Nombre dado el campo Nombre de Pila y el campo Apellido. Si se produce una coincidencia para un valor de par (Apellido, Nombre de Pila) con cualquiera de los analizadores para el tipo de propiedad Nombre, entonces el analizador transforma el par de datos de entrada de (,Apellido, Nombre de Pila) en datos de entrada modificados a almacenar en una instanciación de una propiedad de Nombre.

Si la aplicación de una definición en la etapa 310 resulta en una coincidencia con los datos de entrada, como se prueba en la etapa 312, entonces en la etapa 318 se crea una instancia de propiedad y el valor de campo de datos de entrada se almacena en una propiedad del tipo de propiedad asociado con la subdefinición coincidente de la definición de analizador. Por ejemplo, haciendo referencia a la Figura 5A, supóngase que los datos de entrada coinciden con la expresión regular 508 para un valor DIRECCIÓN. La coincidencia 510 especifica cómo almacenar los datos que emparejan cada agrupación de la expresión regular con un componente de la propiedad DIRECCIÓN. En respuesta, se crea una instancia de una propiedad DIRECCIÓN en la memoria informática y el valor de datos de entrada modificado coincidente se almacena en cada componente de la instancia de propiedad.

Si no se produce ninguna coincidencia en la etapa 312, entonces el control se transfiere a la etapa 314 para probar si otras definiciones de analizador coinciden con el mismo valor de datos de entrada. La Figura 5B ilustra un asistente de edición de propiedad de ejemplo en el que se han creado múltiples analizadores para una propiedad particular y, a través del bucle mostrado en la Figura 3, cada uno de los múltiples analizadores puede usarse en el emparejamiento de datos de entrada. Si no se produce ninguna coincidencia con la definición de analizador dada, entonces cualquier otra definición de analizador para ese tipo de propiedad se emparejan hasta que o bien no se produce ninguna coincidencia, o bien no hay disponibles otras definiciones.

Si un agrupamiento está vacío, entonces el componente se rellena mediante el valor por defecto para ese componente, si existe. Si no hay disponibles otras definiciones de analizador, entonces el control se transfiere desde la etapa 314 a la etapa 316, en la que se eleva apuntar un error o la propiedad se descarta en la etapa 320, las etapas anteriores se repiten para todos los otros valores y filas en los datos de entrada hasta que el proceso ha transformado todos los datos de entrada en propiedades en memoria.

En la etapa 322, se instancia un objeto del tipo de objeto correcto. Por ejemplo, la correlación de propiedad de objeto 101 puede especificar un tipo de objeto para datos de entrada particulares y ese tipo de objeto se instancia. El objeto recientemente creado se asocia en memoria con las propiedades que ya están en memoria. El objeto resultante se almacena en la base de datos en la etapa 324.

Etapas en el proceso anterior pueden organizarse en un conducto. Usando los enfoques en este documento, un usuario puede autodefinir una ontología de base de datos y usar técnicas automatizadas y basadas en máquina para transformar datos de entrada de acuerdo con analizadores definidos por usuario y almacenar los datos transformados en la base de datos de acuerdo con la ontología. El enfoque proporciona movimiento eficiente de datos en una base de datos de acuerdo con una ontología. Los datos de entrada tienen inteligibilidad mejorada después de la transformación porque los datos se almacenan en una ontología canónica. Además, el enfoque es flexible y adaptable, porque el usuario puede modificar la ontología en cualquier momento y no está atado a una ontología fija. El usuario también puede definir múltiples analizadores para resultar en coincidencias semánticas con datos de entrada incluso cuando la sintaxis de los datos de entrada es variable.

IV. HARDWARE DE IMPLEMENTACIÓN DE EJEMPLO

La Figura 6 es un diagrama de bloques que ilustra un sistema informático 600 en el que puede implementarse una realización de la invención. El sistema informático 600 incluye un bus 602 u otro mecanismo de comunicación para comunicar información y un procesador 604 acoplado con el bus 602 para procesar información. El sistema informático 600 también incluye una memoria principal 606, tal como una Memoria de Acceso Aleatorio (RAM) u otro dispositivo de almacenamiento dinámico, acoplado al bus 602 para almacenar información e instrucciones a ejecutar por el procesador 604. La memoria principal 606 también puede usarse para almacenar variables temporales u otra información intermedia durante la ejecución de instrucciones a ejecutar por el procesador 604. El sistema informático 600 incluye adicionalmente una memoria de solo lectura (ROM) 608 o otro dispositivo de almacenamiento estático acoplado al bus 602 para almacenar información estática e instrucciones para el procesador 604. Se proporciona un dispositivo de almacenamiento 610, tal como un disco magnético o disco óptico, y acopla al bus 602 para almacenar información y instrucciones.

El sistema informático 600 puede acoplarse a través del bus 602 a un visualizador 612, tal como un tubo de rayos catódicos (CRT), para visualizar información a un usuario de ordenador. Un dispositivo de entrada 614, incluyendo teclas alfanuméricas y otras, se acopla al bus 602 para comunicar información y ordenar selecciones al procesador 604. Otro tipo de dispositivo de entrada de usuario es el cursor control 616, tal como un ratón, una bola de mando o teclas de dirección de cursor para comunicar información de dirección y ordenar selecciones al procesador 604 y para controlar el movimiento de cursor en el visualizador 612. Este dispositivo de entrada habitualmente tiene dos grados de libertad en dos ejes, un primer eje (por ejemplo, x) y en segundo eje (por ejemplo, y), que permite que el dispositivo especifique posiciones en un plano.

La invención se refiere al uso del sistema informático 600 para implementar las técnicas descritas en este documento. De acuerdo con una realización de la invención, esas técnicas se realizan mediante el sistema informático 600 en respuesta un procesador 604 que ejecuta una o más secuencias de una o más instrucciones contenidas en la memoria principal 606. Tales instrucciones pueden leerse en la memoria principal 606 desde otro medio legible por máquina, tal como el dispositivo de almacenamiento 610. La ejecución de las secuencias de instrucciones contenidas en la memoria principal 606 provocan que el procesador 604 realice las etapas de proceso descritas en este documento. En realizaciones alternativas, puede usarse circuitería por cable en lugar de o en combinación con instrucciones de software para implementar la invención. Por lo tanto, realizaciones de la invención no se limitan a ninguna combinación específica de circuitería de hardware y software.

La expresión "medio legible por máquina" como se usa en este documento se refiere a cualquier medio que participa en la provisión de datos que provocan que una máquina opere de una forma específica. En una realización implementada usando el sistema informático 600, diversos medios legibles por máquina se refieren, por ejemplo, a la provisión de instrucciones al procesador 604 para ejecución. Un medio de este tipo puede tomar muchas formas,

incluyendo pero sin limitación a, medio no volátil, medio volátil y medio de transmisión. Medio no volátil incluye, por ejemplo, discos ópticos o magnéticos, tal como el dispositivo de almacenamiento 610. Medio volátil incluye memoria dinámica, tal como la memoria principal 606. Medio de transmisión incluye cables coaxiales, alambre de cobre y fibra óptica, incluyendo los alambres que comprenden el bus 602. Medio de transmisión también puede tomar la forma de ondas acústicas o luminosas, tal como las generadas durante comunicaciones de datos por ondas de radio o infrarrojos. Todos los medios de este tipo deben ser tangibles para habilitar que las instrucciones transportadas por el medio se detecten mediante un mecanismo físico que lee las instrucciones en una máquina.

Formas comunes de medios legibles por máquina incluyen, por ejemplo, un disquete, un disco flexible, disco duro, cinta magnética, o cualquier otro medio magnético, un CD-ROM, cualquier otro medio óptico, tarjetas perforadas, cinta de papel, cualquier otro medio físico con patrones a agujeros, una RAM, una PROM y EPROM, una FLASH-EPROM, cualquier otro chip o cartucho de memoria, una onda portadora como se describe en lo sucesivo, o cualquier otro medio a partir del cual puede leer un ordenador.

Diversas formas de medios legibles por máquina pueden implicarse en el transporte de una o más secuencias de una o más instrucciones al procesador 604 para ejecución. Por ejemplo, las instrucciones pueden transportarse inicialmente en un disco magnético de un ordenador remoto. El ordenador remoto puede cargar las instrucciones en su memoria dinámica y enviar las instrucciones a través de una línea de teléfono usando un módem. Un módem local al sistema informático 600 puede recibir los datos en la línea de teléfono y usar un transmisor de infrarrojos para convertir los datos a una señal de infrarrojos. Un detector de infrarrojos puede recibir los datos transportados en la señal de infrarrojos y circuitería apropiada puede colocar los datos en el bus 602. El bus 602 transporta los datos a la memoria principal 606, de la que el procesador 604 recupera y ejecuta las instrucciones. Las instrucciones recibidas por la memoria principal 606 puede almacenarse opcionalmente en el dispositivo de almacenamiento 610 o bien antes o bien después de la ejecución por el procesador 604.

El sistema informático 600 también incluye una interfaz de comunicación 618 acoplada al bus 602. La interfaz de comunicación 618 proporciona una comunicación de datos bidireccional que se acopla a un enlace de red 620 que se conecta a una red local 622. Por ejemplo, la interfaz de comunicación 618 puede ser una tarjeta de red digital de servicios integrados (ISDN) o un módem para proporcionar una conexión de comunicación de datos a un correspondiente tipo de línea de teléfono. Como otro ejemplo, la interfaz de comunicación 618 puede ser una tarjeta de red de área local (LAN) para proporcionar una conexión de comunicación de datos a una LAN compatible. También pueden implementarse enlaces inalámbricos. En una implementación cualquiera de este tipo, la interfaz de comunicación 618 envía y recibe señales eléctricas, electromagnéticas u ópticas que transportan flujos de datos digitales que representan diversos tipos de información.

El enlace de red 620 habitualmente proporciona comunicación de datos a través de una o más redes a otros dispositivos de datos. Por ejemplo, el enlace de red 620 puede proporcionar una conexión a través de la red local 622 a un ordenador anfitrión 624 o a equipo de datos operado por un Proveedor de Servicio de Internet (ISP) 626. El ISP 626 a su vez proporciona servicios de comunicación de datos a través de la red de comunicación de datos de paquetes mundial ahora comúnmente denominada como la "Internet" 628. Tanto la red local 622 como Internet 628 usan señales eléctricas, electromagnéticas u ópticas que transportan flujos de datos digitales. Las señales a través de las diversas redes y las señales en el enlace de red 620 y a través de interfaz de comunicación 618, que transportan los datos digitales a y desde el sistema informático 600, son formas ilustrativas de ondas portadoras que transportan la información.

El sistema informático 600 pueden enviar mensajes y recibir datos, incluyendo código de programa, a través de la red o redes, el enlace de red 620 y la interfaz de comunicación 618. En la Internet ejemplo, un servidor 630 podría transmitir un código solicitado para un programa de aplicación a través de la Internet 628, el ISP 626, la red local 622 y la interfaz de comunicación 618.

El código recibido puede ejecutarse por el procesador 604 a medida que se recibe, y/o almacenarse en el dispositivo de almacenamiento 610 u otro almacenamiento no volátil para posterior ejecución. De esta manera, el sistema informático 600 puede obtener código de aplicación en forma de una onda portadora.

En la anterior memoria descriptiva, realizaciones de la invención se han descrito con referencia a numerosos detalles específicos que pueden variar de implementación en implementación. Por lo tanto, el solo y exclusivo indicador de qué es la invención, y se concibe por los solicitantes para que sea la invención, es el conjunto de reivindicaciones que se publican a partir de esta solicitud, en la forma específica en las que tales reivindicaciones se publican, incluyendo cualquier corrección posterior. Cualquier definición expuesta en este documento para términos contenidos en tales reivindicaciones determinarán el significado de tales términos según se usan en las reivindicaciones. Por lo tanto, ninguna limitación, elemento, propiedad, característica, ventaja o atributo que no se cite expresamente en una reivindicación debería limitar el alcance de tal reivindicación de ninguna forma. La memoria descriptiva y dibujos, por consiguiente, deben considerarse en un sentido ilustrativo en lugar de un sentido restrictivo.

REIVINDICACIONES

1. Un método para crear datos en un almacenamiento de datos usando una ontología acoplada al almacenamiento de datos, que comprende realizar una operación ejecutada por máquina que implica instrucciones, en el que dichas instrucciones son instrucciones que, cuando se ejecutan por uno o más procesadores, provocan que el uno o más procesadores realicen ciertas etapas incluyendo:
- almacenar una ontología para el almacenamiento de datos, en el que la ontología comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto;
- almacenar un analizador configurado para recibir primeros datos de entrada y transformar los datos de entrada en datos modificados de acuerdo con una o más definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto;
- almacenar cada una de la una o más definiciones de analizador en asociación con uno de la pluralidad de tipos de propiedad de objeto;
- comprendiendo el método adicionalmente, usando la ontología almacenada:
- recibir primeros datos de entrada;
- recibir una correlación de propiedades de objeto integrada en los primeros datos de entrada, especificando la correlación de propiedades de objeto los tipos de objeto de datos y los tipos de propiedad de objeto de los primeros datos de entrada;
- seleccionar basándose en la correlación de propiedades de objeto una de las subdefiniciones de analizador que se asocia con un tipo de propiedad en los primeros datos de entrada;
- usar la seleccionada de las subdefiniciones de analizador, usando la transformación expresada en la subdefinición para transformar porciones de los primeros datos de entrada en componentes de la propiedad de objeto para crear y almacenar los datos de entrada modificados;
- almacenar los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.
2. El método de la reivindicación 1, en el que crear y almacenar una o más definiciones de analizador comprende crear y almacenar uno o más módulos de código de programa, en el que cada uno de los módulos de código comprende código de programa informático que cuando se ejecuta provoca la transformación de los primeros datos de entrada en los datos de entrada modificados.
3. El método de la reivindicación 1, en el que crear y almacenar una o más definiciones de analizador comprende crear y almacenar una o más expresiones de transformación, en el que cada una de las expresiones de transformación comprende uno o más patrones sintácticos y un identificador de tipo de propiedad asociado con cada uno de los patrones sintácticos.
4. El método de la reivindicación 1, en el que crear y almacenar una o más definiciones de analizador comprende crear y almacenar una o más expresiones regulares, en el que cada una de las expresiones regulares cuando se ejecutan provoca la transformación de los primeros datos de entrada en los datos de entrada modificados.
5. Un sistema de almacenamiento de datos, que comprende:
- un almacenamiento de datos;
- una ontología acoplada al almacenamiento de datos y que comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto;
- un analizador acoplado a la ontología y configurado para recibir datos de entrada y transformar los datos de entrada en datos modificados a almacenar en una propiedad especificada por uno de los tipos de propiedad de acuerdo con una o más definiciones de analizador;
- en el que cada uno de los tipos de propiedad de objeto comprende una o más de las definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar porciones de primeros datos de entrada en datos de entrada modificados que tienen que almacenarse en componentes de uno de los tipos de propiedad de objeto,
- el sistema de almacenamiento de datos comprendiendo además lógica que cuando se ejecuta provoca:
- la recepción de primeros datos de entrada;
- la recepción de una correlación de propiedades de objeto integrada en los primeros datos de entrada, especificando la correlación de propiedades de objeto los tipos de objeto de datos y los tipos de propiedad de objeto de los primeros datos de entrada;
- basándose la selección en la correlación de propiedades de objeto de una de las subdefiniciones de analizador que se asocia con un tipo de propiedad en los primeros datos de entrada;
- el uso de la seleccionada de las subdefiniciones de analizador, usando la transformación expresada en la subdefinición para transformar porciones de los primeros datos de entrada en componentes de la propiedad de objeto para crear y almacenar los datos de entrada modificados;

el almacenamiento de los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.

5 6. El sistema de la reivindicación 5, en el que la una o más definiciones de analizador comprenden uno o más módulos de código de programa, en el que cada uno de los módulos de código comprende código de programa informático que cuando se ejecuta provoca la transformación de los primeros datos de entrada en los datos de entrada modificados.

10 7. El sistema de la reivindicación 5, en el que la una o más definiciones de analizador comprenden una o más expresiones de transformación, en el que cada una de las expresiones de transformación comprende uno o más patrones sintácticos y un identificador de tipo de propiedad asociado con cada uno de los patrones sintácticos.

15 8. El sistema de la reivindicación 6, en el que la una o más definiciones de analizador comprenden una o más expresiones de transformación, en el que cada una de las expresiones de transformación comprende uno o más patrones sintácticos y un identificador de tipo de propiedad asociado con cada uno de los patrones sintácticos.

20 9. El sistema de la reivindicación 5, en el que la una o más definiciones de analizador comprenden una restricción sobre cómo transformar primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto.

10. El sistema de la reivindicación 5, en el que la una o más definiciones de analizador comprenden un valor por defecto a sustituir por un componente faltante de los datos de entrada modificados.

25 11. El sistema de la reivindicación 5, en el que el analizador comprende además lógica que cuando se ejecuta provoca la recepción de los primeros datos de entrada; la determinación de si los primeros datos de entrada coinciden con diferentes sucesivas de las definiciones de analizador hasta que se identifica una definición de analizador coincidente; el uso de una coincidente de las definiciones de analizador, la creación y almacenamiento de los datos de entrada modificados; el almacenamiento de los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.

30 12. Un aparato, que comprende:

35 medios para almacenar una ontología para un almacenamiento de datos, en el que la ontología comprende una pluralidad de tipos de objeto de datos y una pluralidad de tipos de propiedad de objeto;

medios para crear una o más definiciones de analizador, en el que cada una de las definiciones de analizador especifica una o más subdefiniciones de cómo transformar primeros datos de entrada en datos de entrada modificados que son compatibles con uno de los tipos de propiedad de objeto; medios para almacenar cada una de la una o más definiciones de analizador en asociación con uno de la pluralidad de tipos de propiedad de objeto.

40 medios para recibir primeros datos de entrada;
medios para recibir una correlación de propiedades de objeto integrada en los primeros datos de entrada, especificando la correlación de propiedades de objeto los tipos de objeto de datos y los tipos de propiedad de objeto de los primeros datos de entrada;

45 medios para seleccionar basándose en la correlación de propiedades de objeto una de las subdefiniciones de analizador que se asocia con un tipo de propiedad en los primeros datos de entrada;

medios para usar la transformación expresada en la subdefinición para transformar usando una seleccionada de la una de las porciones de subdefiniciones de analizador de los primeros datos de entrada en componentes de la propiedad de objeto para crear y almacenar los datos de entrada modificados;

50 medios para almacenar los datos de entrada modificados en una propiedad del tipo de propiedad que se identifica en la coincidente de las subdefiniciones de analizador.

55 13. Un medio legible por máquina que almacena instrucciones, en el que dichas instrucciones son instrucciones que, cuando se ejecutan por uno o más procesadores, provocan que el uno o más procesadores realicen las etapas del método citadas en cualquiera de las reivindicaciones 1-4.

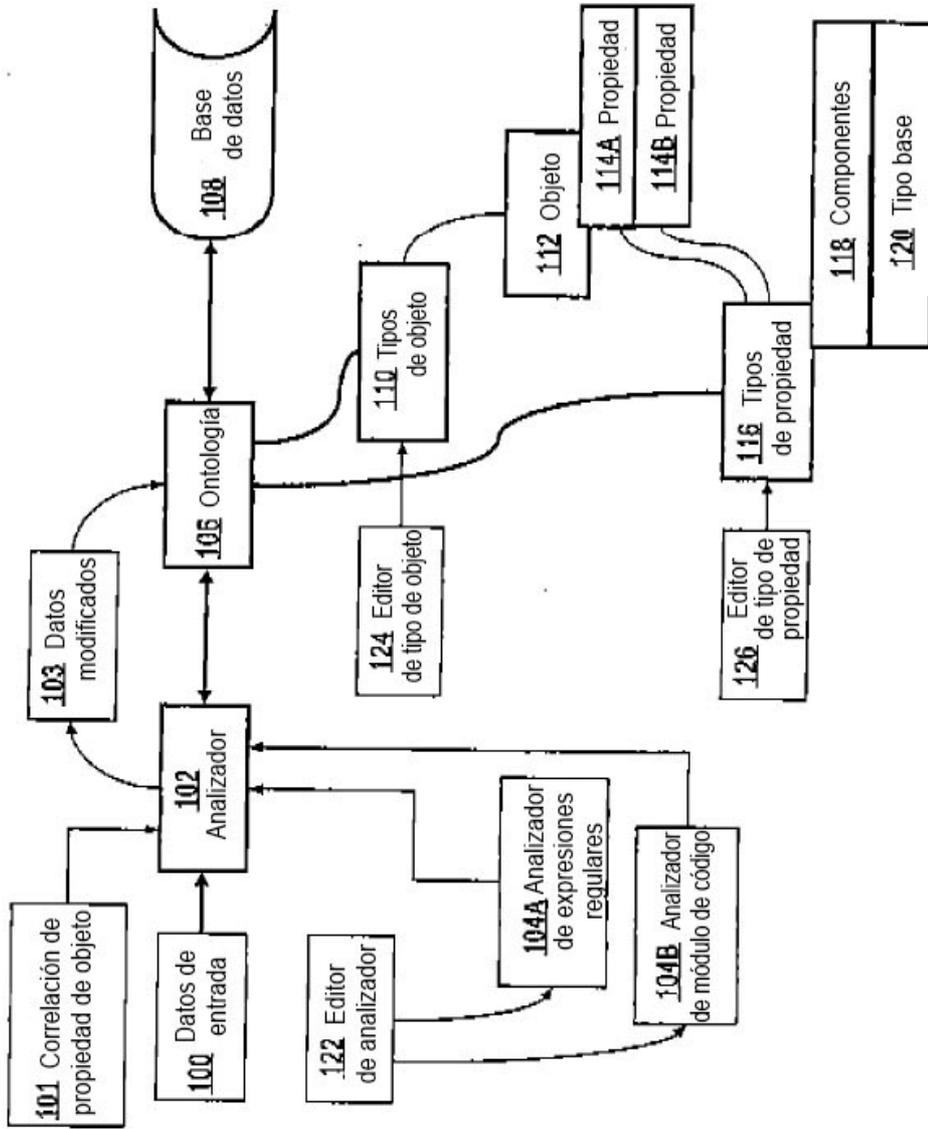


Fig. 1

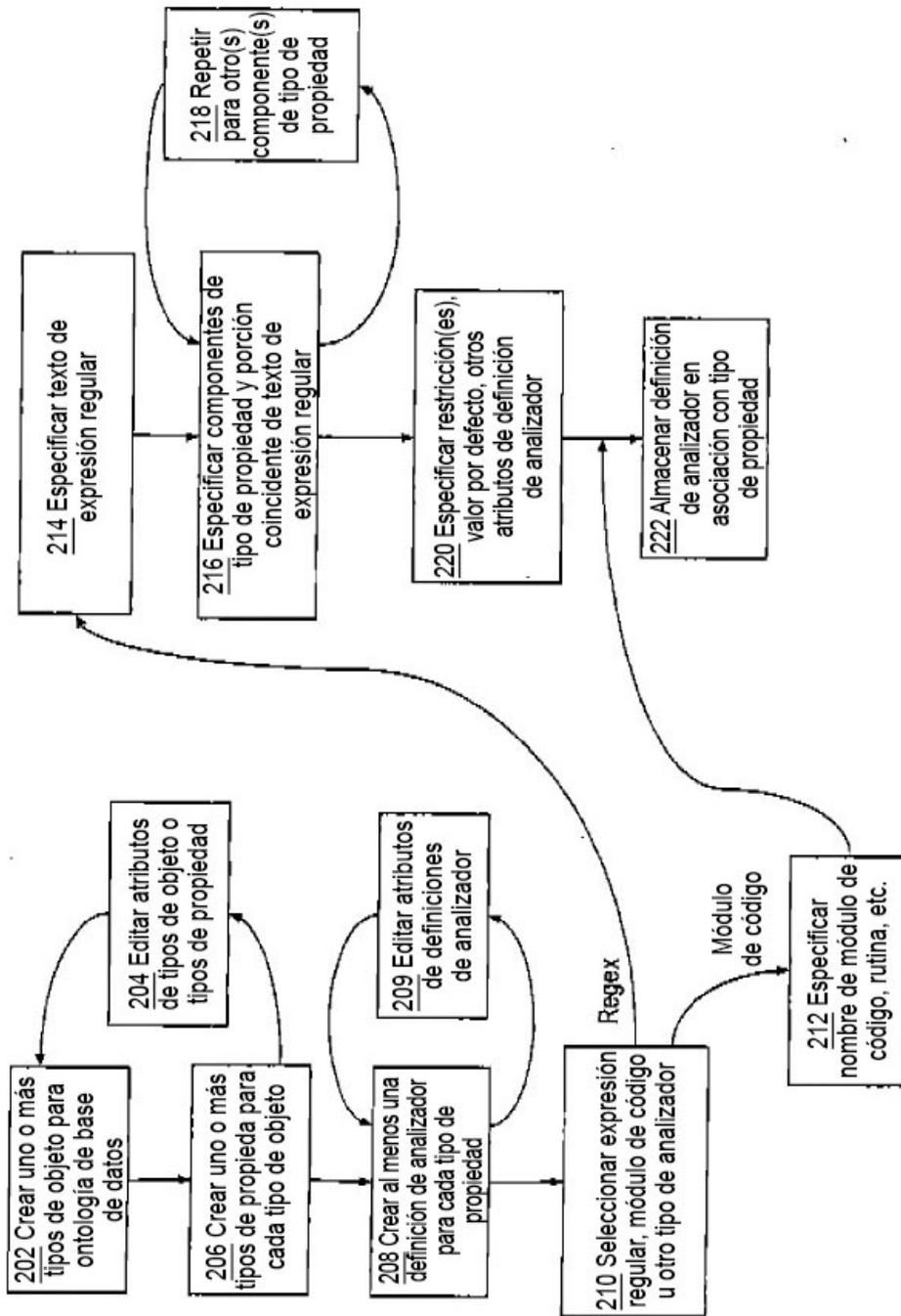


Fig. 2
Definición de una Ontología Dinámica

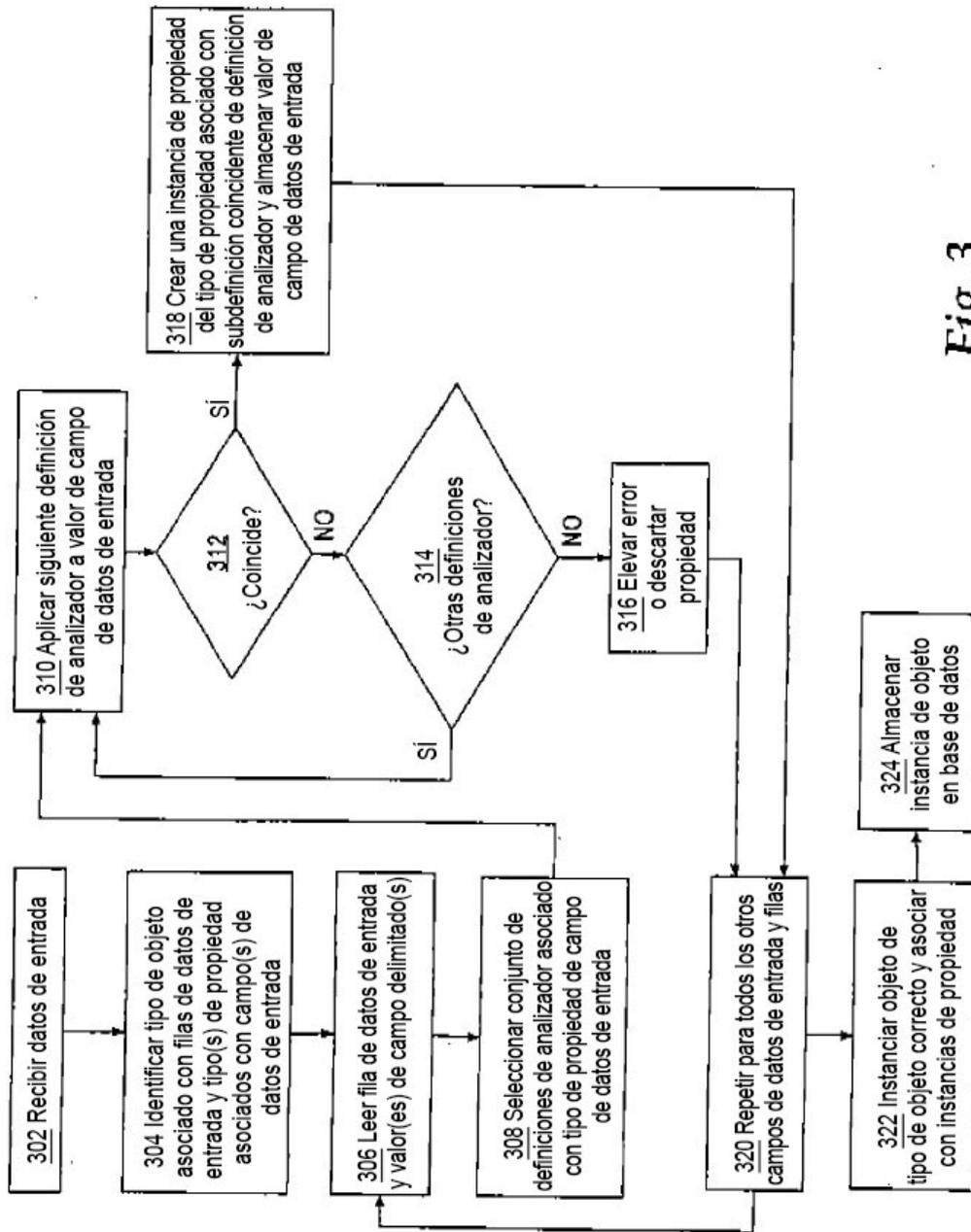


Fig. 3

Transformación de Datos Usando una Ontología Dinámica

402

404

406

Editor de tipo de objeto

Añadir tipo de objeto Borrar tipo de objeto Editar tipo de objeto

NOMBRE DE TIPO DE OBJETO	URI	Tipo Base
ACTIVO	com.palantir.object.dod.as	com.palantir.object.dod.en
NEGOCIO	com.palantir.objec.business 410	com.palantir.object.dod.en
FORMULARIO CTR	com.palantir.object.ctform	com.palantir.object.dod
AGENTE DE CASO	com.palantir.object.dod.caseagen	com.palantir.object.dod.en
DOCUMENTO	com.palantir.object.document	com.palantir.object.dod
CORREC 414	com.palantir.object.email	com.palantir.object.dod.ev
ENTIDAD 408	com.palantir.object.entity	com.palantir.object.dod.en
EVENTO	com.palantir.object.event	com.palantir.object.dod.ev
VUELO	com.palantir.object.flight	com.palantir.object.dod.ev
ATAQUE LED	com.palantir.object.dod.led	com.palantir.object.dod.ev
MEZQUITA	com.palantir.object.dod.mosque	com.palantir.object.dod.en
PAGO	com.palantir.object.payment	com.palantir.object.dod.ev
PERSONA	com.palantir.object.person	com.palantir.object.dod.en
LLAMADA TELEFÓNICA	com.palantir.object.call	com.palantir.object.dod.ev
FORMULARIO SAR	com.palantir.object.sar	com.palantir.object.dod
AVISTAMIENTO	com.palantir.object.dod.sighting	com.palantir.object.dod.ev
TERRORISTA SUICIDA	com.palantir.object.dod.suicid...	com.palantir.object.dod.en
ATENIDO SUICIDA	com.palantir.object.dod.suicid...	com.palantir.object.dod.ev
SINAGOGA	com.palantir.object.dod.synag...	com.palantir.object.dod.en
ORGANIZACIÓN TERR...	com.palantir.object.dod.terrori...	com.palantir.object.dod.en

FIG. 4

502

Editor de analizador

504 Tipo de analizador: Expresión regular Url.com.palantir.markers.regex

Nombre: Expresión regular 506

508 Patrón de expresión: `{[\\w\\s]*}.([\\w\\s]*).([\\w\\s]*).([\\w\\s]*)}` Rehacer análisis de componente

513 DIRECCIÓN3 No requerido: Valor por defecto:

512 DIRECCIÓN3 No requerido: Valor por defecto:

DIRECCIÓN3 No requerido: Valor por defecto:

CIUDAD No requerido: Valor por defecto:

ESTADO No requerido: Valor por defecto:

CP No requerido: Valor por defecto:

520 Guardar Cancelar

510

FIG. 5A

504

Asistente de edición de propiedades

Analizadores

Nombre de analizador	URI	Args
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Expresión regular	com.palantir.makers.regex	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>
Compuesto	com.palantir.makers.composite	Patrón de expresión: <code>{[wls]*}, {[wls]*}, {[wls]*}, {[wls]*}</code>

Añadir nuevo
Eliminar
Mover arriba
Mover abajo

Atrás
Siguiente

FIG. 5B

Fig. 6

