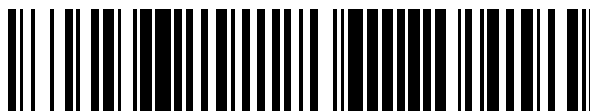


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 401 727**

51 Int. Cl.:

G06F 17/30 (2006.01)

G06F 21/22 (2006.01)

G06F 9/44 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **25.10.2007 E 07866439 (8)**

97 Fecha y número de publicación de la concesión europea: **05.12.2012 EP 2084644**

54 Título: **Herramienta informática de gestión de documentos numéricos**

30 Prioridad:

27.10.2006 FR 0609464

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

24.04.2013

73 Titular/es:

**INRIA - INSTITUT NATIONAL DE RECHERCHE EN
INFORMATIQUE ET EN AUTOMATIQUE (100.0%)
Domaine de Voluceau, Rocquencourt, Boîte
Postale 105
78153 Le Chesnay Cedex, FR**

72 Inventor/es:

**ROUSSEAU, GUILLAUME y
BIAIS, MAXIME**

74 Agente/Representante:

PONTI SALES, Adelaida

ES 2 401 727 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).

HERRAMIENTA INFORMÁTICA DE GESTIÓN DE DOCUMENTOS NUMÉRICOS

DESCRIPCIÓN

5

[0001] La invención se refiere al tratamiento de documentos informáticos.

[0002] Un documento informático es básicamente un archivo de cualquier tipo, cuyo contenido puede hacerse accesible al hombre, directamente o indirectamente. La accesibilidad puede ser bastante directa si se trata de archivos de tipo textual. Es menos directa si se trata de música o de video. Más adelante se considerarán otros ejemplos.

[0003] Los sistemas de archivos están, desde hace ya tiempo, organizados para permitir una datación precisa, y en principio exacta, de cada archivo. Asimismo, se sabe, de diferentes maneras, asociar a un archivo informático diferentes atributos. Básicamente, estos atributos están ligados a la naturaleza informática del propio archivo, por ejemplo los atributos de "solo lectura", "archivable", "sistema", o también "caché". A parte de los atributos ligados al sistema de archivo, pueden asociarse otros atributos a un archivo, por ejemplo en un encabezado de este. La informática permite también comparar documentos. Una comparación "mínima" puede hacerse utilizando solamente los atributos del sistema de archivo. Se aplica a cualquier tipo de archivo, pero es muy imperfecta: considerará por ejemplo como diferentes unos archivos idénticos, pero que tienen una fecha de sistema diferente. Las comparaciones más evolucionadas se interesan por el contenido de los archivos. Las modalidades de la comparación difieren entonces según que se trate por ejemplo de un archivo de tipo "texto", o bien, al contrario, de un archivo binario, con, entre los dos, unos archivos de naturaleza más evolucionada, tal como los archivos de imagen, cuya comparación es más delicada.

[0004] El Solicitante ha observado que no existe, actualmente, técnica alguna que permita de manera metódica y automática, realizar comparaciones, al mismo tiempo que una datación precisa de los elementos de esta comparación. La necesidad de una técnica de este tipo se nota en diferentes sectores, como el de la determinación del estatuto jurídico de obras en el sentido de los derechos de autor, y más especialmente de obras de tipo software.

[0005] Por otro lado, el documento US 6 658 423 B1 describe una técnica de detección de documentos duplicados o casi-duplicados.

[0006] La presente invención es para mejorar la situación.

[0007] La invención tiene por objeto un dispositivo informático de gestión de documentos, tal como se define en la reivindicación 1.

[0008] Unos modos de realización del dispositivo según la invención se describen en las reivindicaciones dependientes 2 a 25.

[0009] Otras características y ventajas de la invención aparecerán con la lectura de la descripción detallada a continuación y de los dibujos en los cuales:

- la figura 1 representa un esquema funcional que ilustra un ejemplo de herramienta según la invención,
- la figura 2 representa un ordinograma que ilustra una función de generación de firma de documento para la herramienta de la invención,
- la figura 3 representa una tabla que ilustra un almacenamiento en la base de trabajo de la figura 1,
- la figura 4 representa un ordinograma que ilustra una función de generación de firma de estado para la herramienta de la invención,
- la figura 5 es un ordinograma que ilustra una función de generación de huella de juego de documentos para la herramienta de la invención,
- la figura 6 representa una tabla que ilustra otra parte del almacenamiento de la base de trabajo de la figura 1,
- la figura 7 representa un ordinograma que ilustra una función de generación de firma de documento para la herramienta de la invención en un modo de realización particular,
- la figura 8 representa un ordinograma que ilustra una función de datación de huella en un modo de realización particular,
- la figura 9 representa un ordinograma que ilustra una función de generación de huella en un modo de realización particular,
- la figura 10 representa un esquema que ilustra la estructura de la base de datos de producción de la figura 1 en un modo de realización particular,
- la figura 11 representa una tabla que ilustra un almacenamiento en la base de datos de producción de la figura 1 en un modo de realización particular,
- la figura 12 representa un ordinograma que ilustra una función de comparación de huella para el módulo de comparación de la figura 1 en un primer modo de realización,
- la figura 13 representa un ordinograma que ilustra un primer ejemplo de una función de agregación para el módulo de comparación de la figura 1,
- la figura 14 representa un ordinograma que ilustra un segundo ejemplo de una función de agregación para el módulo de comparación de la figura 1, y
- la figura 15 representa un ordinograma que ilustra una función de comparación de huella para el módulo de comparación de la figura 1 en un segundo modo de realización.

65

[0010] Los dibujos adjuntos podrán no solamente servir para completar la invención, sino también para contribuir a su definición, en este caso.

[0011] Como se ha visto, la expresión "documentos informáticos" se utiliza aquí en un sentido bastante ancho. Un documento informático puede ser básicamente un archivo de cualquier tipo, aunque unos archivos de tipo puramente textual, mayoritariamente textuales, o con connotación textual sean objetivos especiales en determinados modos de realización de la invención.

[0012] Concretamente, un documento se presenta a menudo como un único archivo informático físico. Sin embargo, la expresión "documento informático" cubre también el caso en que varios documentos se encontrarían en un mismo archivo físico, o bien, inversamente, aquel en que un documento determinado está compuesto por varios archivos informáticos físicos. Se habla a menudo de "archivo lógico", para abstraerse de la relación entre el documento y el o los archivos físicos que contienen el documento o sus partes.

[0013] El contenido del documento puede ser "ejecutado": representar un texto o una página web, escuchar música, ver un video, o bien ejecutar un programa, por ejemplo. En este caso, el documento no ha sido en principio modificado.

[0014] El contenido del documento también puede hacerse accesible al hombre como entrada, es decir creación/modificación. Cualquier entrada por poco organizada que sea se traduce por la existencia de diferentes versiones de un mismo documento.

[0015] Básicamente, se denomina aquí "juego de documentos" un conjunto de versiones de un mismo documento. Sin embargo, según una óptica más extendida, el juego de documentos podrá comprender varios conjuntos de versiones de varios documentos, en el marco de un proyecto más vasto.

Definiciones y convenciones de notación

[0016] Se indica P_i un juego de documentos numéricos. Por ejemplo, el juego P_i puede ser un proyecto de software que comprende un conjunto de archivos informáticos de tipo llamado "fuentes" (es decir que comprende código informático no ejecutable directamente y destinados a ser compilados en una sucesión de instrucciones ejecutables), y documentos numéricos anexos, tales como archivos de tipo texto que comprenden una descripción, una documentación, y/o el conjunto de los documentos de preparación del software. El juego P_i también puede estar constituido por archivos organizados con ayuda de una base de datos, de archivos de tipo XML (para "eXtensible Markup Language" o lenguaje de balizamiento extensible) o también SVG (para "Scalable Vector Graphics" o gráficos vectoriales extensibles).

[0017] Un estado j del juego P_i se define como el conjunto de los documentos numéricos del juego P_i , en el estado en que se presentan en un instante determinado, por ejemplo en una fecha T_{ij} . Se entiende que un estado j_2 del juego P_i puede diferir de un estado j_1 de este juego, por ejemplo por la presencia de documentos numéricos suplementarios, por la ausencia de determinados documentos numéricos y/o también por la presencia de documentos numéricos de contenido diferente.

[0018] El estado j del juego P_i se indica S_{ij} .

[0019] En determinados casos, y en particular cuando el juego P_i comprende un proyecto de software, el estado S_{ij} puede ser visto como una versión del juego de documentos P_i . Puede tratarse de una versión destinada a la distribución, o bien de una versión interna de trabajo: por ejemplo, un estado S_{ij} de un proyecto de software puede no ser puesto en producción, en particular si no constituye un estado funcional del software.

[0020] Un juego P_i puede encontrarse en unos estados S_{ij} , con j comprendido entre 1 y un entero J_i . Formalmente:

$$P_i = \{ S_{ij} \mid j \text{ pertenece a } \{1, \dots, J_i\} \}$$

[0021] Cada estado S_{ij} está constituido por el conjunto de los documentos numéricos que componen el juego P_i en la fecha T_{ij} en su estado particular. En este caso, el estado de dicho documento en la fecha T_{ij} puede diferir de su estado en una fecha diferente.

[0022] Un documento particular l del estado j del juego de documento P_i se indica O_{ijl} .

[0023] Dicho de otro modo, el estado particular de un documento O_{ijl} en el estado general S_{ij} del juego P_i puede ser diferente de su estado $O_{ijl'}$ en un estado general $S_{ij'}$ diferente del juego P_i .

[0024] El estado S_{ij} está constituido por el conjunto de los documentos O_{ijl} , con l comprendido entre 1 y un entero L_{ij} . Formalmente:

$$S_{ij} = \{ O_{ijl} \mid l \text{ pertenece a } \{1, \dots, L_{ij}\} \}$$

[0025] Cada estado S_{ij} puede presentar al menos un contribuidor identificado A_{ij} . Se entiende por contribuidor una persona que haya creado el estado S_{ij} , o que haya participado en la creación de este estado.

[0026] Para claridad de la presente descripción, los índices asociados a una letra mayúscula serán a veces representados separados entre sí por una coma (","), Sin embargo, esta representación solamente se utilizará en casos susceptibles de ambigüedad, para evitar hacer pesada la lectura. Por ejemplo, el estado j de un juego de

documentos i se indicará Sij , mientras que el estado 12 ("doce") de un juego de documentos 1 ("un") se indicará preferentemente $S1, 12$.

Herramienta de gestión según la invención

5 **[0027]** Con referencia a la figura 1, la herramienta de gestión de documentos numéricos 100 comprende un módulo de generación de huella SignM 110 dispuesto para ser capaz de asociar a un juego de documentos Pi una huella numérica Ei que comprende un conjunto de firmas numéricas características del contenido de dichos documentos.

10 **[0028]** El módulo SignM 110 está dispuesto para ser capaz de interactuar con una base de datos de trabajo WDB 120. La base WDB 120 es capaz de memorizar al menos temporalmente datos de trabajo para el módulo SignM 110. Se entiende que la base WDB 120 comprende un espacio de almacenamiento de datos organizado, el cual puede tomar la forma de una base de datos en un soporte de memoria permanente (por ejemplo, de tipo disco duro) o también de una memoria temporal direccionable (por ejemplo, de tipo "RAM").

15 **[0029]** La herramienta 100 comprende además una base de datos de juegos de documentos PDB 130 capaz de mantener datos relativos a diferentes juegos de documentos Pi . En particular, la base PDB 130 es capaz de mantener las huellas Ei de juegos de documento Pi . La base PDB 130 puede tomar la forma de una base de datos organizada en un soporte de memoria permanente, por ejemplo de tipo disco duro.

20 **[0030]** El módulo SignM 110 está dispuesto para ser capaz de interactuar con la base PDB 130, en particular para almacenar las huellas Ei generadas por el módulo SignM 110.

[0031] La herramienta 100 también comprende un módulo de comparación de juegos de documentos ComM 140 dispuesto para ser capaz de identificar firmas comunes a unos juegos de documentos Pi diferentes. En este caso, el módulo ComM 140 es capaz de determinar una dependencia entre estos juegos de documentos.

25 **[0032]** El módulo ComM 140 es capaz de interactuar con la base PDB 130 para extraer de la base PDB las huellas Ei generadas a partir de juegos de documentos Pi a comparar, así como datos asociados a estas huellas, en este caso.

[0033] La herramienta 100 comprende finalmente un módulo de gestión de derechos RightsM 150 dispuesto para ser capaz de interactuar con la base PDB 130 y el módulo de ComM 140 para establecer datos útiles para la determinación de derechos en los documentos.

30 Módulo de generación de huella

[0034] El módulo SignM 110 comprende una función de generación de huella de documento signO() capaz de establecer una huella numérica de documento $Eijl$ característica del contenido de un documento $Oijl$. Una huella numérica de documento $Eijl$ comprende un conjunto de firmas numéricas generadas según el contenido del documento $Oijl$.

35 **[0035]** La función signO() se describe de manera general con ayuda de las figuras 2 y 3.

[0036] La función signO() está configurada para ser capaz de descomponer un documento $Oijl$ en elementos de documento $Cijlk$ según una regla escogida (etapa 200). La función signO() está además dispuesta para establecer para cada elemento $Cijlk$ un valor de firma numérica $Nijlk$ característica del contenido de este elemento (etapa 202). Esto puede hacerse por llamada de una función de firma sign(). La función signO() también está configurada para asociar la firma $Nijlk$ característica de un elemento $Cijlk$ y una referencia a este elemento en una huella de elemento, o elemental, $Eijlk$ (etapa 204). La referencia al elemento $Cijlk$ puede comprender el valor de los índices i, j, l y k . Cada huella elemental $Eijlk$ puede ser almacenada en una tabla de huella elemental SignETab[] manteniendo en relación una referencia al elemento $Cijlk$ y el valor de la firma $Nijlk$.

45 **[0037]** Como opción, un atributo de elemento $AEijlk$ puede ser asociado a cada elemento $Cijlk$. El atributo de elemento $AEijlk$ puede ser almacenado en la tabla SignETab[]. Un atributo de elemento $AEijlk$ puede comprender el valor de un número de líneas de código, por ejemplo en el caso en que el elemento es una función y el objeto un archivo fuente. El atributo de elemento $AEijlk$ puede comprender también el valor de un índice de pertinencia del contenido del elemento $Cijlk$. Uno o varios atributos de elementos suplementarios pueden ser almacenados en la tabla SignETab[].

50 **[0038]** La tabla 300 de la figura 3 ilustra un ejemplo de realización de una tabla SignETab[] para el elemento C1, 1,2,2.

[0039] La huella $Eijl$ característica del contenido del documento $Oijl$ comprende el conjunto de las huellas elementales $Eijlk$, siendo k un entero natural que varía entre 1 y un entero natural $Kijl$ (etapa 206). En este caso, la huella $Eijl$ comprende además el conjunto de los atributos de elemento $AEijlk$ asociados a los elementos $Cijlk$ de las huellas elementales $Eijlk$.

60 **[0040]** La base WDB 120 almacena, al menos temporalmente, la huella $Eijl$, en relación con una referencia de temporalidad $TPijl$ del documento $Oijl$. Por ejemplo, la referencia de temporalidad $TPijl$ puede comprender la fecha Tij asociada al estado Sij . La referencia de temporalidad $TPijl$ podría también comprender una fecha $Tijl$ asociada al documento $Oijl$, por ejemplo una fecha de última modificación de este documento.

[0041] La base WDB 120 puede mantener en relación, por ejemplo en una tabla de huella de documento SignOTab []:

65 - un identificador de documento $Oijl$, por ejemplo un identificador compuesto por el valor de los índices i, j y l ,

- una referencia a cada una de las huellas elementales E_{ijkl} del documento O_{ijl} , por ejemplo una referencia compuesta por los índices i, j, l y k, y
- el valor de la referencia de temporalidad TP_{ijl} del documento O_{ijl} , por ejemplo la fecha T_{ij} .

5 **[0042]** Como opción, un atributo de documento AO_{ijl} puede ser asociado a cada documento O_{ijl} . El atributo de documento AO_{ijl} puede ser almacenado en la tabla SignOTab[]. Un atributo de documento AO_{ijl} puede comprender el valor de un tamaño de archivo informático, la identidad de un contribuidor A_{ij} o también una nota que califica un nivel de confianza en la referencia de temporalidad TP_{ijl} . Uno o varios atributos de documento suplementarios pueden ser almacenados en la tabla SignOTab[], por ejemplo para poder asociar a un documento

10 O_{ijl} , a la vez una identidad de contribuidor A_{ij} y un índice de confianza en la referencia de temporalidad TP_{ijl} .
[0043] La estructura de las tablas descrita aquí tiene como propósito ante todo hacer comprender la invención. De este modo, otras estructuras de la tabla SignOTab[] pueden así concebirse en la práctica. Por ejemplo, la tabla SignOTab[] puede comprender un identificador de huella E_{ijl} y este identificador puede estar ligado al documento O_{ijl} . La tabla SignOTab[] también puede comprender el valor de la firma N_{ijkl} de cada una de las huellas elementales E_{ijkl} del documento en cuestión, o también el conjunto de las huellas elementales E_{ijkl} en lugar, o además, de las referencias a estas huellas. Por ejemplo, una realización de la invención puede consistir en construir, para un documento O_{ijl} particular, una tabla en la que cada línea comprende un identificador de un elemento de documento C_{ijkl} (por ejemplo, el valor de los índices i, j, l y k), el valor de la firma N_{ijkl} de este elemento, el valor de la referencia de temporalidad TP_{ijl} del documento, y, en este caso, el valor del atributo de documento AO_{ijl} del documento O_{ijl} y el valor del atributo de elemento AE_{ijkl} del elemento, siendo esta tabla rellena para cubrir el conjunto de los elementos C_{ijkl} del documento O_{ijl} .

15 **[0044]** En la figura 3, la tabla 304, constituida por columnas COL301 a COL308 y líneas ROW310 a ROW313, ilustra una tal realización para un objeto $O_{1,1,L11}$. En esta tabla 304, un identificador de contribuidor A_{11} es almacenado en calidad de atributo de documento $AO_{1,1,L11}$ (COL308), y la fecha T_{11} es almacenada en calidad de referencia de temporalidad $TP_{1,1,L11}$ (COL307).

20 **[0045]** El módulo SignM 110 comprende además una función de generación de huella de estado signS() dispuesta para ser capaz de establecer una huella de estado E_{ij} característica del estado S_{ij} de un juego P_i .

25 **[0046]** La función signS() se describe con ayuda de la figura 4.

30 **[0047]** Un estado S_{ij} comprende un conjunto de documentos O_{ijl} a un instante T_{ij} , l siendo un entero natural que varía entre 1 y un entero natural L_{ij} (etapa 400). La función signS() está configurada para llamar sucesivamente la función signO() para cada uno de los documentos O_{ijl} del estado S_{ij} con la finalidad de obtener para cada documento O_{ijl} una huella de documento E_{ijl} que comprende un conjunto de huellas elementales E_{ijkl} (etapa 402).

35 **[0048]** La huella de estado E_{ij} comprende el conjunto de las huellas elementales E_{ijkl} , k que varía entre 1 y K_{ij} e l que varía entre 1 y L_{ij} (etapa 404).

[0049] La base WDB 120 almacena, al menos temporalmente, la huella E_{ij} .

[0050] La base WDB 120 puede mantener en relación, por ejemplo en una tabla de huella de estado SignSTab[]:

- un identificador de la huella de estado E_{ij} , por ejemplo un identificador compuesto por el valor de los índices i y j , y
- una referencia a cada una de las huellas de documento E_{ijl} del estado S_{ij} , por ejemplo compuesta por los índices i, j , y l .

40 **[0051]** Tal como se ha explicado más arriba, la estructura de la tabla SignSTab[] puede ser diferente en la práctica. Por ejemplo, una realización de la invención puede consistir en construir una tabla que reúne el conjunto de las huellas de documento E_{ijl} , por ejemplo en la realización práctica descrita más arriba, de los documentos del estado S_{ij} . En la figura 3, la tabla 306, constituida por las columnas COL301 a COL308 y las líneas ROW314 a ROW326, ilustra este ejemplo de realización para el estado $S_{1,2}$.

45 **[0052]** Como opción, un atributo de estado AS_{ij} puede ser asociado a cada estado S_{ij} . Este atributo AS_{ij} puede ser almacenado en la tabla SignSTab[]. Por ejemplo, el atributo de estado AS_{ij} podría comprender un índice, o nota, que califica la fiabilidad de la fecha T_{ij} , y/o un índice que califica la fiabilidad de la identidad del contribuidor A_{ij} (por ejemplo, un valor de índice mínimo para una identificación con ayuda de un nombre de conexión, o "login", y un valor de índice máximo para una identificación con ayuda de un proceso de autenticación certificada por un tercero).

50 **[0053]** El módulo SignM 110 también comprende una función de generación de huella de juego de documentos signP () dispuesta para ser capaz de establecer una huella de juego de documentos E_i característica del juego P_i .

55 **[0054]** La función signP() se describe con ayuda de la figura 5.

60 **[0055]** El juego de documentos P_i comprende un conjunto de estados siendo $S_{ij,j}$ un entero natural que varía de 1 a un entero natural J_i (etapa 500). La función signP() está configurada para llamar sucesivamente para cada estado S_{ij} del juego P_i la función signS() con la finalidad de obtener para cada estado S_{ij} una huella de estado E_{ij} que comprende un conjunto de huellas de elementos E_{ijkl} , k que varía entre 1 y K_{ij} y l que varía entre 1 y L_{ij} (etapa 502).

65 **[0056]** Una huella de juego de documentos E_i comprende el conjunto de las huellas elementales E_{ijkl} , k que varía de 1 a K_{ij} , l que varía de 1 a L_{ij} y j que varía de 1 a J_i (etapa 504).

[0057] La base WDB 120 almacena, al menos temporalmente, la huella E_i .

[0058] La base WDB 120 puede mantener en relación, por ejemplo en una tabla de huella de juego de documento SignPTab[]:

- un identificador de la huella de juego de documentos Ei , por ejemplo un identificador compuesto por el valor del índice i , y

5 - una referencia a cada una de las huellas de estado Eij , por ejemplo compuesta por los índices i y j .

[0059] Tal como se ha explicado más arriba, la estructura de la tabla SignPTab[] puede ser diferente en la práctica. Una realización de la invención puede consistir en construir una tabla que reúne el conjunto de las huellas de estados Eij del juego de documentos Pi , por ejemplo en la realización práctica descrita más arriba. En la figura 3, la tabla 308, constituido por las columnas COL301 a COL308 y las líneas ROW301 a ROW339, ilustra un tal ejemplo de realización para el juego de documento $P1$. En esta tabla 308, cada línea mantiene:

- una referencia a un elemento de documento constituida por una columna COL301 que almacena el valor del índice i , por una columna COL302 que almacena el valor del índice j , por una columna COL303 que almacena el valor del índice l , y por una columna COL304 que almacena el valor del índice k , y

15 - el valor de la firma de este elemento de documento en una columna COL305,

- el valor del atributo de este elemento en una columna COL306,

- el valor de la referencia de temporalidad del documento del cual proviene este elemento en una columna COL307, y

20 - el valor del atributo del documento del cual proviene este elemento en una columna COL308.

[0060] El módulo SignM 110 comprende una función de datación de firma signTp() dispuesta para ser capaz de asociar a cada valor de firma Niq diferente de un juego Pi cuya huella es almacenada en la base WDB 120 un dato de firma Viq en función de las referencias de temporalidad $TPijl$ de los documentos $Oijl$ a las cuales dicha firma está asociada en la base WDB 120.

25 **[0061]** La función signTp() se describe con ayuda de la figura 6.

[0062] La función signTp() está configurada para establecer para un valor de firma Niq particular de la base WDB 120, conteniendo el conjunto de las huellas elementales $Eijlk$ este valor de firma, y, a cada vez, la referencia de temporalidad $TPijl$ de la huella de documento $Eijl$ en relación con la huella elemental $Eijlk$ en la base WDB 120.

30 **[0063]** La función signTp() está además dispuesta para mantener en relación para este valor de firma Niq , por ejemplo en una tabla de firma SignTb[]:

- un identificador de una huella elemental $Eijlk$ que contiene dicha firma particular, por ejemplo que comprende los valores i, j, l , y k , y

35 - la referencia de temporalidad $TPijl$, por ejemplo la fecha Tij , de la huella de documento $Eijl$ en relación con la huella elemental $Eijlk$ en la base WDB 120; estando la función SignTp() dispuesta para rellenar la tabla SignTb[] de tal manera que contenga un identificador de cada huella de elemento $Eijlk$ almacenada en la base WDB 120 en la cual se encuentra este valor de firma.

40 **[0064]** En la figura 6, la tabla 600 ilustra un ejemplo de realización de la tabla SignTb[] para un valor de firma Nia . Esta tabla 600 contiene n líneas, representadas por las líneas ROW601 a ROW604, siendo n el número de huellas de elementos $Eijlk$ que comprenden el valor de firma Nia . Los cuádrupletes $\{i, ja1, la1, ka1\}$, $\{i, ja2, la2, ka2\}$ a $\{i, jan, lan, kan\}$ representan cada uno un identificador de una huella de elemento que contiene el valor de firma Nia . Estos cuádrupletes se almacenan en las columnas COL602 a COL605 de la tabla 600. La columna COL606 de la tabla 600 almacena, en calidad de referencias de temporalidad, las fechas $Ti,ja1$ a Ti,jan , en las líneas ROW601 a ROW604.

[0065] La tabla SignTb[] puede ser almacenada, al menos temporalmente, por ejemplo en la base WDB 120.

[0066] La función signTp() se repite para crear una tabla SignTb[] para cada una de los diferentes valores de firma Niq existentes en las huellas de documentos $Eijl$ memorizadas en la base WDB 120.

50 **[0067]** En su defecto, la tabla de firma SignTb[] puede comprender además, para cada huella elemental Ni,jqp,lqp cuya firma vale Niq , el valor de atributo de este elemento AEi,jqp,lqp y el valor de atributo del documento AOi,jqp,lqp que comprende este elemento.

[0068] La función signTp() está además dispuesta para asignar a cada tabla SignTb[] memorizado en la base WDB 120, y relativo a un valor de firma Niq particular, al menos un dato de firma Viq .

55 **[0069]** El dato de firma Viq puede obtenerse aplicando una función de datación dat() predeterminada que utiliza las referencias de temporalidades de la tabla SignTb[]. Dicho de otro modo, el dato de firma Viq asociada al tabla SignTb[] puede ser función de las referencias de temporalidad $TPijl$, por ejemplo las fechas Tij , asociadas a los identificadores de huella $Eijl$ de esta tabla. Por ejemplo, el dato de firma Viq puede ser la fecha Tij más antigua contenida en la tabla SignTb[].

60 **[0070]** Cuando la tabla de firma SignTb[] almacena valores de atributos de documento AOi,jqp,lqp , la función dat() puede estar configurada para determinar el dato de firma Viq en función, además, de estos valores de atributos. Por ejemplo, cuando los atributos de documento AOi,jqp,lqp comprenden un índice de fiabilidad de la referencia de temporalidad TPi,jqp,lqp , la función dat() puede estar configurada para determinar la fecha Ti,jqp más antigua de la tabla SignTb[], excluyendo, para esta determinación, las fechas Ti,jqp asociadas en la tabla SignTb[] a un valor de atributo de documento AOi,jqp,lqp inferior a un valor umbral.

65

[0071] La función signTp() puede además estar configurada para asignar a cada tabla SignTb[] memorizada en la base WDB 120, y relativo a un valor de firma *Niq* particular, un dato de documento *Wiq*. El dato de documento *Wiq* puede obtenerse aplicando una función de autenticación aut() predeterminada que utiliza los atributos de documento *AOi,jqp,lqp* de la tabla SignTb[]. Además, la función aut() puede utilizar los datos de firma *Viq*, y/o incluso las referencias de temporalidad *TPi,jqp,lqp*. Por ejemplo, la función aut() puede estar configurada para determinar en calidad de dato *Wiq* el valor del atributo de documento *AOi,jqp,lqp* cuya referencia de temporalidad *TPi,jqp,lqp* asociada es más antigua. En el caso en el cual la referencia de temporalidad *TPi,jqp,lqp* comprende una fecha *Ti, jqp* y el dato de documento un identificador de contribuidor *Aijqp*, una tal función aut() puede estar configurada para establecer en calidad de dato de documento *Wiq* el identificador del contribuidor más antiguo asociado al valor de firma particular *Niq*.

[0072] En la práctica, una realización puede consistir en construir, para un valor de firma particular *Niq*, una tabla, por ejemplo la tabla 602 de la figura 6, en la cual cada línea comprende:

- una columna que almacena el valor de dicha firma *Niq* (COL601),
- unas columnas que almacenan los valores *i, jqp, lqp* y *kqp* de los índices del elemento de documento cuyo contenido ha generado el valor de firma *Niq* (COL602 a COL605),
- la referencia de temporalidad *TPi jqp, lqp*, aquí la fecha *Ti,jqp*, del documento del cual proviene dicho elemento de documento (COL606),
- el valor del dato de firma *Viq* (COL609), y, en este caso,
- el valor del atributo de dicho elemento *AEiqp,jqp,lqp,kqp* (COL607),
- el valor del atributo del documento del cual proviene dicho elemento de documento *AOi,jqp,lqp* (COL608),
- el valor del dato de documento *Wiq* (COL610); esta tabla se rellena para almacenar el conjunto de las huellas elementales que contiene el valor de firma *Niq*, es decir aquí, las *r* huellas *Ei,jq1,lq1,kq1* a *Ei,jqr,lqr,kqr*.

[0073] Como opción, la función signTp() puede además estar configurada para establecer, para cada valor de firma *Niq*, un dato de elemento *Ziq* (no representado en la figura 6) en función de los valores de los atributos de elementos *AEi, jqp,lqp,kqp* asociados a este valor de firma *Niq*. Por ejemplo, el dato de elemento *Ziq* puede comprender el valor máximo de los valores de atributos de elementos *AEi,jqp,lqp,kqp*.

[0074] La función signTp() está además dispuesta para ser capaz de reunir los datos de firma *Viq* de las tablas de firma SignTb[]. Por ejemplo, la función signTp() está configurada para construir una tabla de huella de juego de documento temporalizada TempSignPTb[] que almacena en relación mutua:

- un identificador del juego de documentos *Pi*, por ejemplo el índice *i*
- cada valor de firma *Niq* diferente
- el dato de firma *Viq* asociada a dicho valor de firma *Niq*.

[0075] Esto equivale, en el ejemplo de la figura 6, a configurar la función signTp() para extraer el dato de firma *Viq*, y el valor del índice *i* de una línea cualquiera que contiene el valor de firma *Niq*, y ello para cada valor de firma *Niq* diferente. En la figura 6, por ejemplo, una tabla TempSignPTb[] podría estar constituida por la intersección de las columnas COL601, COL602 y COL609 con las líneas ROW601, ROW605 y ROW609.

[0076] Se obtiene así un almacenamiento especialmente ligero de la huella de juego de documentos. En el caso en el cual la referencia de temporalidad comprende la fecha *Tij*, y donde la función dat() está configurada para determinar en calidad de dato *Viq* más antiguo unas fechas *Tij*, este dato vale como fecha de primera aparición del valor de firma en el juego de documentos *Pi*. La huella del juego de documento está así constituida por el conjunto de las firmas y por su fecha de primera aparición. Otras informaciones podrán ser almacenadas en función de la función dat() como se verá más adelante.

[0077] En este caso, la función dat() puede tener en cuenta los datos de documento *Wiq* y los datos de elementos *Ziq*.

[0078] En otro modo de realización, la tabla TempSignPTh[] puede mantener además una referencia a cada huella de elemento *Eijkl*, por ejemplo que comprende los valores de los índices *i, j, k* y *l*. Esto permite mantener en la base WDB 120 una conexión entre una referencia a un elemento de documento y el dato de firma correspondiente.

[0079] Según aún otro modo de realización, la tabla TempSignPTb[] puede mantener además una referencia a cada huella de elemento *Eijl*, por ejemplo que comprende los valores de los índices *i, j, k* y *l*, y una referencia al documento *Oijl*, por ejemplo que comprende los índices *i, j* y *l*. Así, para un documento *Oijl* particular, il será posible de establecer los datos de firmas *Viq* de los elementos de documento *Cijlk* correspondiente a este documento. Cuando el dato *Viq* se establece como siendo la fecha más antigua *Tij*, es posible determinar la fecha de primera aparición de un elemento de documento *Cijlk* en el juego *Pi*, independientemente del documento *Oijl* en el cual ha aparecido por primera vez. Así, es posible identificar los elementos de un documento producido antes de la creación de este documento en el seno de un juego de documento.

[0080] Una realización práctica de la tabla TempSignPTb[] puede consistir en construir una tabla, tal como, por ejemplo, la tabla 604 de la figura 6, en la cual cada línea presenta:

- una columna que almacena un valor de firma *Niq* (COL601),
- una referencia a un elemento de documento *Ci,jqp,lqp,kqp* del cual proviene este valor de firma *Niq*, aquí bajo la forma de los valores de los índices *i, j, l* y *k*, (COL602 a COL605)

- la referencia de temporalidad $T_{Pi,jqp,lqp}$ del documento $O_{i,jqp,lqp}$ del cual proviene este elemento de documento $C_{i,jqp,lqp}, k_{qp}$ (COL606), aquí la fecha $T_{i,qp,jqp}$,
 - el valor del dato de firma V_{iq} (COL609), y, en este caso,
 - el valor del atributo $AE_{i,jqp,lqp,kqp}$ del elemento $C_{i,jqp,lqp,kqp}$ (COL607),
 5 - el valor del atributo $AO_{i,jqp,lqp}$ del documento $O_{i,jqp,lqp}$ del cual proviene el elemento de documento $C_{i,jqp,lqp,kqp}$ (COL608),
 - el valor del dato de documento W_{iq} (COL610); siendo dicha tabla rellena para reunir el conjunto de los valores de firma N_{iq} del juego de documentos P_i . Dicho de otro modo, la realización descrita aquí consiste en reunir, en una misma tabla, las tablas del tipo de la tabla 602 construidas para cada uno de los valores de firma N_{iq} del juego de documentos P_i .

[0081] El módulo SignM 110, y en particular la función signTp(), es capaz de interactuar con la base PDB 130 para almacenar el resultado de la reunión de los datos de firma de las tablas de firma.

[0082] Por ejemplo, la base PDB 130 puede así mantener un dato relativo al periodo de existencia de un valor de firma en el juego de documentos P_i .

[0083] En un modo de realización particular, la huella de documento E_{ijl} puede ser reducida para limitar el tamaño de la base de datos o para facilitar una búsqueda ulterior de firmas. Por ejemplo, las referencias a determinadas de las huellas elementales (y las propias huellas elementales) pueden ser suprimidas en función de los valores de las referencias de temporalidad, de los datos de atributo elementos y/o los datos de atributo de documento.

Función de generación de huella de documento

[0084] La función signO() se describe en detalle, en un modo de realización particular, con ayuda de la figura 7.

[0085] Se considera un documento O_{ijl} , es decir un documento l particular del estado S_{ij} de un juego de documentos P_i (etapa 700).

[0086] La función signO() está configurada para llamar una función tipo(), ella misma dispuesta para ser capaz de establecer un dato característico del tipo informático del documento O_{ijl} (etapa 702).

[0087] Por la expresión "tipo informático", se entiende aquí la estructura informática del documento O_{ijl} en el sentido en que esta estructura es común a todos los documentos de mismo tipo. El tipo informático puede ser comúnmente extendido o ser el objeto de estándares, como por ejemplo, los archivos de tipo "texto", los archivos de tipo "Microsoft Word" (marca registrada), los archivos de tipo "imagen" de un formato particular. El tipo también puede ser propio de un software que haya generado el documento.

[0088] La caracterización del tipo del documento O_{ijl} , en el caso particular en que este es un archivo, puede hacerse mediante el reconocimiento de la extensión del nombre del archivo, por ejemplo cuando el documento es un archivo creado en un entorno de tipo "Microsoft Windows" (marca registrada).

[0089] La caracterización también puede hacerse mediante el reconocimiento en el archivo de secuencias de bits características de un tipo de archivo. En particular, el reconocimiento de palabras clave características en archivos fuentes puede permitir identificar un lenguaje de programación. Entonces, la función tipo() es también capaz de determinar para un archivo de tipo fuente el lenguaje utilizado.

[0090] La función signO() está configurada para llamar una función de filtro de contenido filterA(), ella misma dispuesta para ser capaz de establecer a partir del documento O_{ijl} y del tipo de este documento, un documento de contenido depurado O'_{ijl} . (etapa 704)

[0091] Por ejemplo, la función filterA() puede estar configurada para suprimir comentarios insertados en un archivo fuente, caracteres ajenos a la semántica del lenguaje de programación, o también caracteres que dependen de un sistema de explotación particular.

[0092] La función filterA() está configurada para mejorar la robustez de la función signO(), y consecuentemente, la pertinencia de la huella generada por esta función. Se entiende aquí por robustez la capacidad de la función signO() para generar una huella de documento idéntico a pesar de la aportación de modificaciones que se consideran como no significativas para este contenido. Unas modificaciones que pueden ser consideradas como no significativas son por ejemplo:

- las modificaciones que se refieren únicamente a la conformación del documento, por ejemplo de tipo texto, pudiendo estas modificaciones haber sido realizadas de manera automática (por ejemplo, la adición de caracteres de espaciamento consecutivos o también de una línea vacía), y
 - la reescritura de un contenido, como, por ejemplo, el cambio de los nombres de función o de variable, o incluso la supresión o la adición de una mención de copyright.

[0093] Se destaca que la reescritura de un contenido podría ser legítima, o bienintencionada, por ejemplo cuando tiene como por objetivo poner el contenido del documento en conformidad con determinadas reglas formales o convenciones de nominación. La reescritura podría también ser malintencionada, o ilícita, en particular cuando se realiza con el único propósito de atribuirse la calidad de autor, o disimular la reutilización de un contenido.

[0094] Se entiende que la depuración del documento O_{ijl} es función del tipo de este documento.

[0095] La función signO() está configurada para llamar una función de extracción de contenido cut() capaz de aislar en un documento O'_{ijl} unos elementos de documento C_{ijkl} , tales como funciones o procesos en un

documento de tipo código fuente, por ejemplo. Esta localización de elementos individuales es función del tipo del documento *Oijl*. (etapa 706)

[0096] La función `cut()` está configurada para establecer a partir del documento *O'ijl* un conjunto de elementos *Cijlk*, donde *k* es un entero natural comprendido entre 1 y un entero *Kijl*. Por ejemplo, la función `cut()` puede estar configurada para ser capaz de extraer de un documento *O'ijl* de tipo fuente de un lenguaje de programación determinado un conjunto de funciones contenidas en este objeto.

[0097] La función `signO()` está configurada para llamar una función de filtro de contenido `filterB()` dispuesta para ser capaz de establecer a partir de un elemento *Cijlk*, y del tipo del documento *Oijl*, un elemento depurado *C'ijlk*. (etapa 708)

[0098] De manera análoga a la función `filterA()`, la función `filterB()` puede, por ejemplo, estar configurada para suprimir comentarios y caracteres ajenos a la semántica del lenguaje de programación identificado, y/o la supresión de caracteres que dependen de un sistema de explotación. La función `filterB()` se propone mejorar la robustez del módulo `SignM 110`, en el sentido expuesto más arriba.

[0099] La función `filterB()` tiene como resultado un conjunto de elementos depurados *C'ijlk*, con *k* un entero natural comprendido entre 1 y un entero *Kijl*.

[0100] La función `signO()` está además dispuesta para llamar una función de generación de firma `sign()` dispuesta para ser capaz de establecer a partir de un elemento depurado *C'ijlk* una firma *Nijlk* característica del contenido de este elemento de documento. (etapa 710)

[0101] La firma *Nijlk* puede tomar la forma de una cadena de caracteres de longitud determinada, por ejemplo una cadena de caracteres ASCII o hexadecimales. Preferentemente, la función `sign()` comprende un algoritmo de generación de firma llamado de "hasheado". La firma *Nijlk* es única para el contenido de un elemento *C'ijlk* determinado. En todos los casos, la función `sign()` está preferentemente determinada para presentar una fuerte probabilidad de no ocurrencia, es decir de manera que la probabilidad de que una misma firma se obtenga a partir de contenidos diferentes de elementos *C'ijlk* sea lo más reducida posible. Además, la función `sign()` se escoge como no reversible, es decir dispuesta de manera que sea en la práctica imposible establecer el contenido del elemento a partir de su firma: entre otras cosas, ello permite el almacenamiento a largo plazo de las firmas sin preocuparse de la confidencialidad, puesto que el contenido explícito de los documentos no se almacena ni se divulga.

[0102] Se destaca aquí que la utilización de una cadena de caracteres de longitud determinada en calidad de firma del contenido de un documento facilita la comparación de documentos puesto que es más simple comparar objetos informáticos de tamaño constante antes que comparar objetos de tamaños diferentes, y más voluminosos. Además, estas cadenas de caracteres permiten la comparación de juegos de documentos y pueden ser ampliamente difundidas puesto que no divulgan el contenido.

[0103] Dicho algoritmo de generación de firma, o de "hasheado", puede ser del tipo MD5 (para "Mensaje Digest 5") o SHA-1 (para "Secure Hash Algorithm"), por ejemplo.

[0104] La función `sign()` se llama para cada elemento *C'ijlk* para obtener un conjunto de firmas *Nijlk*, con *k* un entero natural que varía entre 1 y *Kijl*, a partir de un documento *Oijl*.

[0105] Preferentemente, la función `signO()` está configurada para llamar una función de notación `eval()`, ella misma dispuesta para ser capaz de asociar a una firma *Nijlk* un dato de pertinencia de firma *Bijlk* en calidad de atributo de elemento *AEijlk*. (etapa 712)

[0106] Se entiende aquí por pertinencia una estimación del interés de un elemento de documento con respecto a una contribución o una modificación aportada al documento. La función `eval()` puede estar configurada para tener en cuenta uno o varios criterios de entre:

- el tamaño del elemento de documento, por ejemplo un número de líneas de código, o de caracteres, en el caso de una función de un archivo fuente;
- la presencia de la firma en una lista de firmas asociada a un índice de pertinencia reducida, pudiendo esta lista ser específica de un tipo de documento;
- el número de contribuidores a este documento y/o la actividad en términos de contribuciones sobre este documento.

[0107] Obviamente, esta enumeración no es limitativa, y la función `eval()` puede tener en cuenta otros criterios.

[0108] Cada elemento depurado *C'ijkl* puede ser considerado como "atómico", en el sentido en que no es interesante recortarlo aún más, puesto que perdería entonces lo que está ligado a su naturaleza (propiedades y significación, en especial). Así, su firma *Nijlk* podría ser vista como una "huella atómica", por oposición a la huella *Eijl* (global) del documento.

[0109] La función `signO()` está además dispuesta para asociar cada firma *Nijlk* a una referencia de documento *Oijl*, por ejemplo que comprende los índices *i, j, l* en una huella elemental *Eijlk*. (etapa 714)

[0110] La función `signO()` permite obtener una huella de documento *Eijl* característica del objeto *Oijl* que comprende un conjunto de huellas elementales *Eijlk*, *k* que varía entre 1 y *Kijl*. (etapa 716).

[0111] La huella de documento *Eijl* puede ser almacenada en la base WDB 120 tal como se expuso más arriba.

[0112] Se entiende que la referencia de temporalidad puede ser la fecha *Tij* del estado *Sij*. La referencia de temporalidad podría ser también determinada por el índice *j* solamente, en el caso en el cual las *j* están cronológicamente ordenadas.

[0113] En un modo de realización particular, la huella de documento E_{ijl} comprende también un identificador de contribuidor A_{ij} en calidad de atributo de documento AO_{ijl} que haya creado el estado S_{ij} o participado en la creación de este estado, en calidad de dato de documento.

[0114] Se ha visto que las funciones $filterA()$ y $filterB()$ están dispuestas para evitar que un elemento de documento genere una firma distinta tras unas modificaciones no significativas de su contenido. Sin embargo, estas funciones pueden además estar dispuestas de manera que la huella de un documento particular, cuyo contenido ha sido modificado de manera no significativa, difiera parcialmente de su huella de origen, con la finalidad de obtener una traza de estas modificaciones. Esto puede ser realizado adaptando las funciones $filterA()$, $filterB()$ y $cut()$ las unas con respecto a las otras, por ejemplo para recortar un contenido en elementos cuya firma deberá o no cambiar y adaptar la función $filterB()$ para hacerlo.

[0115] Por ejemplo, se considera la mayoría de las veces el cambio único de los nombres de variables en el interior de una función de un documento como una simple reescritura de contenido. Las funciones $filterA()$, $filterB()$ y $cut()$ pueden estar dispuestas para obtener una huella de documento que contiene determinadas firmas idénticas relativas al contenido de esta función y de otras firmas diferentes para indicar que una reescritura ha sido realizada. Por ejemplo, cuando la reescritura concierne únicamente el cambio del nombre de una función, las funciones $filterA()$, $filterB()$ y $cut()$ pueden por ejemplo estar dispuestas de manera que la función del documento se descomponga en un primer elemento de documento correspondiente a la parte declarativa de esta función y un segundo elemento de documento correspondiente al cuerpo, o contenido, de la función, y que la firma del primer elemento cambie mientras que la del segundo permanezca sin cambiar.

[0116] Las funciones $filterA()$, $filterB()$ y $cut()$ pueden en determinados casos estar dispuestas para conservar un documento idéntico, en particular en función del tipo de documento.

[0117] Se entiende que la función $signO()$ debe ser robusta para permitir la comparación ulterior de huellas. Para que dos huellas puedan ser comparadas (es decir para que se busquen firmas comunes), un mismo documento debe necesariamente generar la misma huella cualquiera que sea el módulo SignM 110 utilizado. Las funciones $cut()$, $filterA()$, y $filterB()$ tienen por lo tanto tendencia, a priori, a ser divulgadas y estandarizadas si se desea emplear herramientas de generación de huella diferentes. Sin embargo, esta divulgación favorece la colocación de estrategias destinadas a crear fallos en la robustez de estos filtros, en particular con respecto a reescrituras, en el caso de una reescritura de contenido malintencionada en especial.

[0118] En un modo de realización particular de la función $signO()$, una función de verificación $ver()$ puede ser llamada previamente a la etapa 706 para asegurar la conformidad del código fuente con determinadas reglas formales, dependiendo eventualmente del lenguaje de programación. Por ejemplo, cuando el documento O_{ijl} es un archivo fuente, la función de verificación $ver()$ puede comprender un analizador gramatical de código fuente capaz de establecer una representación abstracta del código fuente conforme a las reglas de sintaxis y de gramática del lenguaje. Este analizador gramatical puede él mismo ser insensible a las modificaciones no significativas o a la reescritura de contenido. Por ejemplo, la función $ver()$ puede estar configurada para establecer un árbol de sintaxis abstracto o "AST" (para Abstract Syntax Tree). Haciendo esto, la función $signO()$ es más robusta en el sentido expuesto más arriba.

Función de datación de firma

[0119] En un modo de realización particular, la función $dat()$ está configurada para establecer en calidad de dato de firma Viq un periodo de existencia de la firma Niq considerada.

[0120] En este modo de realización, el dato de firma Viq puede comprender una fecha de aparición $TAiq$, o de primera presencia, del valor de firma Niq en el juego de documento Pi y una fecha de desaparición $TDiq$, o fecha de última presencia. En este caso, la función $dat()$ puede estar configurada para establecer, en calidad de fecha de aparición $TAiq$, más antigua de las referencias de temporalidades, por ejemplo la más antigua de las fechas Ti,jq de la tabla $SignTb[]$, y más reciente de estas fechas en calidad de fecha de desaparición $TDiq$.

[0121] Según una variante de realización, la función $dat()$ está configurada para establecer en calidad de dato de firma Viq un conjunto de periodos de existencia. La función $dat()$ puede estar configurada para interactuar con la base WDB 120 de tal manera que pueda extraer de esta base una tabla de estados $STb[]$ que reúne cada una de las fechas Tij diferentes presentes en la huella Ei del juego de documento Pi . La función $dat()$ puede estar configurada para comparar cada una de las fechas Ti,jqp de la tabla $SignTb[]$ con las fechas reunidas en la tabla $STb[]$ para determinar unas fechas Tij a las cuales la firma considerada está, o según el caso no está, asociada.

[0122] En lugar de la fecha Tij , la función $dat()$ puede estar configurada para trabajar sobre identificadores de versión, por ejemplo los diferentes valores del índice j .

[0123] En otro modo de realización, los estados S_{ij} ordenados, es decir que la superioridad de un índice j a un índice j implica que la fecha Tij es más reciente que la fecha Tij .

[0124] La función $dat()$ puede entonces estar configurada para establecer las fechas $TAiq$ y $TDiq$ de manera diferente. La función $dat()$ se describe con ayuda de la figura 8, en este modo de realización particular.

[0125] La función $Dat()$ está configurada para ser capaz de comparar el estado S_{ij} , caracterizado por su huella de estado E_{ij} constituida por el conjunto de las huellas de elementos E_{ijkl} (etapa 800) al conjunto de los estados $S_{ij'}$ anteriores caracterizado por el conjunto de los valores de firma Niq' asociados a estos estados anteriores, es decir para los cuales existe un cuádruplete i, j, l y k' , tal como el valor de firma $N_{ij'l'k'}$ vaille Niq' (etapa 802). Para cada uno de los valores de firma Niq del estado S_{ij} , es decir para cualquier entero q tal que existe un cuádruplete i, j, l y k , tal como N_{ijkl} valga Niq (etapa 804), la función $dat()$ verifica si existe un valor de firma Niq' asociada a un estado $S_{ij'}$ en la base WDB 120 (etapa 806).

[0126] En caso afirmativo (etapa 808), entonces el valor de firma Niq existía en un estado anterior j del juego de documentos Pi . La fecha de aparición $TAiq'$ del valor de firma Niq' en el juego de documentos Pi permanece sin cambiar. La fecha de desaparición $TDiq'$ del valor de firma Niq' en el juego de documentos Pi recibe la fecha Tij del estado Sij .

5 **[0127]** Sino (etapa 810), el valor de firma Niq es nuevo en el juego de documentos Pi . Las fechas de aparición $TAiq$ y de desaparición $TDiq$ adoptan ambas el valor de la fecha Tij del estado Sij .

[0128] En este modo de realización, la huella temporalizada del juego de documento puede ser rápidamente actualizada durante la adición de una huella de estado ulterior Eij , por ejemplo de un último estado conocido. Este modo de realización particular permite construir la huella temporalizada de un juego de documento mediante adiciones sucesivas de sus huellas de estado sucesivas. Por actualizada, se entiende aquí la adición de los valores de firmas que no existían en el estado precedente y el cálculo de los datos de firmas Viq que son modificados por la adición del nuevo estado.

10 **[0129]** Dicho de otro modo, la función `dat()` puede estar configurada para comparar un estado Sij cuya huella Eij se memoriza en la base WDB 120 y un conjunto de estados anteriores Sij' , siendo j un entero natural inferior a j , cuyas huellas Eij' son memorizadas en la base PDB 130, para actualizar la huella temporalizada del juego de documento Pi .

Modo de realización particular del módulo de generación de huella

20 **[0130]** En un modo de realización particular, el módulo SignM 110 está dispuesto para interactuar con una herramienta de gestión de versión, por ejemplo de tipo CVS o equivalente.

[0131] Las herramientas de gestión de versión son capaces de organizar el almacenamiento de juegos de documentos y de sus diferentes estados. En particular, las herramientas de gestión de versión pueden gestionar de manera incremental y coherente los diferentes estados de un juego de documento: la utilización de una herramienta de gestión de versión permite obtener la ordenación de los estados Sij mencionada más arriba.

25 **[0132]** Cuando varios contribuidores son susceptibles de modificar simultáneamente un mismo juego de documento, una herramienta de gestión de versión permite la identificación del estado a partir del cual las modificaciones han sido hechas por el contribuidor. Determinadas herramientas de gestión de versión solamente almacenan las modificaciones sucesivas, y/o son capaces de verificar, en este caso, la ausencia de conflicto entre modificaciones simultáneas realizadas en todo o parte de un mismo documento. Una herramienta de gestión de versión también puede imponer que la actualización se haga obligatoriamente con respecto a la última versión conocida, antes que con respecto a la versión de origen cuando es conocida. En la práctica, el funcionamiento de las diferentes herramientas de gestión de versión difieren: aquí, se utiliza la coherencia entre los estados Sij provenientes de la utilización de las herramientas de gestión de versión para mejorar el procedimiento de generación de huella temporalizada.

30 **[0133]** La herramienta de gestión de versión proporciona también un identificador del contribuidor que ha hecho la actualización, y/o la lista de los documentos modificados desde un estado cualquiera.

[0134] Por ejemplo, unas herramientas de tipo CVS pueden almacenar el último estado de un documento $Oijl$ y las diferencias entre los diferentes estados anteriores de los documentos $Oiml$ sucesivos, siendo m un entero natural inferior a j . Aquí, el índice l está asociado independientemente del índice de estado j a un archivo caracterizado de manera única por su nombre y su camino de acceso.

35 **[0135]** La herramienta de gestión de versión proporciona también una fecha de actualización, por ejemplo en el servidor centralizado. Esta fecha es almacenada en calidad de referencia de temporalidad del estado del juego de documento.

40 **[0136]** Un estado puede corresponder a una actualización de la versión presente en el servidor. También, típicamente un contribuidor está asociado a un estado.

[0137] Se destaca aquí que según la herramienta de versión utilizada, o según la utilización que se hace de ella, puede ser asociada:

50 - ya sea a un nuevo estado a partir del momento en que un documento del juego de documentos se modifica,
- ya sea a un nuevo estado para una modificación de varios documentos en el transcurso de una misma contribución.

[0138] Esto solamente modifica muy ligeramente el funcionamiento de la herramienta 100, en particular aumentando o disminuyendo el número de estados.

[0139] En este modo de realización, el módulo SignM 110 puede comprender una función de comparación de versión `compS()`.

[0140] La función `compS()` se describe con ayuda de la figura 9.

[0141] Para simplificar la descripción, se hace aquí la hipótesis de que dos estados Sij' y Sij son sucesivos.

60 **[0142]** Aquí, la herramienta es de tipo CVS. En este caso, el índice l caracteriza el documento independientemente del estado.

[0143] Para cada documento $Oijl$ del juego de documentos Pi en un estado Sij , la función `compS()` está configurada para determinar si existe un documento $Oij'l$ del juego de documentos Pi en un estado anterior Sij' (etapa 900)

65 **[0144]** Sino, la función `compS()` está configurada para llamar la función `signO()` descrita más arriba para generar el conjunto de las huellas de elementos $Eijlk$ para el documento $Oijl$. (etapa 902)

[0145] Cada huella de elemento *Eijlk*, y en particular la firma *Nijlk* que comprende, es almacenada en la base WDB 120 tal como se describe más arriba. Una referencia de temporalidad *TPijl* del documento *Oijl*, por ejemplo la fecha *Tij* de la actualización (o contribución) obtenida con ayuda de la herramienta de gestión de versión, es también almacenada en la base WDB 120. Preferentemente, un atributo de documento *AOijl*, por ejemplo un

5

identificador del contribuidor *Aij* del estado *Sij*, es también almacenado en la base WDB 120 (etapa 904). En este caso, el atributo de documento *AOijl* podría comprender un identificador de cada contribuidor *Aij* en el estado *Sij*.
[0146] Si el test de la etapa 900 es positivo, entonces la función compS() está configurada para verificar si el documento *Oijl* es idéntico al documento *Oij'l* (etapa 906). Esto puede ser realizado comparando estados. En determinados casos, la herramienta de gestión de versión es capaz de realizar esta verificación.

10

[0147] En caso afirmativo, entonces la función compS() está configurada para asociar cada firma de las huellas de elementos *Eij'lk'* al documento *Oijl*. Dicho de otro modo, la función compS() establece como huella *Eijl* del documento *Oijl*, la huella *Eij'l* del documento *Oij'l*, con excepción de la referencia de temporalidad *TPijl*, que es la del documento *Oijl*. (etapa 908)

15

[0148] En este modo de realización, no es necesario que la función compS() esté dispuesta para generar el conjunto de las huellas de elementos *Eijlk* del documento *Oijl*. La función compS() puede utilizar las huellas de elementos *Eij'lk'* del documento *Oij'l*. La utilización de una herramienta de gestión de versión que interactúa con el módulo SignM 110 permite ahorrar recursos materiales, en particular carga de la unidad de cálculo.

20

[0149] Si el test de la etapa 906 es negativo, entonces los documentos *Oijl* y *Oij'l* son diferentes. El documento *Oijl* puede entonces ser visto como una transformación del objeto *Oij'l* sobrevenida en la fecha *Tij* y que tiene por contribuidor *Aij* (etapa 910). En la figura 9, esta transformación ha sido representada bajo la forma de una función *hijl*().

25

[0150] La función compS() está configurada para llamar la función signO() para generar el conjunto de las huellas de elementos *Eijlk* del documento *Oijl* tal como se describe más arriba (etapa 912). Las huellas de elementos *Eijlk* son almacenadas en la base WDB 120, en este caso en correspondencia con una referencia de temporalidad *TPijl* (por ejemplo la fecha *Tij*) y un atributo de documento *AOijl* (por ejemplo un dato de identificador de contribuidor *Aij*).

30

Estructura de la base de datos de juegos de documentos

30

[0151] Un ejemplo de estructura de la base PDB 130 se ilustra figura 10.

[0152] Para cada juego de documentos *Pi*, la base PDB 130 puede estar configurada para mantener una tabla de huella temporalizada de juegos de documentos *TpSignPTb[] 1000*.

[0153] Cada fila de esta tabla *TpSignPTb[] 1000* puede comprender:

35

- una columna de firma 1002 que almacena un valor de firma *Niq* particular del juego de documento *Pi*, distinta del valor de firma de una otra fila,

- una columna de referencia al juego de documentos 1004, por ejemplo que almacena el valor del índice *i*, y

- una columna de dato de firma 1006 que almacena el dato de firma *Viq* asociada al valor del dato de firma *Niq*, por ejemplo aquí la fecha *TAiq* de aparición del valor de firma *Niq* en el juego de documentos *Pi*.

40

[0154] En este caso, cada fila de esta tabla *TpSignPTb[] 1000* puede comprender además:

- una columna de dato de atributo de elemento 1008 que almacena el dato de elemento *Ziq*, por ejemplo aquí el valor de un índice de pertinencia *Biq*,

45

- una columna de dato de documento 1010 que almacena un dato de documento *Wiq*, por ejemplo aquí un identificador de contribuidor *Aiq*.

[0155] Cada fila de la tabla *TpSignPTb[] 1000* puede ser considerada como una huella de elemento temporalizada *Eiq*.

50

[0156] La tabla *TpSignPTb[] 1000* puede comprender además otros datos, en particular los datos establecidos por el módulo SignM 110 y/o temporalmente almacenados en la base WDB 120.

[0157] Una realización particular puede consistir en almacenar la tabla de la figura 6 en calidad de tabla *TpSignPTb[] 1000* para un juego de documento particular.

55

[0158] La base PDB 130 puede entonces estar configurada para almacenar una tal tabla para cada juego de documentos *Pi*, o para reunir las tablas relativas a los diferentes juegos de documentos *Pi* en una misma tabla. En la figura 11 por ejemplo, la tabla 1100 reúne una tabla 1102 análoga a la tabla 604 para un juego de documentos *P1* constituido por las columnas COL1101 a COL1110 y las líneas ROW1101 a ROW1113, una tabla 1104 análoga a la tabla 604 para un juego de documentos *P2* constituida por las columnas COL1101 a COL1110 y las líneas ROW1114 a ROW1126, así como una tabla 1106 análoga a la tabla 604 para un juego de documento cualquiera *Pi* constituida por las columnas COL1101 a COL1110 y las líneas ROW1128 a ROW1140.

60

Primer modo de realización del módulo de comparación de juegos de documentos

65

[0159] La herramienta según la invención comprende un módulo de comparación de juegos de documentos ComM 140 dispuesta para establecer un conjunto de valores de firmas asociadas en la base PDB 130 a la vez a un juego de documentos *Pi* y a un juego de documentos diferente *Pi'*. Esto puede permitir determinar cuales de

los elementos de documentos O_{ijl} del juego de documentos P_i se vuelven a encontrar en otro juego de documentos P_i' .

[0160] El módulo ComM 140 está además dispuesto para establecer una dependencia entre un juego de documentos P_i y un juego de documentos P_i' , determinando la integración en el juego de documentos P_i de elementos de documentos del juego de documentos P_i' , e inversamente la integración en el juego de documentos P_i' de elementos de documentos del juego de documentos P_i . Dicho de otro modo, el módulo ComM 140 está dispuesto para establecer dependencias entre juegos de documentos y para orientar estas dependencias.

[0161] Se destaca que, por consideraciones prácticas, hace falta que las funciones filterA(), filterB(), y cut() estén configuradas de tal manera que, para unos contenidos idénticos, la función signO() (o una función equivalente) conduzca a al menos un subconjunto de valores de firmas comunes a estos contenidos. Se podrá desear aplicar las funciones filterA(), filterB(), y cut(), dispuestas de manera idéntica, a los diferentes contenidos a comparar de tal manera que se puedan obtener, mediante la función signO(), unos conjuntos de valores de firmas idénticas cuando los contenidos son idénticos.

[0162] El módulo ComM 140 comprende una función de comparación de juegos de documentos comp(). La función comp() está configurada para comparar un primer juego de documentos P_1 con un segundo juego de documentos P_2 para determinar los elementos de documentos comunes a los juegos de documentos P_1 y P_2 .

[0163] La función comp() se describe con ayuda de la figura 12.

[0164] La función comp() está configurada para extraer un subconjunto $E'1$ de la base PDB 130 que contiene los valores de firma $N1q$ relativos al juego de documentos P_1 así como los datos de firmas $V1q$ correspondientes (etapa 1200). Se supone aquí que existen Q_1 valores de firmas $N1q$ diferentes asociadas al juego de documentos P_1 .

[0165] Como opción, la función comp() está configurada para extraer además el conjunto de los datos asociados a los valores de firmas $N1q$ en la base PDB 130. Por ejemplo, la función comp() puede estar configurada para extraer una tabla del tipo de la tabla 1102 de la figura 11.

[0166] De la misma manera, la función comp() está configurada para extraer un subconjunto $E'2$ de la base PDB 130 relativo al juego de documentos P_2 (etapa 1202), y constituido por Q_2 valores de firmas $N2q'$ diferentes. Por ejemplo, la función comp() puede estar configurada para extraer una tabla del tipo de la tabla 1104 de la figura 11.

[0167] La función comp() está configurada para construir un conjunto de intersección $E'12$ constituido por la reunión de subconjuntos de los conjuntos $E'1$ y $E'2$ cuyos valores de firmas $N1q$ y $N2q'$ son iguales (etapa 1204). Dicho de otro modo, la función comp() está configurada para establecer una tabla PintTb[] que agrupa los valores de firmas $N1q$ y $N2q'$ de $E'1$ y de $E'2$ idénticos así como sus datos de firmas respectivos $V1q$ y $Y2q'$. En este caso, la función comp() puede estar configurada para almacenar además determinados, o todos, de los datos asociados a estos valores de firmas en la base PDB 130.

[0168] La función comp() está configurada para establecer un conjunto de dependencia $E''12$. Por ejemplo, la función comp() puede estar configurada para construir una tabla de dependencias PDepTb[] en la que cada fila representa una dependencia orientada entre los juegos de documentos P_1 y P_2 (etapa 1206).

[0169] Se entiende por "dependencia" la presencia de un valor de firma $N1q$ asociado al juego de documentos P_1 en el conjunto de los valores de firma $N2q'$ asociados al juego de documentos P_2 , e inversamente. Dicho de otro modo, una dependencia entre los juegos de documentos P_1 y P_2 revela la presencia en uno de estos juegos de un elemento de documento del otro juego.

[0170] La función comp() está configurada para establecer la tabla PdepTb[] de tal manera que cada línea de esta tabla comprenda:

- un valor de la firma Nm ,
- una referencia a un juego de documentos de origen POm para este valor de firma (es decir a uno de los juegos P_1 o P_2),
- una referencia a un juego de documentos de destino PDm para este valor de firma (es decir al otro de los juegos P_1 o P_2); siendo la tabla PdepTb[] rellenada para censar cada valor de firma Nm del conjunto $E'12$, siendo las referencias a los juegos de documentos de origen POm y de destino PDm determinadas con ayuda de una función de orientación dir () a partir de los datos de firmas $Y1m$ y $Y2m$, es decir de los valores de los datos de firmas $V1q$ y $V2q'$ correspondientes al valor de firma Nm en los conjuntos $E'1$ y $E'2$, respectivamente. Por juego de documentos de origen, se entiende el juego de documentos desde el cual el elemento de documento $Cijkl$ de valor de firma Nm ha sido más probablemente importado en el juego de documentos de destino.

[0171] Cada fila de la tabla PdepTb[] representa una de dependencia orientada que indica una referencia a un juego de documentos "de origen" POm y una referencia a un juego de documentos "de destino" PDm , para un valor de firma Nm .

[0172] Como opción, la tabla PdepTb[] puede además almacenar para cada valor de firma Nm :

- un atributo de dependencia Fm , que puede ser establecido a partir de los datos de firma $V1m$ y $V2m$, por ejemplo una fecha de primera aparición del valor de firma Nm en el juego de documentos de origen POm ,
- un atributo de dependencia de documento Gm , que puede ser establecido a partir de los datos de atributos de documentos $W1q$ y $W2q'$, por ejemplo un dato que caracteriza el tipo del documento del cual proviene el elemento generador del valor de firma, o un índice de fiabilidad de la referencia de temporalidad

- un atributo de dependencia de elemento Hm , que puede ser establecido a partir de los datos de atributos de elemento $Z1q$ y $Z2q$, por ejemplo una indicación del tamaño del elemento generador del valor de firma Nm .

5 **[0173]** En el caso en el cual el atributo de dependencia de documento Gm comprende un dato que caracteriza el tipo del documento, es posible excluir los valores de firmas Nm cuyos elementos generadores provienen de documentos de tipo diferente.

[0174] La determinación de los juegos de documentos de origen POm y de destino PDm puede tener en cuenta diferentes criterios. La función $dir()$ puede estar configurada consecuentemente.

10 **[0175]** En particular, la función $dir()$ puede estar configurada para determinar en calidad de juego de documento de origen POm el juego de documentos en el cual el elemento de documento generador del valor de firma Nm ha aparecido primero. En este caso, el juego POm es el juego de documentos en la huella del cual el valor de firma Nm ha aparecido antes, mientras que el juego PDm puede ser el juego de documentos en el cual este valor de firma Nm solamente ha aparecido más tarde.

15 **[0176]** En un modo de realización particular, los datos de firma $V1m$ y $V2m$ comprenden cada uno una fecha de aparición $TA1m$ y $TA2m$ del valor de firma Nm en los juegos de documentos $P1$ y $P2$, respectivamente. La función $dir()$ puede entonces estar configurada para determinar en calidad de juego de origen POm , el de los juegos $P1$ y $P2$ correspondiente a la más antigua de las fechas $TA1m$ y $TA2m$. Siendo el otro de los juegos $P1$ y $P2$ determinado en calidad de juego de destino PDm .

20 **[0177]** Como opción, la función $comp()$ puede estar configurada para almacenar las fechas $TA1m$ y $TA2m$ en calidad de atributo de dependencia Fm . El atributo Fm corresponde entonces a un intervalo de tiempo en el transcurso del cual se ha producido una integración de contenido del juego documentos POm en el juego de documentos PDm .

25 **[0178]** En otro modo de realización, el dato de firma $V1m$ (respectivamente $V2m$) comprende una fecha de aparición $TA1m$ (respectivamente $TA2m$) en el juego de documentos $P1$ (respectivamente $P2$) del valor de firma $N1m$ (respectivamente $N2m$) y una fecha de desaparición $TD1m$ (respectivamente $TD2m$) de este valor de firma Nm de este juego. En este caso, la función $comp()$ puede estar configurada para establecer los juegos de origen POm y PDm como anteriormente y para mantener en calidad de atributo Fm la fecha de desaparición $TD1m$ o $TD2m$ en el juego de documentos de destino PDm .

30 **[0179]** Como opción, y para los modos de realización de más arriba, los datos de documentos $W1m$ y $W2m$ comprenden un índice de fiabilidad de los datos de firma $V1m$ y $V2m$, en particular cuando estos datos comprenden unas fechas de aparición $TA1m$ y $TA2m$. La función $comp()$ puede entonces estar configurada para rellenar la tabla $PdepTb[]$ únicamente con los valores de firmas Nm cuyo dato de documento $W1m$ o $W2m$ es superior a un valor umbral determinada.

35 **[0180]** Más generalmente, la función $dir()$ puede estar configurada para determinar los juegos POm y PDm considerando datos de atributo de elemento $Z1m$ y $Z2m$, datos de atributos de documentos $W1m$ y $W2m$, datos de atributos de estados $X1m$ y $X2m$, y/o datos de firmas $V1m$ y $V2m$.

[0181] Por ejemplo, la función $comp()$ puede estar configurada para establecer en calidad de juego POm el de los juegos $P1$ y $P2$ correspondiente a la más antigua de las fechas $TA1m$ y $TA2m$, con la condición de que el índice de pertinencia $B1m$, respectivamente $B2m$, correspondiente sea superior a un valor determinado.

40 **[0182]** Como opción, la función $comp()$ puede estar configurada para suprimir de la tabla de dependencia $PdepTb[]$ los valores de firma Nm , y los datos asociados a estas, de los cuales uno de los índices de pertinencia $B1m$ y $B2m$ es inferior a un valor predeterminado o también cuyas fechas de aparición $TA1m$ o $TA2m$ y $TD1m$ o $TD2m$ están espaciadas en un tiempo superior a un valor fijado.

45 Funciones de agregación de las dependencias

[0183] La función $comp()$ puede estar configurada para llamar una función de agregación capaz de reunir las dependencias de la tabla $PdepTb[]$. Esto es especialmente ventajoso cuando el número de dependencias en esta tabla $PdepTb[]$ es elevado.

50 **[0184]** Un primer ejemplo de función de agregación $agrDep1()$ se describe con ayuda de la figura 13.

[0185] La función $agrDep1()$ está configurada para repartir el conjunto de las dependencias $E''12$ según la orientación de las dependencias en dos subconjuntos de dependencias $D12$ y $D21$ que corresponden respectivamente:

- 55 - al conjunto de las dependencias de $E''12$ tales que POm vale $P1$, y
- al conjunto de las dependencias de $E''12$ tales que POm vale $P2$. (etapa 1300)

[0186] Por ejemplo, la función $agrDep1()$ puede estar configurada para repartir la tabla $PdepTb[]$ en dos sub-tablas $Pdep12 Tb[]$ y $Pdep21 Tb[]$ que corresponden respectivamente:

- 60 - a las líneas de la tabla $PdepTb[]$, es decir a los valores de firma Nm y a los datos asociados a estos valores, tales que POm vale $P1$, y
- a las líneas de la tabla $PdepTb[]$, es decir a los valores de firma Nm y a los datos asociados a estos valores, tales que POm vale $P2$.

65

[0187] El conjunto *D12* comprende el conjunto de los valores de firma *Nm* cuya fecha de aparición asociada *TA1m* es más antigua que la fecha de aparición asociada *TA2m* (dicho de otro modo inferior a esta fecha). Asimismo, el conjunto *D21* comprende el conjunto de los valores de firma *Nm* cuya fecha de aparición asociada *TA2m* es más antigua que la fecha de aparición asociada *TA1m*.

5 **[0188]** Se describe en primer lugar la agregación del subconjunto *D12* por la función *agrDep1()*.

[0189] Como opción, la función *agrDep1()* puede estar configurada para organizar los conjuntos *D12* y *D21* de tal manera que pueda reunir las dependencias según la o las fechas asociadas a su valor de firma *Nm*.

[0190] La función *agrDep1()* está configurada para establecer un número *U12* de valores de firmas *Nm* del conjunto *D12*. Esto puede hacerse con ayuda de una función de censo *count()* (etapa 1302).

10 **[0191]** La función *agrDep1()* está configurada para establecer una fecha *T12* en la cual un elemento cualquiera del juego *P1* ha sido integrado por primera vez en el juego *P2*. Dicho de otro modo, la fecha *T12* es la fecha *TA2m* más antigua asociada a los valores de firmas *Nm* que pertenecen al subconjunto *D12*. (etapa 1304)

15 **[0192]** La función *agrDep1()* está configurada para establecer un subconjunto de dependencias, o agregado, *De12u* de dependencias que pertenecen al conjunto *D12* cuya fecha de aparición *TA1m* del valor de firma *Nm*, o fecha en la cual una firma idéntica a la firma *Nm* ha sido por primera asociada al juego *P1*, es más antigua que la fecha *T12*. Dicho de otro modo, el conjunto *De12u* comprende los valores de firmas *Nm*, y datos asociados a estos valores en este caso, que pertenecen al conjunto *D12* cuya fecha de aparición *TA1m* es más antigua que la fecha *T12* (etapa 1306)

20 **[0193]** La función *agrDep1()* está configurada para establecer una fecha *T12'* definida como siendo la más antigua de las fechas asociadas a los valores de firmas *Nm* de las dependencias que pertenecen al conjunto *D12* pero que no pertenecen pas al agregado de dependencias *De12u*. Dicho de otro modo, la fecha *T12'* representa la fecha más antigua en la cual un elemento de documento del juego *P1* ha sido integrado en el juego *P2* pero cuya firma *Nm* no pertenece al reagrupamiento en curso de construcción. (etapa 1308)

25 **[0194]** Como opción, la función *agrDep1()* puede estar configurada para suprimir del agregado *De12u* las dependencias cuya fecha *TA2m* de primera aparición del valor de firma *Nm* en el juego *P2* es más reciente que la fecha *T12'*. (etapa 1310)

30 **[0195]** Las etapas 1304 a 1310 se vuelven a empezar excluyendo las dependencias de los agregados *De12u* ya constituidos hasta que la unión de los conjuntos *De12u* obtenidos sea idéntica al conjunto *D12*. Esto puede ser realizado excluyendo las dependencias del conjunto *De12u* que se acaba de construir (etapa 1312) y volviendo a empezar las etapas 1304 a 1312 hasta que el conjunto *D12* esté vacío (etapa 1314)

[0196] La función *agrDep1()* está configurada para realizar simétricamente las etapas 1302 a 1314 para el subconjunto *D21*, de tal manera que pueda obtener agregados de dependencias *De21u* cuya unión corresponde al subconjunto de dependencias *D21*.

35 **[0197]** Como opción, la función *agrDep1()* puede estar configurada para establecer para cada agregado *De12u* (respectivamente *De21u*) datos de agregados, en particular a partir de los atributos de dependencia *Fm*, de dependencia de documento *Gm*, de dependencia de elemento *Hm* asociados a los valores de firma *Nm* de este agregado. En este caso, la función *agrDep1()* puede estar configurada para establecer dichos datos de agregado a partir de los datos *V1m* o *V2m*, *W1m* o *W2m*, y/o *Z1m* o *Z2m*.

40 **[0198]** Por ejemplo, la función *agrDep1()* puede estar configurada para establecer para cada agregado *De12u* (respectivamente *De21u*) una fecha de primera integración de un elemento cualquiera del agregado, es decir más antigua de las fechas *TA1m* asociadas a los valores de firmas *Nm* del agregado en cuestión. La función *agrDep1()* puede, además o en sustitución, estar configurada para establecer para cada agregado:

- 45 - el número de valores de firmas *Nm* (por lo tanto de dependencias) contenidas en el agregado en cuestión,
- un valor medio de los índices de fiabilidad de las referencias de temporalidad, por ejemplo calculado a partir de los atributos de dependencia de documento *Gm*, y/o
- un valor indicativo del tamaño acumulado de los elementos de documentos cuyos valores de las firmas *Nm* están comprendidos en el agregado en cuestión, por ejemplo calculado a partir de los atributos de dependencia de elementos *Hm* de estas firmas.

50 **[0199]** Como opción, la función *agrDep1()* puede estar configurada para establecer un dato característico de la probabilidad del agregado. Por probabilidad del agregado, se entiende principalmente el hecho de que el conjunto de los elementos de documentos cuyos valores de firmas *Nm* están comprendidos en el agregado en cuestión proceden de un mismo proceso de integración de contenido. Por ejemplo, la función *agrDep1()* puede estar configurada para verificar para un agregado *De12u* (respectivamente *De21u*) si existe un intervalo de tiempo durante el cual el conjunto de los valores de firmas de este agregado coexistía en la huella del juego de documentos de origen.

60 **[0200]** Se destaca aquí que la adición de nuevos estados a uno de los juegos de documentos *P1* y *P2* no modifica las dependencias del conjunto *E"12* ya censadas siempre que estos estados sean posteriores a los estados a partir de los cuales se ha establecido el conjunto *E"12*. Además, la adición de nuevos estados no modifica los agregados obtenidos gracias a la función *agrDep1()*, pero puede solamente modificar el número de dependencias de estos agregados. En la práctica, cualquiera que sea la adición o la supresión de elementos de documentos en los juegos *P1* y *P2*, posteriores a la comparación, las dependencias establecidas siguen siendo válidas. Por ejemplo, la supresión en el juego *P2* de un elemento de documento del juego *P1*, y que por lo tanto ha generado una dependencia, no hace que la dependencia desaparezca. La comparación entre juegos de documentos, tal como lo permite la invención, puede ser calificada por ello de "robusta". Esta propiedad está

principalmente debida al hecho de que la huella temporalizada de un juego de documento, tal como ha sido establecida por el módulo SignM 110, no queda aplastada por la adición de estados ulteriores: pueden añadirse nuevos valores de firmas a la huella pero los datos de firma de los valores de firma existentes no se modifican, en particular en el caso en el cual se trata de una fecha de primera aparición. Dicho de otro modo, la huella de un juego de documento en una fecha considerada comprende la huella de este documento establecida en una fecha anterior.

[0201] Un segundo ejemplo de función de agregación $\text{agrDep2}()$ se describe con ayuda de la figura 14.

[0202] La función $\text{agrDep2}()$ está configurada para repartir el conjunto de las dependencias $E''12$ según la orientación de las dependencias en dos subconjuntos de dependencias $D12$ y $D21$ después de la etapa 1300 de la figura 13. (etapa 1400)

[0203] Para cada uno de los valores de fecha $TA1u$ de primera aparición en el juego de documentos $P1$ tal que existe un valor de firma Nu comprendido en el conjunto de dependencias $D12$, la función $\text{agrDep2}()$ está configurada para establecer un subconjunto de dependencias, o agregado, $De12u$ que comprende el conjunto de los valores de firmas Nm , y los datos asociados a estos valores de firma, cuya fecha $TA1m$ es idéntica a la fecha $TA1u$ (etapa 1402). Dicho de otro modo, la función $\text{agrDep2}()$ está configurada para repartir el conjunto $D12$ en de los agregados $De12u$ cuyas firmas tienen la misma fecha de aparición en el juego de documentos $P1$.

[0204] Como opción, la función $\text{agrDep2}()$ puede estar configurada para determinar el número de agregados $De12u$ comprendido en el conjunto $D12$, por ejemplo con ayuda de una función de recuento $\text{nbr}()$, y para verificar si dicho número es superior a un valor predeterminado VAL (etapa 1404). El valor VAL equivale al número de agregados que se estima razonable tratar. El valor VAL puede ser determinado considerando por un lado que un número elevado de agregados conlleva un tratamiento ulterior engorroso y por otro lado que un número de agregados reducido conlleva una pérdida de información que puede hacer la explotación ulterior sin objeto. Por explotación ulterior, se entiende en particular el tratamiento realizado por el módulo RightsM 150.

[0205] Si el test de la etapa 1404 es negativo, la función $\text{agrDep2}()$ se para. Si no, la función $\text{agrDep2}()$ está configurada para determinar un conjunto de intervalos temporales $\Delta T12v$, de manera descrita más adelante. (etapa 1406)

[0206] Para cada uno de los intervalos $\Delta T12v$ determinados en la etapa 1406, la función $\text{agrDep2}()$ está configurada para construir un sobre-conjunto de dependencias, o agregado, $De12v$ que comprende el conjunto de los agregados $De12u$ de modo que la fecha $TA1u$ esté comprendida en el intervalo $\Delta T12v$ en cuestión (etapa 1408). Dicho de otro modo, la función $\text{agrDep2}()$ está configurada para reunir las dependencias del conjunto $D12$ según su valor de firma Nm en función de la fecha de primera aparición de esta firma en el juego de documentos $P1$, y más concretamente según la pertenencia de estas fechas a unos intervalos temporales $\Delta T12v$. En determinados casos, la construcción de los agregados $De12v$ podrá tener en cuenta la fecha $TA2u$ en lugar de la fecha $TA1u$.

[0207] Los intervalos $\Delta T12v$ se determinan para dar cuenta de una coherencia en el proceso de producción de los elementos de documentos. La determinación de estos intervalos $\Delta T12v$ puede tener en cuenta uno o varios criterios.

[0208] Por ejemplo, la función $\text{agrDep2}()$ puede estar configurada para establecer intervalos $\Delta T12v$ que corresponden cada uno a un estado del juego de documento de destino PDm , aquí el juego de documentos $P2$. Las dependencias del conjunto $D12$ son entonces agrupadas en función de los estados del juego de documentos de destino. En este caso, la función $\text{agrDep2}()$ está configurada para definir los agregados $De12v$ en función de las fechas $TA2u$.

[0209] La función o la fecha $TA2u$ también puede estar configurada para determinar los intervalos $\Delta T12v$ de tal manera que cada uno de los intervalos corresponda a una sucesión de estados del juego de documentos cuyos valores del atributo de estado, en particular el identificador de contribuidor Aij , son idénticos. Las dependencias del conjunto $D12$ son entonces agrupadas en función de los estados del juego de documentos de origen y por contribuidores.

[0210] De manera general, la definición de los intervalos $\Delta T12v$ podría resultar de una elección de periodos temporales característicos de la frecuencia de actualización de los contenidos de documentos, por ejemplo extraídos de datos obtenidos mediante la herramienta de gestión de versión, por ejemplo las fechas, Tij .

[0211] Se entiende que la función $\text{agrDep2}()$ está configurada para realizar las etapas 1402 a 1408 de manera simétrica para el conjunto $D21$.

[0212] De manera análoga a la función $\text{agrDep1}()$, la función $\text{agrDep2}()$ puede estar configurada para verificar la probabilidad de los agregados $De12v$ y $De21v$ obtenidos.

[0213] De nuevo de manera análoga, la función $\text{agrDep2}()$ puede estar configurada para establecer para cada agregado $De12v$ unos datos de agregados, en particular a partir de los atributos de dependencia Fm , de dependencia de documento Gm , de dependencia de elemento Hm asociados a los valores de firmas Nm de este agregado, y en este caso a partir de los datos $V1m$ o $Y2m$, $W1m$ o $W2m$, y/o $Z1m$ o $Z2m$.

[0214] Por ejemplo, la función $\text{agrDep2}()$ puede estar configurada para establecer para cada agregado $De12v$ (respectivamente $De21v$) una fecha de primera integración de un elemento cualquiera del agregado, el número de valores de firmas Nm (por lo tanto de dependencias) contenidas en el agregado en cuestión, un valor medio de los índices de fiabilidad de las referencias de temporalidad, y/o el tamaño acumulado de los elementos de documentos cuyos valores de las firmas Nm están comprendidos en el agregado en cuestión.

[0215] Las funciones $\text{agr1Dep}()$ y $\text{agr2Dep}()$ descritas aquí no son limitativas y pueden definirse otras funciones de agregación, como complemento o en sustitución. Por ejemplo, los valores de firmas Nm , y los datos

asociados a estos valores, pueden ser reagrupados según sus datos F_m , G_m y H_m , y/o en este caso según unos datos $Y1m$ o $V2m$, $W1m$ o $W2m$, y/o $Z1m$ o $Z2m$.

[0216] En todos los modos de realización de las funciones de agregación, estas funciones pueden además estar dispuestas para establecer un dato equivalente a un peso para cada agregado obtenido. Esto permite establecer una jerarquización de los agregados de manera que el tratamiento ulterior por ejemplo puede ser facilitado. Por ejemplo, las funciones de agregación pueden estar dispuestas para tener en cuenta para la determinación de dicho peso, el número de dependencias comprendidas en el agregado, el tamaño acumulado de los elementos de documentos cuya firma representa una dependencia, un peso en el juego de documentos de origen y/o de destino, la fiabilidad de la o de las referencias temporales asociadas a las firmas del agregado, y cualquier información disponible.

Segundo modo de realización del módulo de comparación de juegos de documentos

[0217] En este modo de realización, el módulo ComM 140 está dispuesto para comparar en el sentido descrito más arriba un juego de documento P_i con varios juegos de documentos P_i .

[0218] El módulo ComM 140 está dispuesto para establecer una dependencia entre un juego de documentos P_i y unos juegos de documentos P_i , determinando la integración en el juego de documentos P_i de elementos de documentos de los juegos de documentos P_i , e inversamente la integración en los juegos de documentos P_i de elementos de documentos del juego de documentos P_i . Dicho de otro modo, el módulo ComM 140 está dispuesto para establecer dependencias entre un juego de documentos y un conjunto de juegos de documentos, y para orientar estas dependencias.

[0219] El módulo ComM 140 comprende una función de comparación de juegos de documentos comp1N(). La función comp1N() está configurada para comparar un primer juego de documentos P_1 con unos juegos de documentos $P_2, P_3, \dots, P_n, \dots, P_N$, siendo N un entero natural, para determinar los elementos de documento comunes a los juegos de documentos P_1 y P_n , para cualquier valor de n .

[0220] La función comp1N() se describe con ayuda de la figura 15.

[0221] La función comp1N() está configurada para extraer un subconjunto E^1 de la base PDB 130 que contiene los valores de firma $N1q$ relativos al juego de documentos P_1 así como los datos de firmas $V1q$ correspondientes (etapa 1500). Se supone aquí que existen Q_1 valores de firmas $N1q$ diferentes asociados al juego de documentos P_1 .

[0222] Como opción, la función comp1N() está configurada para extraer además el conjunto de los datos asociados a los valores de firmas $N1q$ en la base PDB 130. Por ejemplo, la función comp1N() puede estar configurada para extraer una tabla del tipo de la tabla 1102 de la figura 11.

[0223] La función comp1N() está configurada para extraer subconjuntos E^n de la base PDB 130 relativos a cada juego de documentos P_n (etapa 1502), y constituidos por Q_n valores de firmas Nq' diferentes. Por ejemplo, la función comp 1N() puede estar configurada para extraer tablas del tipo como la tabla 1104 de la figura 11, para cada juego de documentos P_n .

[0224] La función comp1N() está configurada para construir un conjunto de intersección E^1N constituido por la reunión de los subconjuntos de intersección del conjunto E^1 con cada uno de los conjuntos E^n . Por intersección entre dos conjuntos, se entiende el conjunto de los valores de firmas contenidos a la vez en estos dos conjuntos, y, en este caso, los datos asociados a estos valores de firma en el uno y el otro de los conjuntos (etapa 1504). Dicho de otro modo, la función comp1N() está configurada para establecer una tabla PintTb[] que agrupa los valores de firmas comunes a E^1 y a al menos uno de los conjuntos E^n . Para cada valor de firma común, la función comp1N () está configurada para almacenar en la tabla PintTb[] los datos de firmas asociadas a este valor de firma en el conjunto E^1 y en cada uno de los conjuntos E^n donde se encuentra. En este caso, la función comp1N() puede estar configurada para almacenar además determinados o todos los datos asociados a estos valores de firmas en la base PDB 130.

[0225] La función comp() está configurada para establecer un conjunto de dependencia $E''1N$. Por ejemplo, la función comp1N() puede estar configurada para construir una tabla de dependencias PDepTb[] en la que cada fila representa una dependencia orientada entre el juego de documentos P_1 y un juego de documentos P_n (etapa 1506).

[0226] La función comp1N() está configurada para establecer la tabla PdepTb[] de tal manera que cada línea de esta tabla comprenda:

- un valor de la firma Nm ,
- una referencia a un juego de documentos de origen POm para este valor de firma (es decir al juego P_1 o uno de los juegos P_n en particular),
- una referencia a un juego de documentos de destino PDM para este valor de firma (es decir el juego P_n particular si POm vale P_1 o el juego P_1 si POm vale el juego P_n particular);

[0227] La tabla PdepTb[] se rellena para censar cada valor de firma Nm del conjunto E^1N . Las referencias a los juegos de documentos de origen POm y de destino PDM se determinan con ayuda de una función de orientación dir() a partir de los datos de firmas $V1m$ y Vnm .

[0228] Cada fila de la tabla PdepTb[] representa una dependencia orientada que indica una referencia a un juego de documentos "de origen" POm y una referencia a un juego de documentos "de destino" PDM , para un valor de firma Nm .

[0229] Como opción, la tabla PdepTb[] puede almacenar además para cada valor de firma Nm :

- un atributo de dependencia Fm , que puede ser establecido a partir de los datos de firmas $V1m$ y Vnm , por ejemplo una fecha de primera aparición del valor de firma Nm en el juego de documentos de origen POm ,
- un atributo de dependencia de documento Gm , que puede ser establecido a partir de los datos de atributos de documentos $W1q$ y Wnq , por ejemplo un dato que caracteriza el tipo del documento del cual proviene el elemento generador del valor de firma, o un índice de fiabilidad de la referencia de temporalidad
- un atributo de dependencia de elemento Hm , que puede ser establecido a partir de los datos de atributos de elemento $Z1q$ y Znq , por ejemplo una indicación del tamaño del elemento generador del valor de firma Nm .

[0230] La determinación de los juegos de documentos de origen POm y de destino PDm puede tener en cuenta diferentes criterios. La función dir() puede estar configurada consecuentemente.

[0231] En particular, la función dir() puede estar configurada para determinar en calidad de juego de documento de origen POm el juego de documentos en el cual el elemento de documento generador del valor de firma Nm ha aparecido primero. En este caso, el juego POm es el juego de documentos en la huella del cual el valor de firma Nm ha aparecido antes, mientras que el juego PDm puede ser el juego de documentos en el cual este valor de firma Nm solamente ha aparecido más tarde.

[0232] Las diferentes variantes de la función comp() descrita más arriba pueden ser transpuestas a la función comp1N() puesto que se trata a cada vez de orientar una dependencia entre dos juegos de documentos, a saber el juego $P1$ y un juego Pn .

[0233] La función comp1N() está por lo tanto configurada para establecer un conjunto $E"1N$ de dependencias entre el juego de documentos $P1$ y uno cualquiera de los juegos de documentos Pn . Dicho de otro modo, cada dependencia (o valor de firma Nm) del conjunto $E"1N$ caracteriza la presencia de un elemento de documento del juego $P1$ en un juego de documentos Pn , con un juego de documento de origen POm y un juego de documento de destino PDm .

[0234] Este modo de realización puede ser visto como la aplicación de la función comp() del primer modo de realización del módulo de comparación entre el juego de documento $P1$ y cada uno de los juegos de documentos $P2, \dots, Pn, \dots, PN$.

[0235] En este modo de realización, la función comp1N() puede llamar una función de agregación para reunir las dependencias en agregados, o subconjuntos del conjunto $E"1N$.

[0236] En un primer ejemplo, la función comp1N() está configurada para llamar una función de agregación agrDep3(). La función agrDep3() está configurada para repartir el conjunto de dependencias $E"1N$ en subconjuntos $E"1n$ de tal manera que cada uno de los sub -conjuntos $E"1n$ reúne las dependencias entre el juego de documentos $P1$ y el juego de documento Pn en cuestión. La función agrDep3() está además dispuesta para llamar a una de las funciones de agregación del primer modo de realización del módulo de comparación, en particular una de entre las funciones agrDep1() y agrDep2(), para cada uno de los conjuntos de dependencias $E"1n$.

[0237] En un segundo ejemplo, la función comp1N() está configurada para llamar una función de agregación agrDep4(). La función agrDep4() está configurada para suprimir del conjunto de dependencias $E"1N$, para un valor de firma Nm , particular, las dependencias entre el juego $P1$ y uno de los juegos Pn , para las cuales la fecha de aparición del valor de firma Nm en el juego de origen POm no es la más antigua de todas las fechas de aparición de este valor de firma Nm en el conjunto de los juegos de documentos $P1, P2, \dots, Pn, \dots, PN$.

[0238] Como opción, en particular cuando el número de dependencias es demasiado elevado, por ejemplo para ser convenientemente tratado ulteriormente, la función agrDep4() puede estar configurada para realizar agregados en función del juego de documentos de origen POm . La función agrDep4() puede también estar configurada para realizar agregados en función de los estados del juego de documentos de origen. La función agrDep4() puede también estar configurada para realizar agregados en función de los criterios siguientes:

- estado del juego de documentos de destino,
- identificador del contribuidor, y/o
- intervalos temporales, de manera análoga a lo que ha sido descrito más arriba en el primer modo de realización del módulo de comparación.

[0239] Igual que antes, unos datos característicos pueden ser atribuidos a los agregados.

Tercer modo de realización del módulo de comparación

[0240] En este modo de realización, el módulo ComM 140 está dispuesto para comparar varios juegos de documentos Pi entre sí. El módulo ComM 140 comprende una función de comparación compNN() dispuesta para llamar sucesivamente la función comp1N() para cada uno de los juegos de documentos Pi para establecer un conjunto de dependencias $E"NN$.

[0241] En un desarrollo particular, la función compNN() está configurada para suprimir en el conjunto de dependencias $E"NN$, para un valor de firma Nm particular, las dependencias entre dos de los juegos Pn para las cuales la fecha de aparición del valor de firma Nm en el juego de origen POm no es la más antigua de todas las fechas de aparición de este valor de firma Nm en el conjunto de los juegos de documentos $P1, \dots, Pn, \dots, PN$.

[0242] Las dependencias pueden a continuación ser agregadas según una u otra de las funciones descritas más arriba, ya sea de manera combinada, ya sea en sustitución. En particular, las dependencias pueden ser agregadas según el juego de documentos de destino *PDm*, es decir que se reúnen los valores de firma *Nm*, y los datos que están asociados a estos datos, cuyo juego de documentos de destino *PDm* es idéntico.

[0243] Como anteriormente, unos datos característicos pueden ser asociados a los agregados así obtenidos, tales como el tamaño acumulado de los elementos de documentos, por ejemplo obtenido por adición de los atributos de elementos *AEijkl* cuando estos comprendan una indicación del tamaño del elemento, o también una media, eventualmente ponderada según los índices de pertinencia *Bijkl* asociados a los elementos de documentos, cuyas firmas están presentes en el agregado en cuestión, la más antigua de las fechas de primera aparición en el juego de destino, u otro.

Módulo de gestión de los derechos

[0244] El módulo de gestión de los derechos RightsM 150 está configurado para interactuar con el módulo ComM 140 y la base PDB 130.

[0245] El módulo RightsM 150 comprende una función de listado *list()* configurada para establecer para un juego de documentos *Pi* el conjunto de las dependencias, es decir valores de la firmas *Nm* y de los datos que están asociados a estos valores de firmas, cuyo juego de documentos de destino *PDm* es el juego de documentos *Pi*. En este caso, cuando los conjuntos de dependencias *E"12*, *E"1N* o *E"NN* han sido objeto de una agregación (es decir cuando se han realizado unos subconjuntos de estos conjuntos de dependencias), la función *list()* está configurada para establecer la lista de estos agregados de los cuales un atributo asociado es el juego de documentos de destino *PDm* y de los cuales este atributo vale *Pi*.

[0246] En un modo de realización particular, la función *list()* está configurada para establecer para cada uno de los valores de firmas *Nim* presentes en el conjunto de dependencias *E"NN*:

- el conjunto de las referencias a los juegos de documentos de origen *POm* (designado a continuación *Pi'*),
- el conjunto de los atributos de elementos *AEi'.jqp,lqp,kqp* asociados a cada uno de los juegos de documentos de origen *Pi'*,
- el conjunto de los atributos de documento *AOi'.jqp,lqp* asociados a cada uno de los juegos de documentos de origen *Pi'*, por ejemplo el conjunto de los identificadores de contribuidores *Ai'.jqp*,
- el conjunto de los atributos de estados *ASi'.iqp* asociados a cada uno de los juegos de documentos de origen *Pi'*, y/o
- el conjunto de las referencias de temporalidad de los elementos de documentos, por ejemplo el conjunto de las fechas *Ti'.jqp*.

[0247] En un desarrollo particular, la función *list()* está configurada para establecer además:

- el conjunto de los atributos de dependencias *Fm*, de dependencia de documento *Gm* y/o de dependencia de elemento *Hm*, y/o
- el conjunto de los atributos asociados a cada uno de los agregados de dependencias, por ejemplo un atributo de dependencia indicativo de una licencia.

[0248] Por ejemplo, el atributo de dependencia *Fm* puede indicar un tipo de licencia al cual está sometido el juego de documentos de origen *POm*.

[0249] En un desarrollo particular, los atributos de elementos *AEi'.jqp,lqp,kqp* comprenden una información identificadora de un tipo de licencia (por ejemplo, "licencia GPL", o licencia "BSD"). La función *list()* está entonces configurada para establecer un conjunto de las licencias de los juegos de documentos de origen *Pi'*. Ello permite, por ejemplo, establecer una conexión entre una dependencia y una licencia para ayudar a determinar si las modalidades de explotación del juego de documentos *Pi* es compatible con las del o de los juegos *Pi'*.

[0250] En otro modo de realización particular, la función de *list()* está configurada para establecer también una lista de las dependencias (o agregados de dependencias) cuyo juego de documentos de origen *POm* es *Pi*, y un atributo de cada juego de documentos de destino *PDm* (aquí *Pi'*) indicador de un tipo de licencia.

[0251] En aún otro modo de realización, la función *list()* puede estar configurada para llamar una función de verificación de incompatibilidad de licencias *licver()*, dispuesta para comparar un tipo de licencia del juego de documentos de origen *POm* almacenado en calidad de atributo de juego de documentos y un tipo de licencia del juego de documentos de destino *PDm*, también almacenado en calidad de atributo de juego de documentos, a una lista preestablecida de incompatibilidades jurídicas de licencias. La función *list()* puede estar configurada para llamar la función *liever()* para cada dependencia del conjunto *E"NN*.

[0252] En determinados casos, los valores de firmas *Nim* y *Ni'm*, así como los datos que tiene asociados, podrían también ser establecidos por la función *list()*. En el caso de la utilización de una herramienta de gestión de versión, la cual está configurada para conservar un histórico de las diferentes modificaciones aportadas a un juego de documentos, aquí al juego de documento de origen y al juego de documentos de destino, esto puede permitir ayudar al usuario a determinar la titularidad de los derechos relativos a los diferentes juegos de documentos.

[0253] En un modo de realización particular, la función *list()* está configurada para establecer para un juego de documentos *Pi*, los valores de firmas *Nm* encontrados en las dependencias cuyo juego de documentos de

destino P_{Dm} es este juego de documentos P_i , y los valores de firmas ausentes de cualquier conjunto de dependencias. Cada valor de firma está asociado a al menos un dato de firma V_{iq} o $V_{i'q'}$, y a un dato de documento W_{iq} o $W_{i'q'}$, según el caso. Por ejemplo, el dato de firma V_{iq} o $V_{i'q'}$ puede comprender una fecha de primera aparición (independientemente del juego de documentos), y el dato de documento W_{iq} o $W_{i'q'}$, un

5

identificador del documento donde el contenido que haya generado este valor de firmas ha sido encontrado por primera vez.
[0254] Por otro lado, la función list() puede estar configurada para establecer el conjunto, o un extracto, de los atributos de elementos $AEijkl$, de documentos $AOijl$, o también de estados $ASij$, asociados en la base PDB 130, o WDB 120, al juego de documentos P_i .

10

[0255] En un modo de realización particular, por ejemplo, los atributos de documento $AO_{i,jqp,lqp}$ comprenden cada uno al menos un identificador de contribuidor así como una indicación de un estatuto de este contribuidor (por ejemplo, "empleado", "particular", o "buscador"). y la función list() está entonces configurada para establecer una lista de los identificadores y de su estatuto.

15 Desarrollos

[0256] La invención halla su máximo interés en el caso en el cual cada documento está recortado en elementos, a cada uno de los cuales está asociada una firma. Sin embargo, determinados de los documentos pueden no estar recortados en elementos, por ejemplo en el caso en que el recorte no tiene sentido, o no hay interés en el objetivo buscado. En tal caso, se puede asociar una firma al documento tomado en su totalidad.

20

[0257] Debido a lo que precede, el algoritmo de recorte puede ser propio de cada sistema de gestión informática de un juego de documentos. Puede ser interesante que los algoritmos de recorte estén estandarizados, en función de cada naturaleza de archivo y/u objetivo buscado. Así, unas comparaciones se vuelven posibles directamente, de un sistema a otro.

25

[0258] La descripción anterior recurre a unas fechas T. En determinados casos, unos identificadores de fecha serán suficientes. Por "identificador de fecha" se entiende aquí cualquier parámetro susceptible de ser objeto de una selección conforme al orden cronológico. Así, un identificador de fecha puede ser simplemente el identificador de versión en el juego de documentos.

30

[0259] Por otro lado, no siempre es necesario almacenar todas las firmas. Se ha visto que se puede minimizar la importancia de determinadas firmas almacenadas en una lista que se podría llamar "lista gris". Es concebible hacer una "lista negra" de firmas que no serán almacenadas, eventualmente bajo condición.

Aplicaciones

35

[0260] La descripción detallada de más arriba se sitúa esencialmente en el marco de proyectos de desarrollo de software. Los medios descritos suministran herramientas automatizables que permiten en especial definir el estatuto jurídico de los softwares. Se sabe que es una tarea especialmente difícil, a partir del momento en que se aumenta el número de los contribuidores personas físicas, y de sus entidades jurídicas de asociación, aún más debido a que esta asociación puede haber evolucionado en función del tiempo. La tarea es aún más difícil cuando se trata de softwares a base de componentes y/o desarrollados en colaboración.

40

[0261] Frente a esta complejidad de la tarea, es corriente definir, de manera puramente intelectual, soluciones aproximadas pero inexactas, como, por ejemplo:

45

- una calificación arbitraria de obra colectiva, simplemente porque no se ha sabido resolver exactamente las contribuciones de cada una y las devoluciones de derechos correspondientes,
- negar sin justificación contribuciones consideradas menores, de cara a la personalidad de un autor influyente.

50

[0262] La invención permite al contrario utilizar técnicas rigurosas, exactas y fiables, con el grado de granularidad que se desea en el recorte de cada documento en elementos. Esto constituye un elemento de base clave, en la evaluación y la gestión del riesgo jurídico en el marco de la utilización o de la explotación de software, y en particular de software difundido bajo licencia "código abierto".

55

[0263] Pero la invención es susceptible de otras aplicaciones, al nivel de la cadena de producción de las obras numéricas, y/o de las herramientas de numeración y de trazabilidad que utiliza o a las cuales recurre. En este campo, se hallan también obras en colaboración y/o a base de componentes reutilizables.

60

[0264] Más generalmente, la invención es especialmente interesante a partir del momento en que hace posible descomponer los documentos en elementos característicos de un título y a partir de criterios cualesquiera (derecho de autor o copyright para una obra), de manera que la comparación de los elementos provenientes de la descomposición de dos estados sucesivos sea característica de las modificaciones realizadas el estado más antiguo para producir una versión en curso o final.

60

[0265] Para el software (código), el recorte utiliza la sintaxis y la gramática del lenguaje. En cambio, para un video/ animación, el recorte dependerá del formato de almacenamiento; un formato como SVG (para "Scalable Vector Graphics") está especialmente bien adaptado para la definición de un recorte a la vez eficaz y significativo en términos de contenido.

65

[0266] La búsqueda de similitud entre documentos se hace en base a la presencia o no de los elementos provenientes de la descomposición.

1. Contrariamente al estado de la técnica, la temporalidad de la huella permite en función de la fiabilidad de la referencia temporal determinar cuando dos softwares presentan similitudes, cual ha sido integrado en total o en parte por el otro, y en que versión de este. Por ejemplo, de este modo es posible determinar:

- 5 (1) el estatuto jurídico de una contribución externa debidamente identificada sin restringirse a la cuestión de la compatibilidad de licencia,
 (2) de qué etapa de la cadena de producción proviene la parte del software integrada,
 (3) la " difusión " de un código de fuente abierta es decir su " notoriedad " para identificar los proyectos que lo utilizan.

10 2. Permite hacer búsquedas de códigos exógenos sin tener que acceder directamente a los códigos fuentes.

3. El procedimiento es robusto desde el punto de vista de la herramienta de producción en el sentido en que una actualizada incremental del histórico de las contribuciones (la adición de estado posteriormente al último estado conocido) solamente podrá extender la firma temporal añadiendo en este caso nuevas firmas y por lo tanto nuevas dependencias (atómicas y agregados) posibles, pero sin suprimir las anteriores (hay que destacar sin embargo que su pertinencia puede verse afectada). Las dependencias pueden ser modificadas al tener en cuenta nuevos juegos de documentos para la comparación. Se trata de una propiedad clave desde el punto de vista de la problemática de trazabilidad (es decir desde el punto de vista de los procesos de control calidad)

20 4. La aplicación de filtros y algoritmos de descomposición de contenido por tipo de archivo (para cada lenguaje en especial) permite garantizar la perennidad de los datos por ejemplo al ser insensibles a los transportes invisibles de un sistema a otro, y/o de un entorno de desarrollo a otro. También es posible definir filtros o funciones de recorte robustas a las estrategias de rodeo (por ejemplo filtro insensible a un cambio de nombre de variable o a una permutación de bloques que no cambian las funcionalidades).

25 5. El enlace entre " herramientas de producción del contenido ", generación de huella temporalizada y búsqueda de dependencias, permite, para definir el estatuto jurídico de un software, tener en cuenta el conjunto de los elementos constitutivos de este (lista de los autores y de su estatuto), incluidos los elementos de terceros entre los cuales se cuentan las modalidades de explotación.

30 6. Permite también definir estrategias para evitar problemas de compatibilidad con licencias (por ejemplo cuando unas versiones anteriores o posteriores están bajo licencias más permisivas), cuando se trata de hacer medidas de impactos en la integración de código exógeno contaminante, o también identificar las responsabilidades (quien ha integrado código exógeno, y cuando ?).

35 7. Permite también de caracterizar las actividades de tal o tal contribuidor en término de "contenido ". Por ejemplo, el número de huellas atómicas generadas permite definir una nueva métrica, o ponderar una métrica a de la actividad en el sentido de la herramienta de producción (es decir el número de actualizaciones o el número de archivos modificados). En este caso, es posible tener en cuenta también la pertinencia de las huellas (por ejemplo para proponer partes de copropiedad de manera más pertinente y robusta que el simple número de líneas generadas).

40 **[0267]** Lo que precede concierne sobre todo al software, pero puede aplicarse a otros tipos de obras, y más generalmente a otros tipos de documentos informáticos en los que se desea hacer un análisis fino acoplado a la temporalidad.

[0268] De manera muy general, se propone aquí un procedimiento de gestión informática de documentos, que comprende en combinación las etapas siguientes:

50 a/ considerar un primer ($j = 1$) contenido de documento (O_{ijl}), que tiene una primera referencia de temporalidad (TP_{ijl}),

b/ descomponer (200) este primer contenido de documento en elementos (C_{ijkl}), según una regla escogida,

55 c/ asociar (202) a cada elemento de documento (C_{ijkl}) obtenido en la etapa b/ una firma de su contenido (N_{ijkl}),

d/ almacenar para el primer contenido de documento una huella (E_{ijl}), que comprende determinadas al menos de las firmas obtenidas en la etapa b/, así como un identificador de dicha primera referencia de temporalidad de este primer contenido de documento,

60 e/ repetir las etapas a/ a d/ a cada vez ($j = j + 1$) para al menos otro contenido de documento, que tiene su referencia de temporalidad,

f/ considerar un primer valor de firma (N_{ijkl}) como valor de trabajo (N_{iq}),

65

g/ aplicar una función de datación (dat()) a los identificadores de referencia de temporalidad (*TPijl*) de las huellas (*Eijl*) que contiene este valor de trabajo, y asociar a este valor de trabajo (*Niq*), el valor resultado (*Viq*) de dicha función de datación, en calidad de referencia de temporalidad unificada,

5 h/ repetir las etapas f/ a h/ con cada vez otro valor de firma (*Nijlk*) como valor de trabajo (*Niq*),

i/ almacenar una correspondencia entre cada valor de firma y su referencia de temporalidad unificada.

10 **[0269]** En principio, los contenidos de documentos (*Oijl*) son identificables (o "clasificables") según al menos dos categorías de entre: un documento determinado, un estado de documento, una versión de documento, y un juego de documentos susceptible de comprender varias versiones de documentos. Consecuentemente, en la descripción detallada de más arriba, la expresión "contenido de documento" puede tener como objetivo indiferentemente un documento preciso (*l*), un estado de documento (*j*), una versión de documento, o un juego

15 de documento (*i*), excepto donde el contexto requiera otra cosa.

[0270] En la descripción detallada que precede, las cuatro categorías están presentes. En principio, hasta el nivel del juego de documento, se conserva la misma finalidad: es la evolución de este que se percibe como una misma cosa. En cambio, dos juegos de documentos diferentes tienen como objetivos dos cosas diferentes, al menos parcialmente.

20 **[0271]** En un modo de realización interesante, en especial para la búsqueda de contribuciones potencialmente creativas:

- la función de datación (dat()) está configurada para establecer en calidad de dato de firma (*Viq*) al menos uno de entre el identificador de la referencia de temporalidad (*TPi,jqp,lqp*) más antiguo de la lista y el identificador de la referencia de temporalidad (*TPi,jqp,lqp*) más reciente de la lista.

25

[0272] Puede ser interesante particularizar determinados al menos de los elementos de documento, por ejemplo para modular su importancia. A tal efecto:

30 - la etapa c/ comprende el hecho de asociar además a cada elemento de documento (*Cijlk*) un atributo de elemento (*AEijlk*) relativo a este elemento de documento,

- en la etapa d/, la huella (*Eijl*) comprende además un identificador de determinados al menos de los atributos de elemento (*AEijlk*),

35 - en la etapa g/, la función de datación (dat()) está configurada para tener en cuenta los identificadores de atributos de elementos (*AEijlk*), para la elección de temporalidad, en particular si verifican una condición.

[0273] En realizaciones particulares, combinables:

- en la etapa d/, la huella (*Eijl*) comprende las firmas de los elementos cuyo atributo de elemento (*AEijlk*) es superior a un valor determinado.

40 - en la etapa c/, el atributo de elemento depende de datos de tipo estadístico relativos al contenido del elemento de documento.

[0274] También, puede ser interesante particularizar determinados al menos de los propios documentos. A tal efecto:

45 - en la etapa a/, el contenido de documento (*Oijl*) tiene además un atributo de documento (*AOijl*),

- en la etapa d/, la huella (*Eijl*) comprende además un identificador de dicho atributo de documento del primer contenido de documento,

50 - la etapa g/ comprende además el hecho de aplicar una función de autenticación (Aut()) a los identificadores de atributos de documento (*AOijl*) de las huellas (*Eijl*) que contienen el valor de trabajo, y de asociar a este valor de trabajo (*Niq*), el valor-resultado (*Wiq*) de dicha función de autenticación, en calidad de dato de atributo de documento unificado,

- la etapa i/ comprende además el hecho de almacenar una correspondencia entre cada valor de firma y su dato de atributo de documento unificado.

55 **[0275]** En una realización particular de lo que precede:

- en la etapa g/, la función de autenticación se aplica además a los identificadores de referencias de temporalidad de las huellas (*Eijl*) que contienen el valor de trabajo.

60 **[0276]** Según otras realizaciones particulares, combinables, y de aplicación más general:

- el atributo de contenido de documento (*AOijl*) comprende un identificador de un contribuidor a dicho contenido de documento (*Oijl*).

65 - está previsto (por ejemplo en la etapa d/) reconocer al menos una estructura de contenido de documento (*Oijl*) conocida y, en presencia de un documento cuya estructura ha sido reconocida, asociar a este documento un dato (702) característico de esta estructura, lo cual interviene en el establecimiento de la huella (*Eijl*).

- está previsto (por ejemplo en la etapa a/) establecer a partir de un contenido de documento (O_{ijl}) un contenido de documento depurado (O'_{ijl}) según un primer conjunto de reglas predeterminadas, las cuales dependen de dicho dato característico de la estructura del contenido del documento (O_{ijl}), mientras que el resto de las operaciones b/ a d/ se realizan a partir de este contenido de documento depurado (O'_{ijl}).

5 - está previsto (por ejemplo en la etapa b/) establecer a partir de un elemento de documento (C_{ijkl}) un elemento de documento depurado (C'_{ijkl}) según un segundo conjunto de reglas predeterminadas, en particular dependiendo del dato característico de la estructura del documento (O_{ijl}), mientras que el resto de las operaciones b/ a d/ se efectúan a partir de los elementos de documento depurados (C'_{ijkl}) así obtenidos.

10 **[0277]** Se pueden obtener así varias huellas, correspondientes a varios contenidos de documento.

[0278] Para acercarlos, el procedimiento puede completarse tal como sigue:

- para un valor de firma determinado (N_{iq}), se determina una lista de los identificadores de referencia de temporalidad ($T_{Pi, jqp, lqp}$) de las huellas ($E_{i, jqp, lqp}$) que contienen este valor de firma (N_{iq}), y

15 - se establece una referencia de temporalidad unificada (V_{iq}) para este valor de firma (N_{iq}), buscando el identificador de la referencia de temporalidad ($T_{Pi, jqp, lqp}$) más antiguo de la lista (o aplicando otro criterio temporal, tal como la búsqueda de la referencia de temporalidad ($T_{Pi, jqp, lqp}$) más reciente de la lista).

20 **[0279]** Entonces se puede, para la firma determinada, completar (o reemplazar) su referencia de temporalidad inicial por la referencia de temporalidad unificada (V_{iq}).

[0280] Por su lado, cada contenido de documento está provisto de una lista de identificadores de referencias de temporalidad (TP_{ijl}), que se pueden comparar con una lista global que comprende el conjunto de las referencias de temporalidad (TP_{ijl}) de los contenidos de documentos (O_{ijl}). Se extrae, para un contenido de documento, uno o varios periodos temporales. Este periodo temporal comprende referencias de temporalidades sucesivas con respecto al conjunto de las referencias de temporalidad (TP_{ijl}). Representa el periodo durante el cual un elemento de documento que tiene una firma determinada ha estado presente de manera continua en un contenido de documento.

[0281] Las referencias de temporalidad pueden ser definidas de diferentes maneras, incluso por captura.

30 **[0282]** Se prefiere utilizar una herramienta de gestión de versiones de contenidos de documentos, y recibir en memoria, desde esta herramienta de gestión de versiones, referencias de temporalidad (TP_{ijl}) para unos contenidos de documentos.

[0283] Asimismo, los identificadores de contribuidor pueden ser definidos de diferentes maneras, incluso por captura. También se prefiere aquí utilizar una herramienta de gestión de versiones, y recibir en memoria, desde esta herramienta de gestión de versiones, al menos un identificador de contribuidor para cada uno de los contenidos de documentos (O_{ijl}).

35 **[0284]** Obviamente, la herramienta de gestión de versión también indicará versiones diferentes, que presentan contenidos diferentes, para lo que se considera globalmente como un único documento, en diferentes fases de su elaboración o de su vida. También puede indicar sub-versiones o "estados". A partir de una pluralidad de contenidos de documentos (O_{ijl}), la herramienta de gestión de versiones puede establecer el subconjunto de contenidos de documentos (O_{ijl}) que son diferentes entre sí. En este caso, es suficiente con aplicar las operaciones a/ a c/ una sola vez, sobre la base de dicho subconjunto de contenidos de documentos (O_{ijl}). Todas las firmas útiles se calculan entonces antes de la operación d/.

40 **[0285]** Se ha visto que el procedimiento prevé almacenar la correspondencia entre cada valor de firma y su referencia de temporalidad. Puede ser interesante almacenar además una conexión, por ejemplo un identificador, hacia el conjunto de los contenidos de documentos (O_{ijl}) de los que provienen el valor de firma y la referencia de temporalidad. Se dispone así, para una firma de temporalidad determinada, del conjunto de los contenidos de documentos que contienen esta firma.

50 **[0286]** El acercamiento entre contenidos de documentos hará aparecer partes comunes. Tienen en principio un mismo origen, que se puede buscar mediante el examen de las dependencias. Para ello:

- α se compara un primer conjunto de huellas ($E'1$) y un segundo conjunto de huellas ($E'2$), que tienen respectivamente un primer identificador (1) y un segundo identificador (2). El primer conjunto de huellas ($E'1$) comprende valores de firmas ($N1q$) y sus referencias de temporalidad unificada ($V1q$) correspondientes al primer identificador (1). Y el segundo conjunto de huellas ($E'2$) que comprende valores de firmas ($N2q$) y sus referencias de temporalidad unificada ($V2q$) correspondiente a dicho segundo identificador (2).

55 - β el conjunto de las comparaciones logradas proporciona un subconjunto de dependencias ($E''12$). Para cada valor de firma ($N1q; N2q'$) que es común, este subconjunto contiene referencias de temporalidades unificadas ($V1q; V2q'$) correspondientes para dichos conjuntos de huellas primero y segundo.

60 - γ se define (o predefine) un sentido u orientación de cada dependencia. Lo más corriente es colocar el origen o "fuente" en el momento más antiguo, y el destino o "diana" (PDm) en los tiempos ulteriores. En general, esto implica un conjunto de reglas predeterminadas, aplicadas sobre la base de las referencias de temporalidad unificadas ($V1q; V2q'$). Considerando el primer identificador (1) y el segundo identificador (2), cuyas huellas tienen un valor de firma común ($N1q$), esta operación de orientación permite calificar uno de entre los identificadores primero y segundo (1;2) en calidad de identificador de origen o "fuente" (POm) y el otro identificador (2;1) en calidad de identificador de destino o "diana" (PDm).

65

[0287] Un contenido de documento puede ser descompuesto en pocos elementos o muchos elementos, según el contexto, y según la granularidad escogida para esta descomposición. Con una granularidad fina, por lo tanto muchos elementos (de documento), el subconjunto (E^{12}) de dependencias (orientadas o no) también comprenderá muchos elementos. Es entonces interesante buscar ahí los elementos que están en la misma situación, lo cual se puede llamar un "agregado". Se puede repartir el subconjunto (E^{12}) según la orientación de las dependencias y luego crear respectivamente agregados de dependencias (De^{12u} ; De^{21u}), sobre la base de la aplicación iterativa de al menos uno de los criterios de partición siguientes:

- tomar la firma (Nm) común más antigua (1304) aún no agrupada, y agrupar con ella las otras firmas (Nm) comunes que provienen de referencias de temporalidad lado "fuente" que son anteriores a esta firma (Nm) común más antigua (1306),
- extraer, de un grupo determinado, una firma común cuya referencia de temporalidad "diana" es posterior a la firma más antigua de entre las firmas comunes aún no agrupadas (1310),
- efectuándose la iteración hasta la verificación de un criterio de fin de bucle determinado.

[0288] Se puede hacer entonces una comparación multi-documentos, sobre N conjuntos de huellas (N contenidos de documentos). En este caso:

- se hacen todas las comparaciones posibles 1 a 1 sobre los N documentos (o sobre determinados de entre estos). Hay en total $N(N-1)$ comparaciones posibles. Esto proporciona una pluralidad de subconjuntos de dependencias, cuya reunión es un conjunto de dependencias.
- se orientan las dependencias, tal como se ha descrito como anteriormente.
- para cada firma en estos subconjuntos de dependencia (o bien en el conjunto de dependencias), se separan las dependencias cuya referencia de temporalidad "diana", para la firma considerada, es posterior a la fecha más antigua de entre los N conjuntos de huellas, para la firma considerada.

[0289] Esta comparación puede hacerse a un nivel superior, por ejemplo entre N juegos de documentos, tales como se han definido más arriba.

[0290] En una aplicación como la gestión de los derechos, el procedimiento puede comprender la etapa siguiente:

- establecer una lista fechada de contribuidores, verificados a partir de las huellas y de las eventuales dependencias.

[0291] La invención cubre también los programas ejecutables, o programas producto, capaces de animar el dispositivo informático descrito, y/o de llevar a cabo los procesos correspondientes. Un tal programa ejecutable puede ser puesto a disposición bajo cualquier "medio" (soporte) legible por ordenador. La expresión "medio legible por ordenador" comprende los soportes de almacenamiento de datos, magnéticos, ópticos y/o electrónicos, así como un soporte o vehículo de transmisión, como una señal analógica o numérica.

[0292] La invención no se limita a los modos de realización descritos más arriba, sino que engloba todas las variantes que podría concebir el experto en la materia.

REIVINDICACIONES

1. Dispositivo informático de gestión de documentos, que comprende una memoria para almacenar contenidos de documentos, que tienen referencias de temporalidad, **caracterizado por el hecho de que** comprende:

- un extractor (200) capaz de descomponer un contenido de documentos (Oijl) en elementos (Ci/lk), según una regla escogida,

- un generador de firma (202), capaz de tratar un elemento de documento (Cijlk) para devolver una firma numérica característica de su contenido (Nijlk), estando dicho dispositivo **caracterizado por el hecho de que** comprende además:

- un generador de huella (signO()), capaz de establecer, para un contenido de documento (Oijl) una huella (Eijl), que comprende la referencia de temporalidad (TPijl) de este contenido de documento, así como determinadas al menos de las firmas (Ni/lk) obtenidas llamando al extractor (200) para transformar este contenido de documento en elementos (Cijlk), luego el generador de firmas (202) para los elementos (Cijlk) así obtenidos,

- un módulo marcador de hora (dat()) capaz de calcular el valor-resultado de una función predeterminada de elección de temporalidad en una lista de identificadores de referencias de temporalidad (TPijl), dicha función predeterminada de elección de temporalidad, calculada por el módulo marcador de hora, está configurada para determinar en calidad de valor resultado al menos uno de entre el identificador de la referencia de temporalidad más antigua de la lista y el identificador de la referencia de temporalidad más reciente de la lista.

- un supervisor (110) dispuesto para: * hacer funcionar el generador de huella (signO()) sobre al menos dos contenidos de documentos (Oijl), lo cual proporciona dos huellas (Eijl), que tienen conjuntamente una pluralidad de firmas, * para cada valor de firma (Niq), llamar repetitivamente el módulo marcador de hora (dat()), con a cada vez una lista de determinados al menos de los identificadores de referencia de temporalidad (TPijgp,lgp) de las huellas (Ei,jgp,lgp) que contienen un mismo valor de firma (Niq), lo cual proporciona una referencia de temporalidad unificada (Viq) para cada valor de firma (Niq), * asociar a un juego de contenidos de documento que comprende al menos dichos dos contenidos de documentos una huella numérica que comprende cada valor de firma de dicha pluralidad de firmas y que indica, para dicho valor de firma, la referencia de temporalidad unificada suministrada; almacenándose una correspondencia entre cada firma y el elemento de documento a partir del cual dicha firma ha sido generada.

2. Dispositivo según la reivindicación 1, **caracterizado por el hecho de que:**

- comprende además un calificador de contenido (eval()) capaz de tratar un elemento de documento para devolver un atributo de elemento relativo a este elemento de documento.

3. Dispositivo según la reivindicación 2, **caracterizado por el hecho de que:**

- el generador de huella (signO()) es capaz de establecer, para un contenido de documento, una huella que comprende además identificadores de determinados al menos de los atributos de elementos de este contenido de documento, llamando al calificador de contenido (eval()) para los elementos obtenidos del extractor (200),

- dicha función predeterminada de elección de temporalidad está configurada para tener en cuenta los identificadores de atributos de elementos, para la elección de temporalidad,

- el supervisor (110) está dispuesto para llamar repetitivamente el módulo marcador de hora (dat()), con a cada vez una lista de los identificadores de referencia de temporalidad de las huellas que contienen un mismo valor de firma y los identificadores de atributos de elementos asociados.

4. Dispositivo según cualquiera de las reivindicaciones 2 y 3, **caracterizado por el hecho de que:**

- el generador de huella está dispuesto para establecer, para un contenido de documento, una huella que comprende la referencia de temporalidad de este contenido de documento, y las firmas de los elementos de documento cuyo atributo de elemento verifica una condición determinada.

5. Dispositivo según cualquiera de las reivindicación 2 a 4, **caracterizado por el hecho de que** el calificador de contenido (eval()) es capaz de tratar un elemento de documento para devolver en calidad de atributo de elemento el valor-resultado de una función predeterminada de tipo estadístico que se refiere al contenido de este elemento de documento.

6. Dispositivo según cualquiera de las reivindicaciones anteriores, determinados al menos de los documentos que tienen además cada uno al menos un atributo de contenido de documento, **caracterizado por el hecho de que:**

- el generador de huella (signO()) es capaz de establecer, para un contenido de documento, una huella que comprende además un identificador del atributo de este contenido de documento.

7. Dispositivo según la reivindicación 6, **caracterizado por el hecho de que** comprende además:

- un módulo de autenticación (aut()) capaz de calcular el valor-resultado de una función predeterminada de elección de atributo de documento, en una lista de identificadores de atributos de contenidos de documentos, en función de una lista de identificadores de referencias de temporalidad.

5 **8.** Dispositivo según la reivindicación 7, **caracterizado por el hecho de que:**

- el supervisor (110) está dispuesto para:

* llamar repetitivamente el módulo de autenticación (aut()), con a cada vez una lista de los identificadores de atributos de contenido de documentos de las huellas que contienen un mismo valor de firma, lo cual proporciona un dato de atributo de contenido de documento unificado para cada valor de firma, y

* almacenar una correspondencia entre cada valor de firma y su dato de atributo de contenido de documento unificado.

15 **9.** Dispositivo según cualquiera de las reivindicaciones 6 a 8, **caracteriza por el hecho de que** el atributo de contenido de documento comprende un identificador de un contribuidor a dicho contenido de documento.

10. Dispositivo según cualquiera de las reivindicaciones anteriores, **caracterizado por el hecho de que** comprende además:

- un módulo calificador (type()) capaz de reconocer al menos una estructura de contenido de documento conocida y, en presencia de un documento cuya estructura ha sido reconocida, de asociar a este documento una determinada (702) característica de esta estructura, y **por el hecho de que** el generador de huella (signO()) está dispuesto para llamar el módulo calificador (tipo ()) para el establecimiento de la huella.

25 **11.** Dispositivo según la reivindicación 10, **caracterizado por el hecho de que** comprende además:

- un primer módulo de filtrado (filterA()) dispuesto para establecer a partir de un contenido de documento un contenido de documento depurado según un primer conjunto de reglas predeterminadas, las cuales dependen de dicho dato característico de la estructura del contenido del documento, y por el hecho de que:

- el generador de huella (signO()) está dispuesto para llamar, para un documento, el primer módulo de filtrado (filterA()) para obtener un contenido de documento depurado, y para establecer una huella llamando al extractor (200) para transformar este contenido de documento depurado en elementos, luego el generador de firmas (202) para los elementos así obtenidos.

35 **12.** Dispositivo según cualquiera de las reivindicaciones 10 y 11, **caracterizado por el hecho de que** comprende además:

- un segundo módulo de filtrado (filterB()) dispuesto para establecer a partir de un elemento de documento un elemento de documento depurado según un segundo conjunto de reglas predeterminadas que dependen del dato característico de la estructura del documento, y por el hecho de que:

- el generador de huella (signO()) está dispuesto para llamar, para un documento, el segundo módulo de filtrado (filterB()) para obtener elementos de documento depurados a partir de los elementos de documento obtenidos llamando al extractor (200), luego el generador de firmas (202) para estos elementos de documento depurados así obtenidos.

50 **13.** Dispositivo según cualquiera de las reivindicaciones anteriores, **caracterizado por el hecho de que** el supervisor (110) está dispuesto para llamar repetitivamente el módulo marcador de hora (dat()), con a cada vez una lista de todos los identificadores de referencia de temporalidad de las huellas que contienen un mismo valor de firma, para suministrar una referencia de temporalidad unificada para cada valor de firma.

55 **14.** Dispositivo según cualquiera de las reivindicaciones 1 a 13, **caracterizado por el hecho de que** el supervisor (110) está dispuesto para llamar repetitivamente el módulo marcador de hora (dat()), para cada valor de firma, con a cada vez un identificador de referencia de temporalidad de una huella que contiene este valor de firma y un identificador de la referencia de temporalidad unificada de este valor de firma para suministrar una nueva referencia de temporalidad unificada para cada valor de firma.

60 **15.** Dispositivo según cualquiera de las reivindicaciones anteriores, **caracterizado por el hecho de que** el módulo marcador de hora (dat()) está dispuesto para comparar dicha lista de identificadores de referencias de temporalidad con el conjunto de las referencias de temporalidad de los contenidos de documentos, y para deducir de esta comparación, en calidad de valor-resultado, unos identificadores de periodos temporales en dicha lista, estando cada periodo temporal constituido por referencias de temporalidades sucesivas con respecto al conjunto de las referencias de temporalidad.

65 **16.** Dispositivo según cualquiera de las reivindicaciones anteriores, **caracterizado por el hecho de que** está dispuesto para interactuar con una herramienta de gestión de versiones de contenidos de documentos,

recibiendo dicha memoria, desde esta herramienta de gestión de versiones, unas referencias de temporalidad para unos contenidos de documentos.

5 17. Dispositivo según la reivindicación 16, **caracterizado por el hecho de que** la herramienta de gestión de versiones de contenidos de documentos es además capaz de asociar un identificador de contribuidor a cada uno de los contenidos de documentos, recibiendo dicha memoria también estos identificadores de contribuidor.

10 18. Dispositivo según cualquiera de las reivindicaciones 16 y 17, **caracterizado por el hecho de que** el supervisor (110) está dispuesto para:

* interactuar con la herramienta de gestión de versiones de contenidos de documentos para establecer, a partir de una pluralidad de contenidos de documentos, un subconjunto de contenidos de documentos diferentes entre sí,

15 * hacer funcionar el generador de huella (signO()) sobre cada uno de los contenidos de documentos de dicho subconjunto, para obtener unas huellas que tienen conjuntamente una pluralidad de firmas,

* llamar repetitivamente, para cada uno de los valores de firmas obtenidos del subconjunto, el módulo marcador de hora (dat()) con una lista que comprende a la vez los identificadores de temporalidad de las huellas obtenidas del subconjunto y los identificadores de temporalidad de cada uno de los contenidos de documentos idénticos al contenido de documento del subconjunto del cual proviene el valor de firma.

20 19. Dispositivo según cualquiera de las reivindicaciones anteriores, **caracterizado por el hecho de que** el supervisor (110) está además dispuesto para:

25 * almacenar, además de la correspondencia entre cada valor de firma y su referencia de temporalidad, un identificador para el conjunto de los contenidos de documentos de los que provienen el valor de firma y la referencia de temporalidad.

20. Dispositivo según la reivindicación 19, **caracterizado por el hecho de que** comprende además:

30 - un comparador de huellas (comp()) capaz de comparar un primer conjunto de huellas y un segundo conjunto de huellas, que tienen respectivamente un primer identificador y un segundo identificador, comprendiendo el primer conjunto de huellas unos valores de firmas y sus referencias de temporalidad unificada correspondiente al primer identificador, y comprendiendo el segundo conjunto de huellas valores de firmas y sus referencias de temporalidad unificada correspondiente a dicho segundo identificador, proporcionando el conjunto de las comparaciones logradas un subconjunto de dependencias, con, para cada valor de firma que es común, unas referencias de temporalidades unificadas correspondientes a dichos conjuntos de huellas primero y segundo,

35 - un módulo de orientación (dir()) capaz de calificar, para un valor de firma común, uno de entre los identificadores primero y segundo en calidad de identificador de origen o "fuente" y el otro identificador en calidad de identificador de destino o "diana", según un conjunto de reglas predeterminadas, sobre la base de las referencias de temporalidad unificadas,

40 - siendo el comparador de huellas (comp()) capaz de llamar el módulo de orientación (dir()) para cada valor de firma de dicho subconjunto de dependencias con las referencias de temporalidad unificadas correspondientes, con el fin de proponer una distinción entre la fuente y la diana.

45 21. Dispositivo según la reivindicación 20, **caracterizado por el hecho de que** el módulo de orientación (dir()) está dispuesto para establecer en calidad de identificador de conjunto de contenidos de documentos de origen el identificador correspondiente a la referencia de temporalidad unificada más antigua.

50 22. Dispositivo según la reivindicación 21, **caracterizado por el hecho de que** comprende además un módulo de partición (agrDep1 ()) de dicho subconjunto de dependencias entre dichos conjuntos de huellas primero y segundo, estando este módulo de partición agrDep1() dispuesto para repartir dicho subconjunto según el identificador de origen de las firmas y luego para respectivamente crear agregados de dependencias, sobre la base de la aplicación iterativa de al menos uno de los criterios de partición siguientes:

55 - tomar la firma común más antigua (1304) aún no agrupada, y agrupar con ella las otras firmas comunes que provienen de referencias de temporalidad lado "fuente" que son anteriores a esta firma común más antigua (1306),

- extraer, de un grupo determinado, una firma común cuya referencia de temporalidad "diana" es posterior a la firma más antigua de entre las firmas comunes aún no agrupadas (1310),

60 - efectuándose la iteración hasta verificación de un criterio de fin de bucle determinado.

23. Dispositivo según cualquiera de las reivindicaciones 20 a 22, **caracterizado por el hecho de que** comprende además:

65 - un comparador multi-documentos, capaz de recibir N conjuntos de huellas, luego de realizar las comparaciones orientadas, de un tal conjunto de huellas a otro, para determinadas al menos de las firmas, lo cual proporciona, por comparación de al menos uno de los conjuntos de huellas con los otros, una pluralidad de subconjuntos de

dependencia, y a continuación, para cada firma en estos subconjuntos de dependencia, de separar las dependencias cuya referencia de temporalidad "diana", para la firma considerada, es posterior en la fecha más antigua de entre los N conjuntos de huellas, para la firma considerada.

- 5 **24.** Dispositivo según cualquiera de las reivindicaciones 20 a 23, **caracterizado por el hecho de que** comprende un módulo de gestión global de documentos (150), capaz de establecer una lista fechada de contribuidores, verificados a partir de las huellas y de las eventuales dependencias.
- 10 **25.** Dispositivo según cualquiera de las reivindicación anteriores, **caracterizado por el hecho de que** los contenidos de documentos son identificables según al menos dos categorías de entre: un documento determinado, un estado de documento, una versión de documento, y un juego de documentos susceptible de comprender varias versiones de documentos.

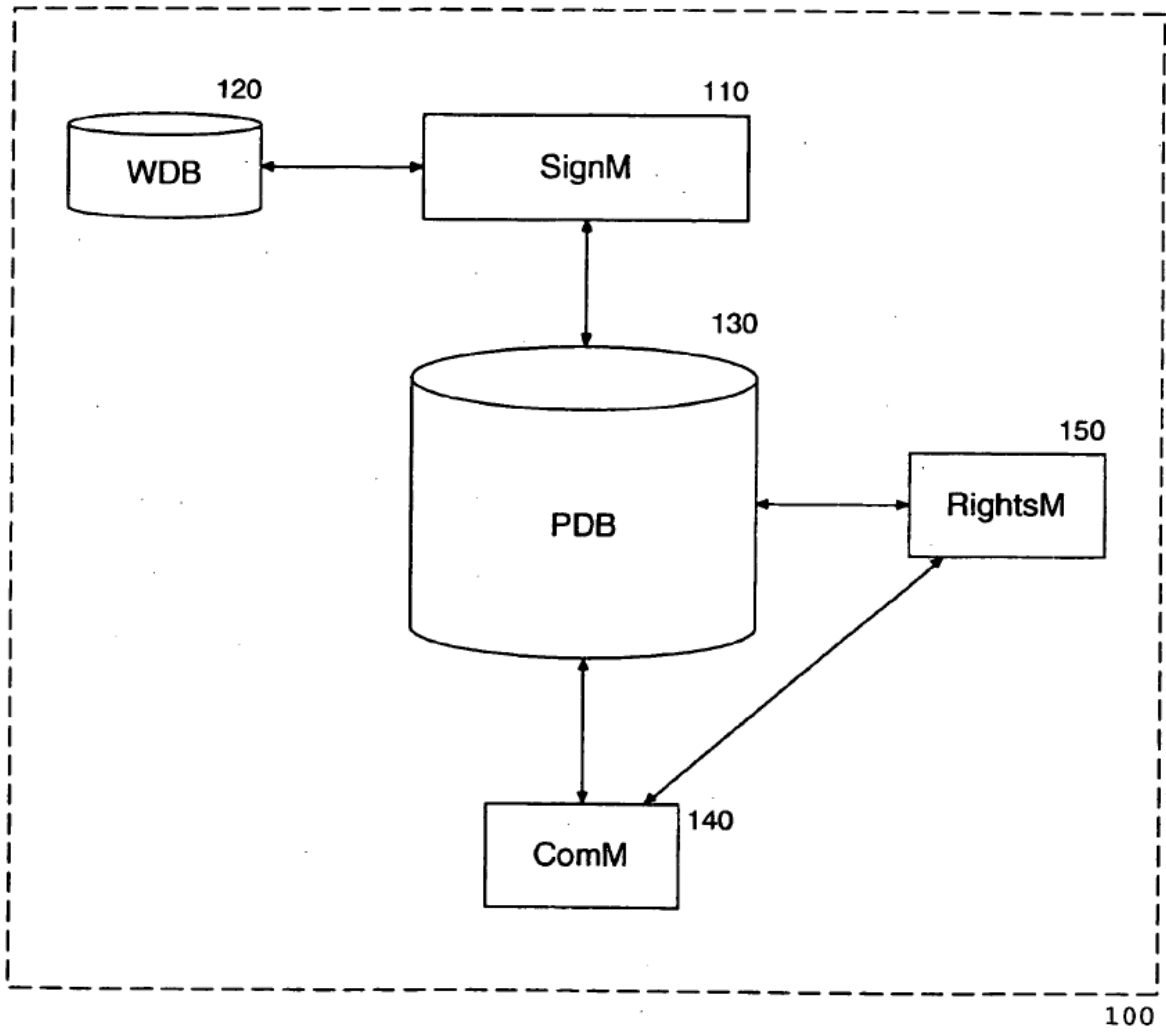


Fig. 1

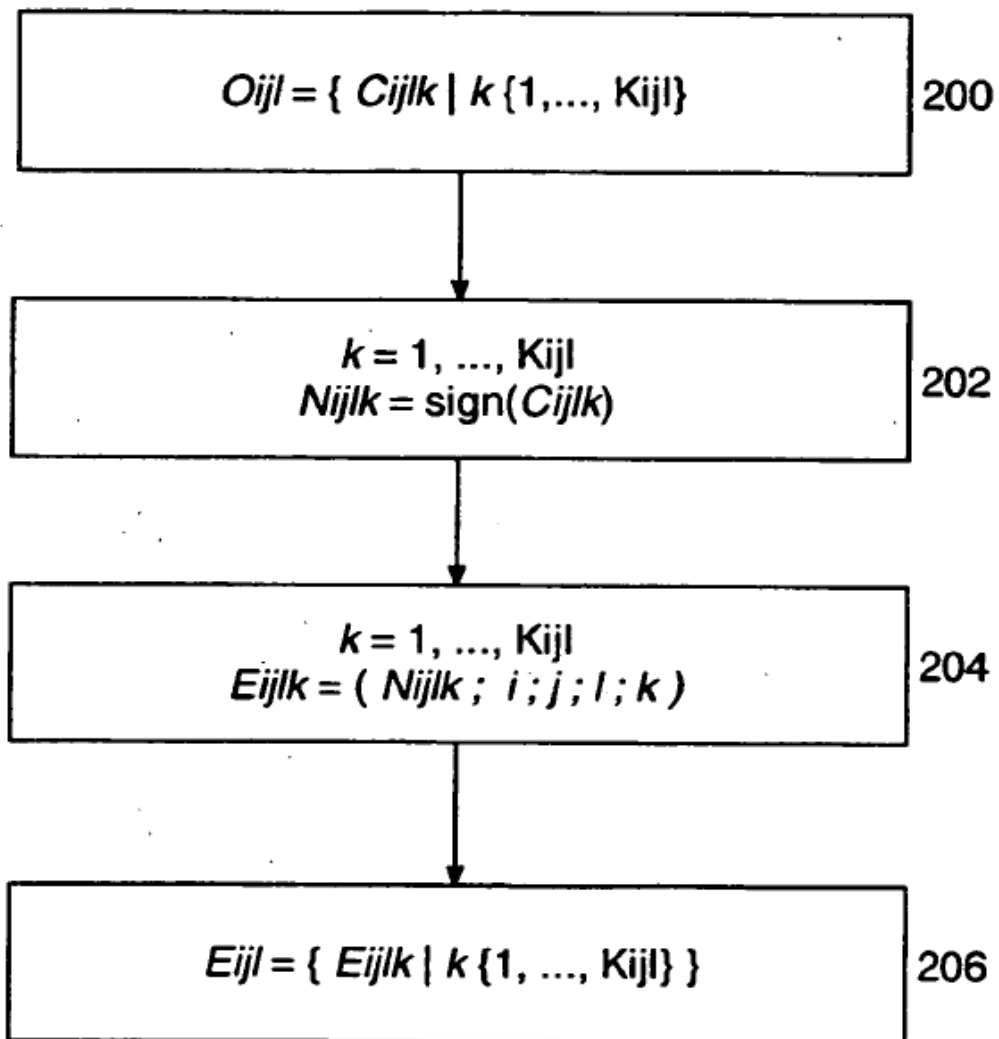


Fig. 2

COL	301	302	303	304	305	306	307	308	ROW
	1	1	1	1	N1,1,1,1	AE1,1,1,1	T11	A11	301
	1	1	1	2	N1,1,1,2	AE1,1,1,2	T11	A11	302
	303
	1	1	1	K111	N1,1,1,K111	AE1,1,1,K111	T11	A11	304
300	1	1	2	1	N1,1,2,1	AE1,1,2,1	T11	A11	305
	1	1	2	2	N1,1,2,2	AE1,1,2,2	T11	A11	306
	307
	1	1	2	K112	N1,1,2,K112	AE1,1,2,K112	T11	A11	308
	309
304	1	1	L11	1	N1,1,L11,1	AE1,1,L11,1	T11	A11	310
	1	1	L11	2	N1,1,L11,2	AE1,1,L11,2	T11	A11	311
	312
	1	1	L11	K11L11	N1,1,L11,K11L11	AE1,1,L11,K11L11	T11	A11	313
	1	2	1	1	N1,2,1,1	AE1,2,1,1	T12	A12	314
	1	2	1	2	N1,2,1,2	AE1,2,1,2	T12	A12	315
	316
	1	2	1	K121	N1,2,1,K121	AE1,2,1,K121	T12	A12	317
306	1	2	2	1	N1,2,2,1	AE1,2,2,1	T12	A12	318
	1	2	2	2	N1,2,2,2	AE1,2,2,2	T12	A12	319
	320
	1	2	2	K122	N1,2,2,K122	AE1,2,2,K122	T12	A12	321
	322
	1	2	L12	1	N1,2,L12,1	AE1,2,L12,1	T12	A12	323
	1	2	L12	2	N1,2,L12,2	AE1,2,L12,2	T12	A12	324
	325
	1	2	L12	K12L12	N1,2,L12,K12L12	AE1,2,L12,K12L12	T12	A12	326
	327
	1	J1	1	1	N1,J1,1,1	AE1,J1,1,1	T1J1	A1J1	328
	1	J1	1	2	N1,J1,1,2	AE1,J1,1,2	T1J1	A1J1	329
	330
	1	J1	1	K1J11	N1,J1,1,K1J11	AE1,J1,1,K1J11	T1J1	A1J1	331
	1	J1	2	1	N1,J1,2,1	AE1,J1,2,1	T1J1	A1J1	332
	1	J1	2	2	N1,J1,2,2	AE1,J1,2,2	T1J1	A1J1	333
	334
	1	J1	2	K1J12	N1,J1,2,K1J12	AE1,J1,2,K1J12	T1J1	A1J1	335
	1	J1	L1J1	1	N1,J1,L1J1,1	AE1,J1,L1J1,1	T1J1	A1J1	336
	1	J1	L1J1	2	N1,J1,L1J1,2	AE1,J1,L1J1,2	T1J1	A1J1	337
	338
308	1	J1	L1J1	K1J1L1J1	N1,J1,L1J1,K1J1L1J1	AE1,J1,L1J1,K1J1L1J1	T1J1	A1J1	339

Fig. 3

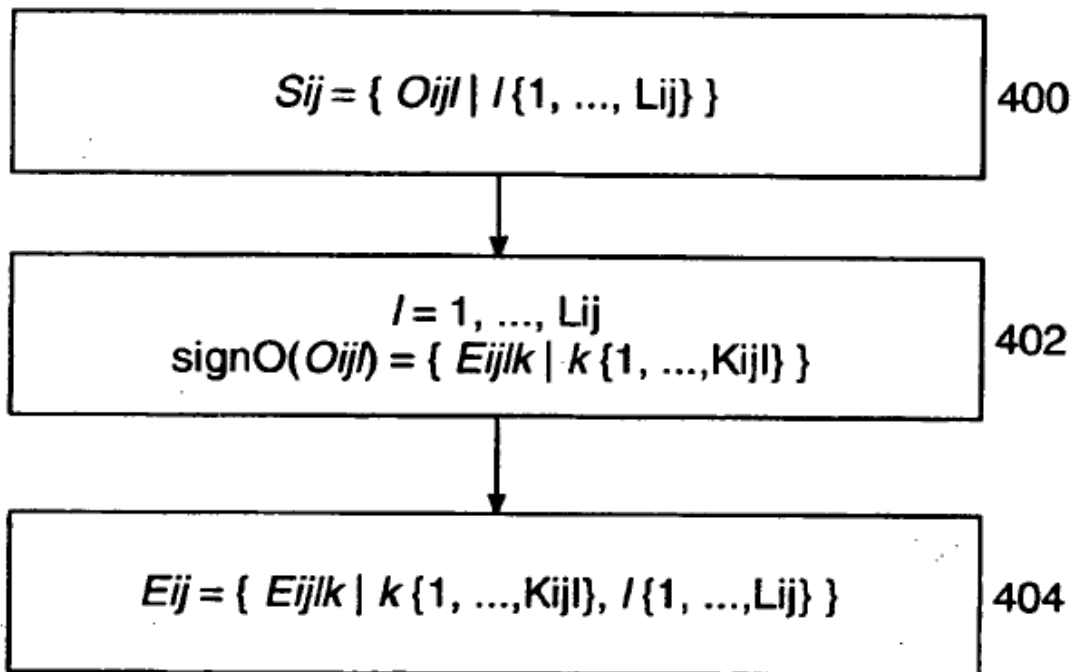


Fig. 4

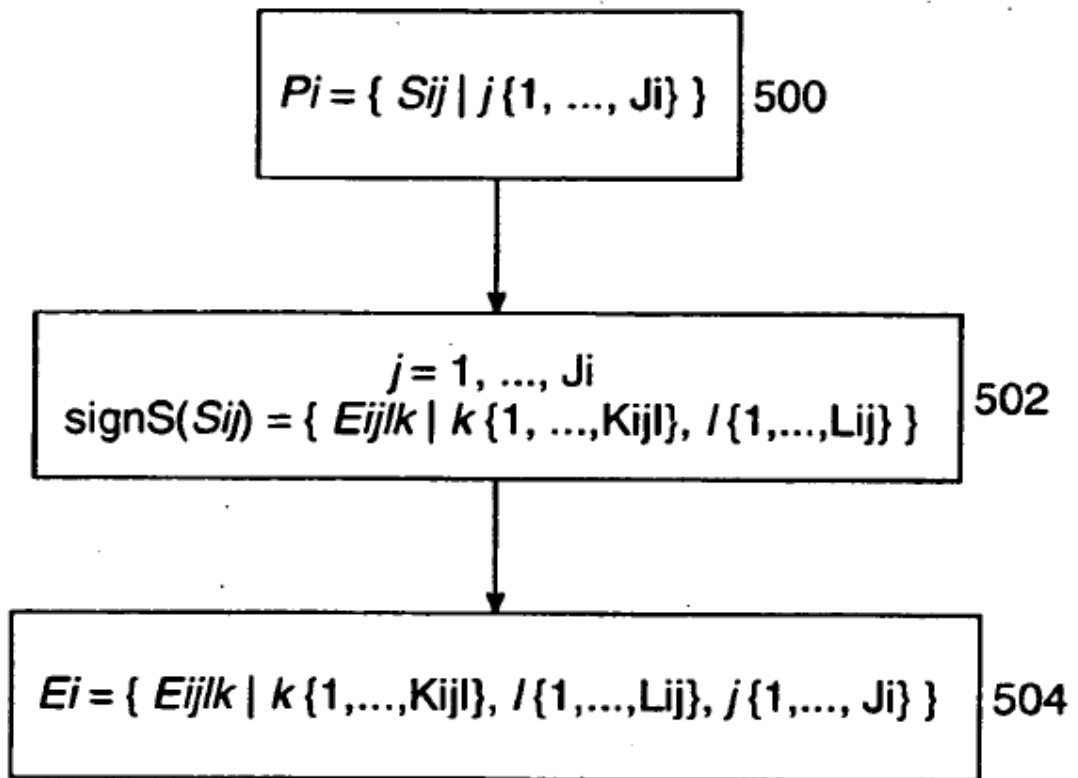


Fig. 5

604

COL	601	602	603	604	605	606	607	608	609	610
Nia	i	ja1	la1	ka1	Ti,ja1	AEi,ja1,la1,ka1	AOi,ja1,la1	Via	Wia	601
Nia	i	ja2	la2	ka2	Ti,ja2	AEi,ja2,la2,ka2	AOi,ja2,la2	Via	Wia	602
...	603
Nia	i	jan	lan	kan	Ti,jan	AEi,jan,lan,kan	AOi,jan,lan	Via	Wia	604
Nib	i	jb1	lb1	kb1	Ti,jb1	AEi,jb1,lb1,kb1	AOi,jb1,lb1	Vib	Wib	605
...	606
Nib	i	jbm	lbm	kbm	Ti,jbm	AEi,jbm,lbm,kbm	AOi,jbm,lbm	Vib	Wib	607
...	608
Niq	i	jq1	lq1	kq1	Ti,jq1	AEi,jq1,lq1,kq1	AOi,jq1,lq1	Viq	Wiq	609
...	610
Niq	i	jqp	lqp	kqp	Ti,jqp	AEi,jqp,lqp,kqp	AOi,jqp,lqp	Viq	Wiq	611
...	612
Niq	i	qgr	lgr	kgr	Ti,jgr	AEi,jgr,lgr,kgr	AOi,jgr,lgr	Viq	Wiq	613

600

602

Fig. 6

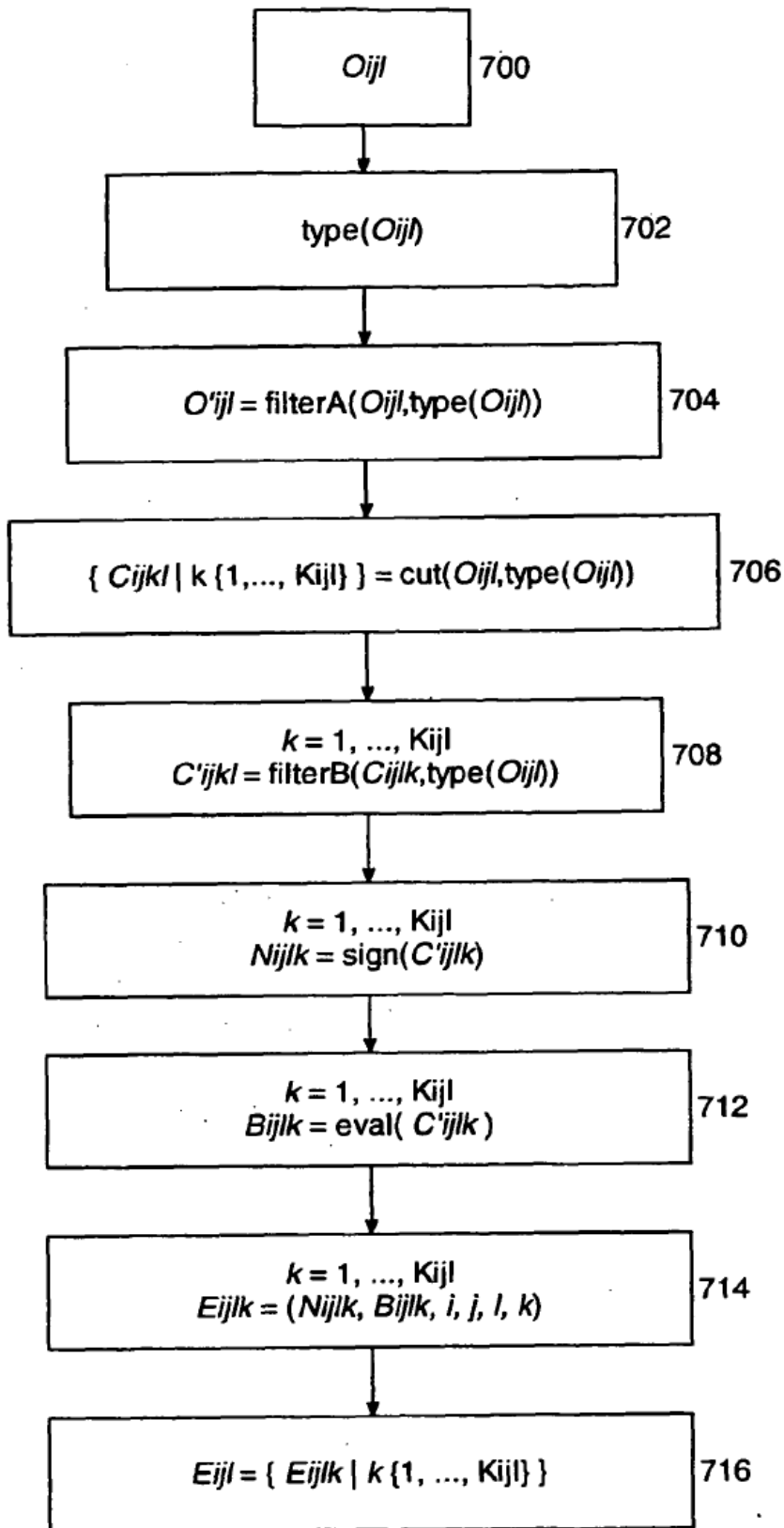


Fig. 7

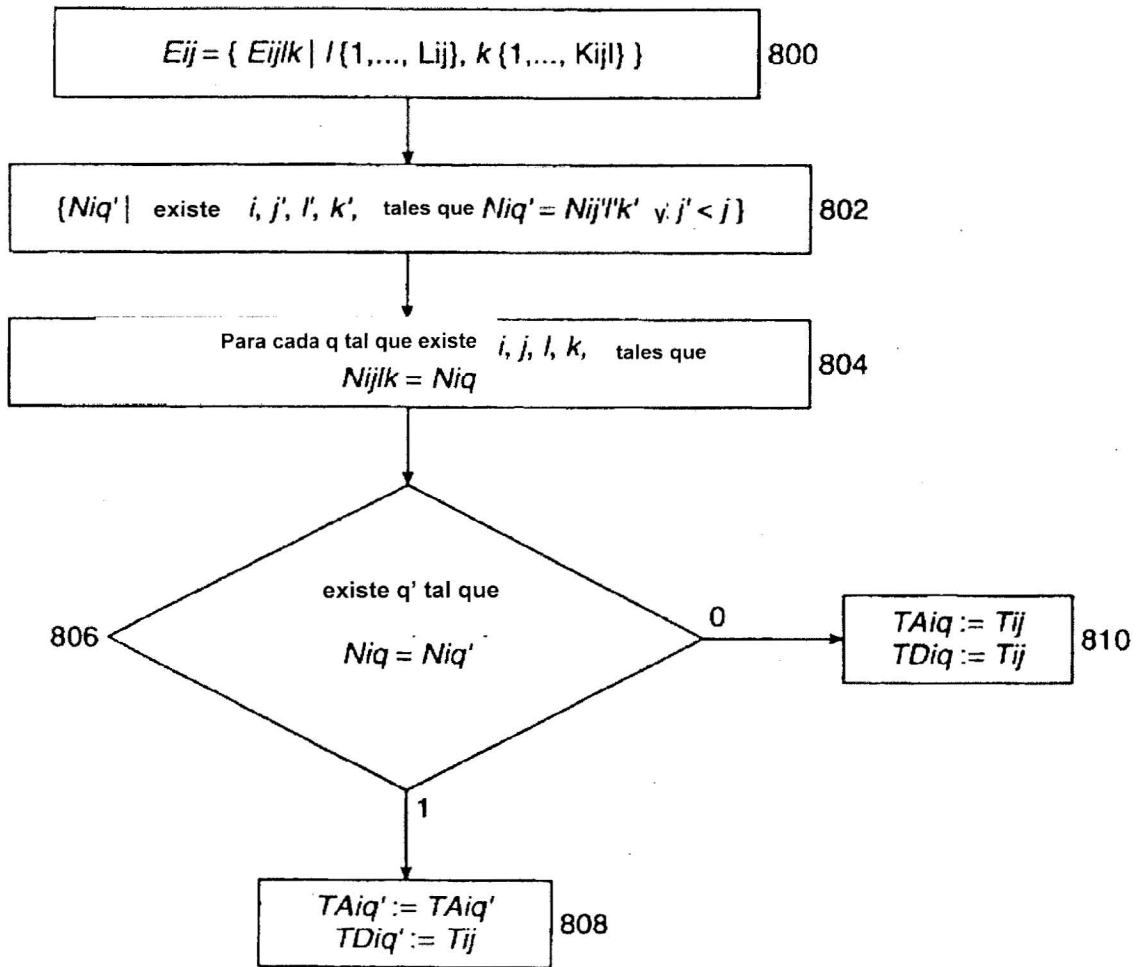


Fig. 8

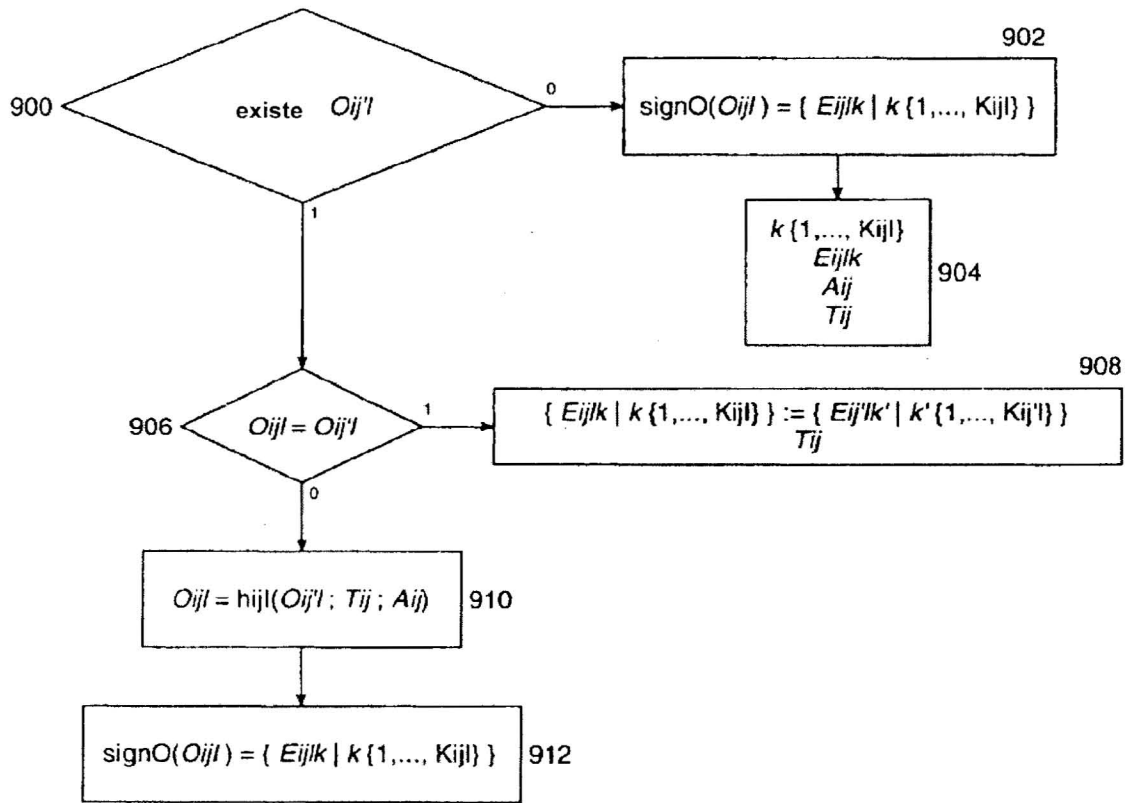


Fig. 9

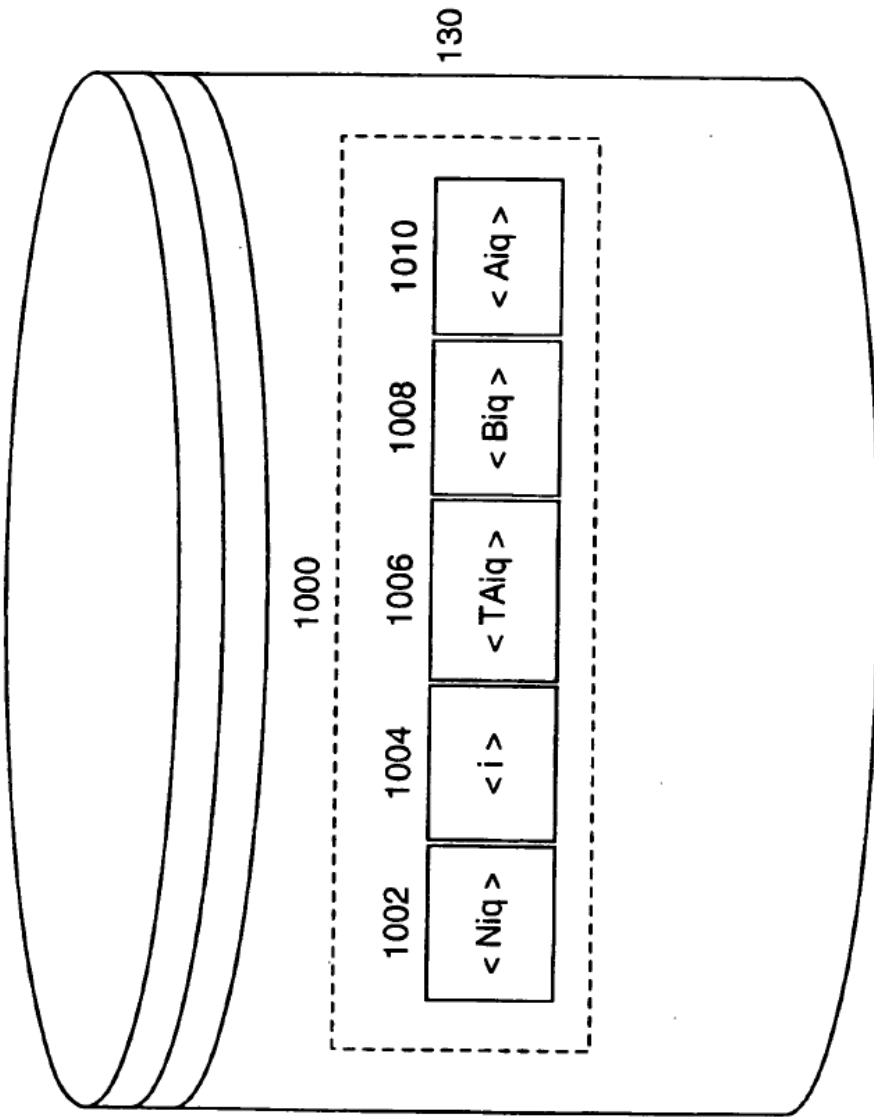


Fig. 10

		1100									
col	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	ROW
	N1a	1	ja1	la1	ka1	T1ja1	AE1,ja1,la1,ka1	AO1,ja1,la1	V1a	W1a	1101
	N1a	1	ja2	la2	ka2	T1ja2	AE1,ja2,la2,ka2	AO1,ja2,la2	V1a	W1a	1102
	1103
	N1a	1	jan	lan	kan	T1jan	AE1,jan,lan,kan	AO1,jan,lan	V1a	W1a	1104
	N1b	1	jb1	lb1	kb1	T1jb1	AE1,jb1,lb1,kb1	AO1,jb1,lb1	V1b	W1b	1105
	1106
1102	N1b	1	jbm	lbm	kbm	T1jbm	AE1,jbm,lbm,kbm	AO1,jbm,lbm	V1b	W1b	1107
	1108
	N1q	1	jq1	lq1	kq1	T1jq1	AE1,jq1,lq1,kq1	AO1,jq1,lq1	V1q	W1q	1109
	1110
	N1q	1	jqp	lqp	kqp	T1jqp	AE1,jqp,lqp,kqp	AO1,jqp,lqp	V1q	W1q	1111
	1112
	N1q	1	jqr	lqr	kqr	T1jqr	AE1,jqr,lqr,kqr	AO1,jqr,lqr	V1q	W1q	1113
	N2a	2	ja1	la1	ka1	T2ja1	AE2,ja1,la1,ka1	AO2,ja1,la1	V2a	W2a	1114
	N2a	2	ja2	la2	ka2	T2ja2	AE2,ja2,la2,ka2	AO2,ja2,la2	V2a	W2a	1115
	1116
	N2a	2	jan	lan	kan	T2jan	AE2,jan,lan,kan	AO2,jan,lan	V2a	W2a	1117
	N2b	2	jb1	lb1	kb1	T2jb1	AE2,jb1,lb1,kb1	AO2,jb1,lb1	V2b	W2b	1118
	1119
1104	N2b	2	jbm	lbm	kbm	T2jbm	AE2,jbm,lbm,kbm	AO2,jbm,lbm	V2b	W2b	1120
	1121
	N2q	2	jq1	lq1	kq1	T2jq1	AE2,jq1,lq1,kq1	AO2,jq1,lq1	V2q	W2q	1122
	1123
	N2q	2	jqp	lqp	kqp	T2jqp	AE2,jqp,lqp,kqp	AO2,jqp,lqp	V2q	W2q	1124
	1125
	N2q	2	jqr	lqr	kqr	T2jqr	AE2,jqr,lqr,kqr	AO2,jqr,lqr	V2q	W2q	1126
	1127
	Nia	i	ja1	la1	ka1	Tija1	AEi,ja1,la1,ka1	AOi,ja1,la1	Via	Wia	1128
	Nia	i	ja2	la2	ka2	Tija2	AEi,ja2,la2,ka2	AOi,ja2,la2	Via	Wia	1129
	1130
	Nia	i	jan	lan	kan	Tijan	AEi,jan,lan,kan	AOi,jan,lan	Via	Wia	1131
	Nib	i	jb1	lb1	kb1	Tijb1	AEi,jb1,lb1,kb1	AOi,jb1,lb1	Vib	Wib	1132
	1133
1106	Nib	i	jbm	lbm	kbm	Tijbm	AEi,jbm,lbm,kbm	AOi,jbm,lbm	Vib	Wib	1134
	1135
	Niq	i	jq1	lq1	kq1	Tijq1	AEi,jq1,lq1,kq1	AOi,jq1,lq1	Viq	Wiq	1136
	1137
	Niq	i	jqp	lqp	kqp	Tijqp	AEi,jqp,lqp,kqp	AOi,jqp,lqp	Viq	Wiq	1138
	1139
	Niq	i	jqr	lqr	kqr	Tijqr	AEi,jqr,lqr,kqr	AOi,jqr,lqr	Viq	Wiq	1140

Fig. 11

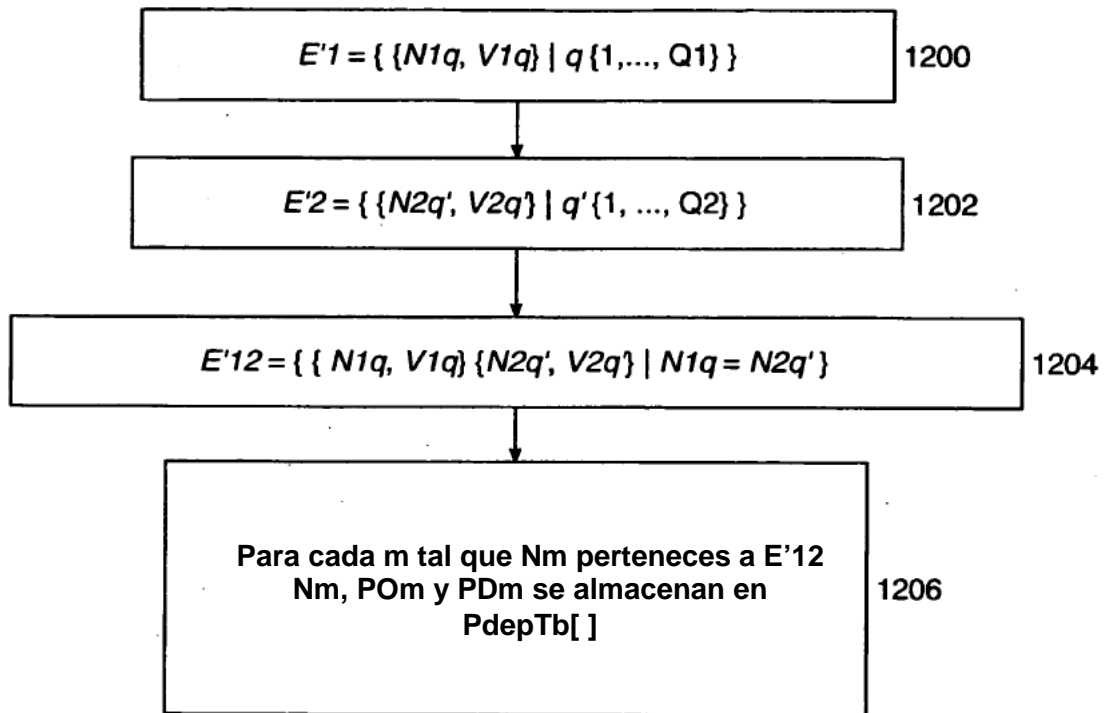


Fig. 12

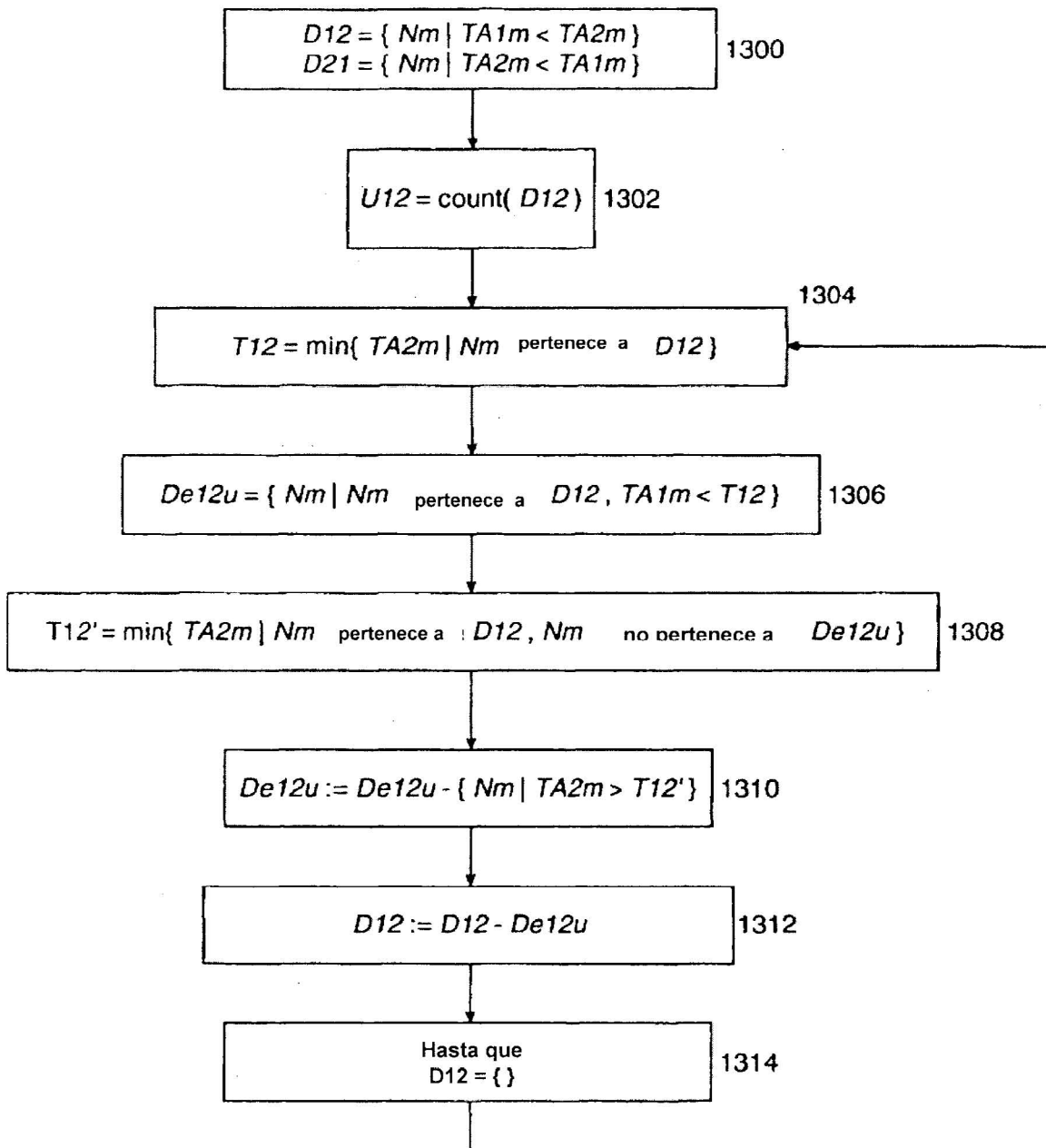


Fig. 13

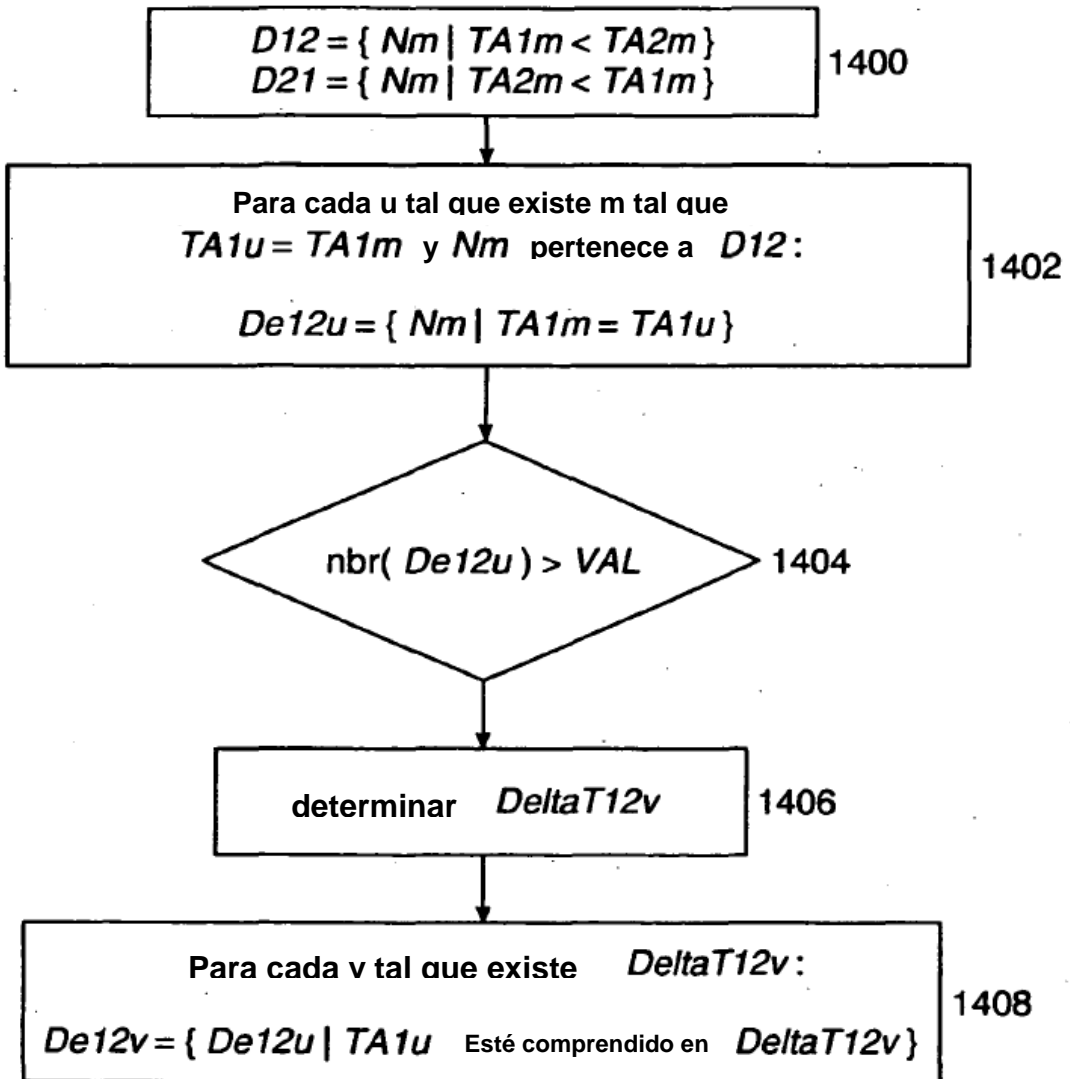


Fig. 14

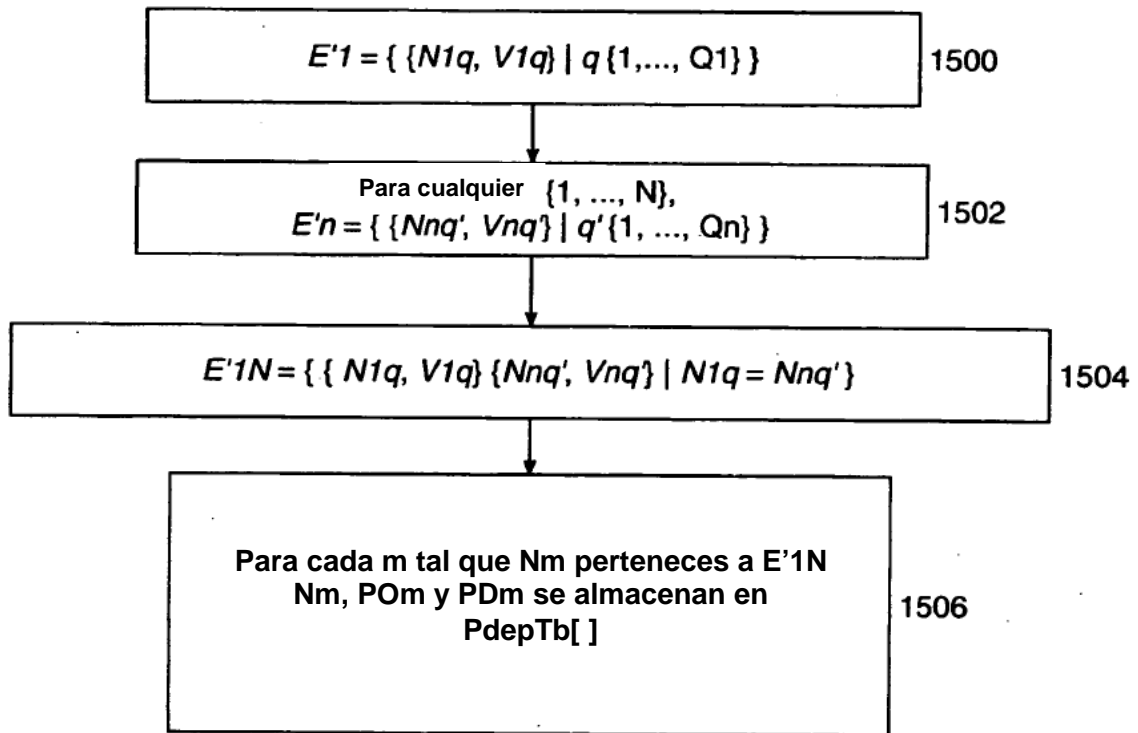


Fig. 15