



(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 300 429**

(51) Int. Cl.:
G06F 17/30 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(86) Número de solicitud europea: **02701352 .3**

(86) Fecha de presentación : **30.01.2002**

(87) Número de publicación de la solicitud: **1358583**

(87) Fecha de publicación de la solicitud: **05.11.2003**

(54) Título: **Procedimiento de codificación y decodificación de un trayecto en un diagrama de árbol de un documento estructurado.**

(30) Prioridad: **30.01.2001 FR 01 01243**

(73) Titular/es: **Expway**
16, rue Vauthier Le Noir
51100 Reims, FR

(45) Fecha de publicación de la mención BOPI:
16.06.2008

(72) Inventor/es: **Seyrat, Claude y**
Thienot, Cédric

(45) Fecha de la publicación del folleto de la patente:
16.06.2008

(74) Agente: **Carvajal y Urquijo, Isabel**

ES 2 300 429 T3

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

DESCRIPCIÓN

Procedimiento de codificación y decodificación de un trayecto en un diagrama de árbol de un documento estructurado.

La presente invención se refiere a un procedimiento de codificación y de decodificación de un trayecto en un diagrama de árbol de documento estructurado.

Ésta se aplica especialmente, aunque no exclusivamente, a la compresión/descompresión de partes de documentos estructurados. Tales documentos consisten por ejemplo en datos estructurados multimedia, imágenes o secuencias de imágenes de vídeo o digitales, obras cinematográficas o programas de vídeo, o datos que describen tal información.

Un documento estructurado es una colección de conjuntos de información asociados cada uno a un tipo y a atributos, y compuestos entre ellos según relaciones principalmente jerárquicas. Estos documentos emplean un lenguaje de estructuración tal como SGML, HTML, XML, que permite especialmente distinguir los diferentes subconjuntos de información que componen el documento. Al contrario, en un documento denominado lineal, la información de contenido del documento está mezclada con información de presentación y de tipificación.

Un documento estructurado incluye marcas de separación de los diferentes conjuntos de información del documento. En el caso de los formatos SGML, XML, o HTML, estas marcas denominadas “etiquetas” tienen la forma “<XXXX>” y “</_XXXX>”, indicando la primera marca el principio de un conjunto de información “XXXX” y la segunda el fin de este conjunto. Un conjunto de información puede estar compuesto por varios conjuntos de información de nivel más bajo. Así, un documento estructurado presenta un esquema de estructura jerárquica o en árbol, representando cada nodo un conjunto de información y estando unido a un nodo de nivel jerárquico superior que representa un conjunto de información que contiene los conjuntos de información de nivel inferior. Los nodos situados en el extremo de una rama de esta estructura en árbol representan conjuntos de información que contienen datos de un tipo predefinido, que no pueden descomponerse en subconjuntos de información.

Así, un documento estructurado contiene marcas de separación representadas en forma de datos de texto o binarios, delimitando estas marcas conjuntos o subconjuntos de información que pueden contener a su vez otros subconjuntos de información delimitados por marcas.

Por otro lado, un documento estructurado está asociado a lo que se denomina un esquema de estructura que define en forma de reglas la estructura y el tipo de información de cada conjunto de información del documento. Un esquema está constituido por grupos superpuestos de estructuras de conjuntos de información, pudiendo ser estos grupos secuencias ordenadas, grupos de elementos alternativos o grupos de elementos necesarios, ordenados o no ordenados.

Actualmente, cuando un documento estructurado debe transmitirse, preferiblemente se comprime previamente, para minimizar el volumen de los datos que deben transmitirse. Para mayor eficacia de un tratamiento de compresión de este tipo, los datos de estructuración del documento también se comprimen, sabiendo que el destinatario del documento supuestamente debería conocer previamente el esquema de estructura del documento y puede utilizar este esquema para determinar en cada momento qué conjunto de información va a recibir. Por tanto es indispensable que la estructura del documento transmitido corresponda exactamente al esquema de estructura que el destinatario del documento prevea utilizar para la recepción y la decodificación del documento, sin lo cual, el destinatario no puede determinar el tipo de datos transmitidos concretamente, y por tanto no puede decodificarlos ni reconstituir el documento de origen.

Ahora bien los documentos estructurados que deben transmitirse tienden a hacerse cada vez más voluminosos. Se prevé por ejemplo transmitir o difundir de esta manera descripciones completas de obras cinematográficas o de programas de televisión. En este contexto, si se produce un error de transmisión durante la transmisión de un documento, el destinatario del documento puede no estar ya en condiciones de determinar qué subconjunto está transmitiéndose, de manera que el conjunto del documento debe transmitirse de nuevo. Además, si se desea transmitir y simultáneamente visualizar en una pantalla una secuencia cinematográfica, puede ser necesario respetar intervalos horarios de transmisión de los diferentes elementos de la secuencia. Determinados elementos de la secuencia deben además poder transmitirse varias veces para permitir a un destinatario que no estaba conectado al principio de la transmisión de la secuencia, recibir y visualizar el fin de ésta.

Puede ser también necesario sustituir una parte de documento por otra, teniendo estas dos partes el mismo esquema de estructura. La solución que consiste en retransmitir el conjunto del documento conduciría a aumentar considerablemente el volumen de información que debe transmitirse. Por tanto es deseable dividir un documento en varias partes que pueden utilizarse o transmitirse por separado. No obstante, para poder descomprimir una parte de documento, es necesario poder determinar dónde se sitúa esta parte de documento en el esquema de estructura del documento.

Con este fin, existen varias soluciones que consisten en describir un trayecto en el diagrama de árbol del documento, a partir del nodo raíz del documento, para llegar al nodo principal de la parte de documento en cuestión. Para ello, se han perfeccionado métodos de descripción de trayectos en un diagrama de árbol. No obstante, estos métodos no

están optimizados en cuanto al número de elementos de información necesarios para describir un trayecto de este tipo. Además, estos métodos no tienen en cuenta todas las posibilidades que existen en la definición de un esquema de estructura de un documento, de tal manera que no garantizan en todos los casos que el trayecto que se reconstituirá, corresponde al trayecto de origen. Por tanto esto tiene como resultado riesgos de error en la determinación de la posición en el diagrama de árbol del documento de una parte de éste último, y por tanto riesgos de error, incluso imposibilidad de descodificación de la parte de documento.

Así, el lenguaje XML-esquema, normalmente utilizado en los documentos estructurados, permite lo que se denomina el polimorfismo, es decir que pueden definirse subtipos de un tipo de datos estructurado, siendo los subtipos casos particulares de datos que corresponden al tipo. Por ejemplo, en un tipo "cadena de caracteres" puede encontrarse un subtipo "mes del año". En este caso, el modelo de estructura puede indicar que un nodo del diagrama de árbol es de tipo "cadena de carácter" y el documento comprende a nivel de este nodo un conjunto de información de tipo "mes del año". Este lenguaje autoriza también sustituciones de nombres de conjuntos de información. Se observa que los métodos de codificación de trayecto actuales no tienen en cuenta estas posibilidades.

La presente invención tiene como objetivo suprimir estos inconvenientes. Este objetivo se logra previendo un procedimiento de codificación de un trayecto en una estructura jerárquica de documento estructurado, definida por un esquema de estructura de documento, estando definido este trayecto por una sucesión de segmentos, uniendo cada segmento un nodo de origen y un nodo de destino, representando cada nodo un elemento de información del documento, estando cada elemento de información asociado en el esquema de estructura a al menos un tipo de información, caracterizado porque comprende:

- una fase previa que comprende una etapa que consiste en asociar a cada nodo en cuestión del esquema de estructura, una lista de parejas constituidas por un nombre y un tipo de elemento de información, representadas por todos los nodos susceptibles de relacionarse directamente con el nodo en cuestión, y en asociar un código binario respectivo a cada pareja de nombre y tipo de elemento de información, y

- una fase de codificación del trayecto que comprende una etapa que consiste para cada segmento del trayecto que debe codificarse, en determinar e insertar en el código de trayecto un código binario de nodo asociado a la pareja nombre - tipo del nodo de destino del segmento.

Ventajosamente, la fase de codificación de trayecto comprende además una etapa que consiste en determinar un código binario de posición del nodo de destino del segmento, con respecto a los otros nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.

Según una particularidad de la invención, la fase de codificación de trayecto comprende además una etapa que consiste en generar un código de trayecto que comprende una sucesión de códigos de segmentos, comprendiendo cada código de segmento un código binario de nodo del nodo de destino del segmento, y un código binario de posición del nodo de destino del segmento.

Según otra particularidad de la invención, la fase de codificación de trayecto comprende además una etapa que consiste en generar un código de trayecto que comprende una sucesión de códigos de segmentos, comprendiendo cada código de segmento un código binario de nodo del nodo de destino del segmento, y una secuencia de códigos de posición que da la posición de todos los nodos referenciados en la sucesión de código de segmentos.

Preferiblemente, la fase previa comprende además una etapa que consiste en determinar un número máximo de nodos susceptibles de relacionarse directamente con el nodo en cuestión, para determinar el tamaño del código binario de posición de nodo.

Según otra particularidad de la invención, al menos uno de los elementos de información de la estructura del documento comprende atributos, teniendo el trayecto que debe codificarse como elemento de destino un atributo, comprendiendo además la fase de codificación una etapa de inserción en el código de cada segmento de un código de tipo de segmento que indica si el nodo de destino del segmento es un atributo o un elemento de información.

Según otra particularidad adicional de la invención, la fase de codificación comprende además una etapa de inserción en el código de trayecto de un código de fin de trayecto.

Preferiblemente, el código de fin de trayecto es un código de tipo de segmento que tiene un valor predefinido.

Según otra particularidad adicional de la invención, el nodo de origen de cada segmento está situado en un nivel jerárquico superior al del nodo de destino en el esquema de estructura del documento, y porque la fase de codificación comprende además una etapa de inserción en el código de trayecto de al menos un código de tipo de segmento que tiene un valor predefinido que indica que el nodo de origen del segmento que debe codificarse siguiente es el nodo de destino del segmento que debe codificarse precedente.

Según otra particularidad adicional de la invención, la fase de codificación comprende además una etapa de inserción en el código de trayecto de un código que indica si el trayecto codificado es un trayecto absoluto desde el nodo raíz del documento, o un trayecto relativo desde un nodo cualquiera del esquema de estructura del documento.

El objeto de la invención se refiere también a un procedimiento de descodificación de un código de trayecto en una estructura jerárquica de documento estructurado, definida por un esquema de estructura de documento, comprendiendo este código de trayecto una sucesión de códigos de segmentos, uniendo cada segmento un nodo de origen a un nodo de destino que constituye el nodo de origen del segmento siguiente, representando cada nodo un elemento de información del documento, estando cada elemento de información asociado en el esquema de estructura a al menos un tipo de información, caracterizado porque cada segmento está definido en el código de trayecto al menos por un código binario de nodo que representa una pareja nombre - tipo, constituido por un nombre y por un tipo de elemento de información del elemento de información representado por el nodo de destino del segmento, comprendiendo el procedimiento:

- una fase previa que consiste en asociar a cada nodo en cuestión del esquema de estructura, una lista de parejas nombre - tipo de elemento de información constituidas por un nombre y por un tipo de elementos de información, representadas por todos los nodos susceptibles de relacionarse directamente con el nodo en cuestión, y en asociar un código binario respectivo a cada pareja nombre - tipo de elemento de información, y

- una fase de descodificación del código de trayecto que consiste, para segmento del código de trayecto que debe descodificarse, en descodificar el código de nodo que representa la pareja nombre, tipo del nodo de destino del código de segmento con ayuda de la lista de parejas nombre - tipo del nodo de destino.

Ventajosamente, cada segmento comprende además en el código de trayecto que debe descodificarse un código de posición del nodo de destino con respecto a los otros nodos susceptibles de estar unidos directamente al nodo de origen del segmento, comprendiendo la fase de descodificación además una etapa que consiste, para cada segmento, en descodificar el código binario de posición del nodo de destino del segmento, en función de las posiciones respectivas de todos los nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.

Según una particularidad de la invención, la descodificación del código binario de nodo que representa la pareja nombre - tipo de elemento de información comprende una etapa de determinación del tamaño en número de bits de este código y de búsqueda de este código en la lista de las parejas nombre - tipo del nodo de origen del segmento.

Según otra particularidad de la invención, la descodificación del código binario de posición del nodo de destino del segmento comprende la determinación del tamaño en número de bits de este código en función del número máximo de nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.

Preferiblemente, cada código de segmento comprende un código de tipo de segmento, comprendiendo la fase de descodificación del trayecto además, para cada segmento, la descodificación del código de tipo de segmento.

Ventajosamente, el código de tipo de segmento de cada código de segmento del código de trayecto permite determinar si el segmento tiene como nodo de destino un elemento de información o un atributo del nodo de origen del segmento.

Según otra particularidad de la invención, el procedimiento comprende la determinación del fin del código de trayecto, que está marcado por un código de tipo de segmento que tiene un primer valor predefinido.

Preferiblemente, si el código de tipo de segmento tiene un segundo valor predefinido, el código de segmento que debe descodificarse siguiente en el código de trayecto tiene como nodo de destino el nodo de origen del segmento que debe descodificarse precedente.

Una forma de realización preferida de la invención se describirá a continuación, a modo de ejemplo no limitativo, con referencia a los dibujos adjuntos en los que:

las figuras 1a y 1b representan una parte de una estructura en árbol de documento estructurado, en la que cada nodo simboliza un conjunto o subconjunto de información, respectivamente antes y después de la definición de una rama entre dos nodos;

la figura 2 representa la estructura general de un trayecto según la invención en un diagrama de árbol de estructura de documento;

la figura 3 muestra en forma de un organigrama el tratamiento ejecutado por un ordenador de codificación de trayecto, según la invención;

la figura 4 muestra en forma de un organigrama el tratamiento ejecutado por un ordenador de descodificación según la invención.

La figura 1a representa un esquema de estructura de un documento estructurado que comprende un nodo x que no es necesariamente el nodo raíz del documento. Este nodo x se descompone en tres nodos, estando representado sólo el segundo nodo y. A continuación el nodo y se descompone en tres nodos de los que el segundo nodo es T, que comprende cuatro nodos a, b, y c indicados en la figura mediante un recuadro 1.

ES 2 300 429 T3

El conjunto de información correspondiente al nodo T está definido por el esquema de estructura siguiente:

```
5      <complexType name="T">
        <choice minOccurs="2" maxOccurs="4">
          <element ref="a" minOccurs="0" maxOccurs="1" />
          <element ref="b" minOccurs="1" maxOccurs="1" />
          <element name="c" type="tc"/>
        </choice>
10     </complexType>
```

15 Esto significa que el tipo complejo T comprende dos o tres casos de un grupo de elementos alternativos (tipo “choice”) que comprenden como máximo un elemento a, un elemento b y un elemento c de tipo tc. Esta estructura puede también representarse de una manera más compacta de la siguiente manera:

CHOICE[2,4](a[0,1], b[1,1], c[1,1])

20 Los campos que introducen los elementos a y b hacen referencia a una definición de estos elementos, que figura más adelante en el esquema de estructura del documento, del tipo:

```
25     <element name="a" type="ta"/>
     <element name="b" type="tb"/>
```

El esquema de estructura comprende a continuación la definición de los tipos ta, tb y tc que se definen de una manera análoga al tipo T. Puede comprender también instrucciones de sustitución de elementos, como la siguiente:

```
30     <element name="a1" type="ta1" substitutionGroup="a"/>
```

35 Esta instrucción indica que el elemento a1 de tipo ta1 puede sustituirse por un elemento a. En este caso, el tipo ta1 constituye un subtipo de ta. Asimismo, el tipo tb puede comprender un subtipo td. Tales subtipos se definen en un esquema de estructura de la manera siguiente, utilizando la etiqueta “restricción” o “extensión” prevista con para ello:

```
40     <complexType name="ta1">
        <restriction base="ta">
          ...
        </restriction>
      </complexType>

45     <complexType name="td">
        <restriction base="tb">
          ...
        </restriction>
      </complexType>
```

50 Según la norma XML-_Xpath, el segundo nodo b, unido al nodo T, está marcado de la manera siguiente:

.../T/b[1]

55 Esta notación referencia el primer nodo b unido al nodo T. Se observa que esta notación no es óptima desde el punto de vista del tamaño de la palabra binaria necesaria para su representación, y no permite tener en cuenta todas las especificidades autorizadas por el lenguaje XML-esquema, tales como el polimorfismo (posibilidad de definir subtipos de un tipo de elemento de información), o incluso la posibilidad de sustituir un elemento de un tipo por otro que tenga el mismo tipo o un subtipo del mismo tipo.

65 Según el procedimiento según la invención, se comienza en primer lugar por analizar el esquema de estructura del tipo complejo T del nodo de origen del segmento 2 que une el nodo T al nodo b, que se quiere referenciar. Este análisis tiene como objetivo construir una tabla que recoja todos los elementos susceptibles de pertenecer a la estructura del tipo complejo T y todos los tipos posibles de estos elementos. En el caso del tipo T, se obtiene la tabla siguiente:

TABLA 1

Elemento	Tipos posibles	Elementos de sustitución
a	t_a, t_{a1}	a1
a1	t_{a1}	Ninguno
b	t_b, t_d	Ninguno
c	t_c	Ninguno

Esta tabla indica, conforme a la definición del esquema en XML, que el elemento a puede sustituirse por el elemento a1. A partir de esta tabla, se determina la lista de todas las parejas (elemento, tipo) del tipo complejo T, estando dispuestas estas parejas en un orden predeterminado, por ejemplo por orden alfabético de nombres de elementos de información y de nombres de tipos de elementos de información. A continuación, a cada pareja, se asocia un código binario, obtenido por ejemplo numerándolas de manera secuencial en el orden en el que están dispuestas, lo que permite obtener la tabla siguiente:

TABLA 2

Código	Pareja (elemento, tipo)
000	(a, t_a)
001	(a, t_{a1})
010	(a1, t_{a1})
011	(b, t_b)
100	(b, t_d)
101	(c, t_c)
110	Reservado
111	Reservado

De una manera general, es necesario un código cada k bits para numerar objetos cuyo número está comprendido entre $2^{k-1}+1$ y 2^k . A la inversa, si N es el número de parejas, la codificación de estas parejas puede efectuarse cada E ($\log_2(N)$) bits (E(x) siendo la función “parte entera”). Los códigos no utilizados para la numeración pueden reservarse para realizar operaciones de verificación durante la descodificación del trayecto.

A continuación, se trata de definir el número M de elementos posibles contenidos en el nodo de origen del segmento. De una manera general, es necesario distinguir según se tenga relación con un grupo de elementos de tipo “secuencia” (grupo de elementos ordenados), de tipo “choice” (grupo de elementos alternativos) o de tipo “all” (grupo de elementos necesarios, ordenados o no) o con un elemento simple, pudiendo cada elemento por supuesto representar un grupo de elementos de nivel jerárquico inferior o un elemento simple.

Un grupo de elementos e1, e2,..., en de tipo “secuencia” (lista de elementos ordenados) puede representarse de la manera siguiente:

$$\text{SEQ}[\min_{seq}, \max_{seq}](e1[\min_{e1}, \max_{e1}], e2[\min_{e2}, \max_{e2}], \dots, en[\min_{en}, \max_{en}])$$

en la que “min_i” y “max_i” representan los números mínimo y máximo de casos del elemento e_i.

Si uno de los números de casos máximo max_i es indefinido o no delimitado, entonces el número máximo M de posiciones posibles de un grupo de este tipo no está delimitado. En el caso contrario, se obtiene con ayuda de la fórmula siguiente:

$$M = \max_{seq} \cdot \sum_{k=1}^n \max_{ek} \quad (1)$$

ES 2 300 429 T3

El número mínimo m de casos puede obtenerse con ayuda de la fórmula siguiente:

$$m = \min_{seq} \cdot \sum_{k=1}^n \min_{ek} \quad (2)$$

Un grupo de elementos de tipo CHOICE (grupo de elementos alternativos) puede representarse de la manera siguiente:

$$\text{CHOICE}[\min_{ch}, \max_{ch}](e1[\min_{e1}, \max_{e1}], e2[\min_{e2}, \max_{e2}], \dots, en[\min_{en}, \max_{en}])$$

Si uno de los números de casos máximo $\max_i$ es indefinido o no está delimitado, entonces el número máximo M de posiciones posibles de un grupo de este tipo no está delimitado. En el caso contrario, se obtiene con ayuda de la fórmula siguiente:

$$M = \max_{ch} \cdot \max_{k=1}^n (\max_{ek}) \quad (3)$$

$$M_j = (\max_{ek} - 1) \cdot \max_{k=1}^n (\max_{ek}) + \max_{ek}$$

siendo $\max()$ una función que da el valor máximo de todas los valores en parámetro.

Por lo que respecta al número de casos mínimo m de un grupo de tipo “choice”, éste viene dado por la fórmula siguiente:

$$m = \min_{ch} \cdot \min_{k=1}^n (\min_{ek}) \quad (4)$$

siendo $\min()$ una función que da el valor mínimo de todos los valores en parámetro.

Un grupo de elementos de tipo “all” (lista de elementos no ordenados) puede representarse de la manera siguiente:

$$\text{ALL}[\min_{all}, \max_{all}](e1[\min_{e1}, \max_{e1}], e2[\min_{e2}, \max_{e2}], \dots, en[\min_{en}, \max_{en}])$$

Los números de casos máximo M y mínimo m de un grupo de este tipo se obtienen con ayuda de las mismas fórmulas (1) y (2) que para un grupo de tipo SEQ.

En el caso de un elemento simple ek , los números de casos máximo M y mínimo m del elemento vienen dados directamente por el esquema de estructura del documento.

Si el número de elementos máximo M así obtenido está delimitado o es inferior a un determinado límite, por ejemplo 2^{16} , entonces la codificación de la posición de un elemento requiere $E(\log_2(M))$ bits. En el caso contrario, es necesario adoptar una codificación que pueda codificar cualquier número entero. Se puede así por ejemplo codificar un número tal por grupos de un número predefinido de bits, por ejemplo 5 bits, indicando el primer bit de un grupo si sí o no los cuatro bits siguientes son los últimos bits de codificación del número.

Si en el ejemplo anterior representado en la figura 1b, se quiere referenciar el segmento 2 que une el elemento T al tercer elemento (marcado por el recuadro 3), del nodo T, de nombre b y de tipo td. Teniendo en cuenta la Tabla 2, el número máximo de posiciones posibles aguas abajo del nodo T y la posición del nodo b (tercer nodo) entre estas posiciones posibles, el segmento 2 lleva el número:

“100 10”.

En efecto, el número de bits requerido para codificar seis elementos (véase la Tabla 2) es tres. Además, el número máximo de posiciones posibles aguas abajo del elemento T (en el recuadro 1) es 4, lo que necesita una codificación cada dos bits.

En el caso de un grupo de tipo SEQ, esta codificación puede ventajosamente optimizarse según dos métodos, sabiendo que cuando todos los elementos de una secuencia no son opcionales, su posición en el grupo está definida de una manera fija.

Según el primer método, se calculan los límites entre los que la posición de cada elemento e_i de la secuencia puede variar, lo que permite reducir el número de bits necesario para la codificación de la posición del elemento. Estos límites de posición $P_{\min}$ y $P_{\max}$ para un elemento e_i ($1 \leq i \leq n$, siendo n el número de elementos de la secuencia) pueden obtenerse con ayuda de las fórmulas siguientes:

$$P_{\min_i} = 1 + \sum_{k=1}^{i-1} \min_{ek} \quad (5)$$

$$P_{\max_i} = \sum_{k=1}^i \max_{ek} + (\max_{seq} - 1) \sum_{k=1}^n \max_{ek} \quad (6)$$

Según el segundo método, se calculan los valores de las posiciones posibles de cada elemento e_i de la secuencia, para cada caso j de la secuencia ($\min_{seq} \leq j \leq \max_{seq}$), con ayuda de las fórmulas siguientes:

$$P_{\min_{i,j}} = 1 + \sum_{k=1}^{i-1} \min_{ek} + (j-1) \sum_{k=1}^n \min_{ek} \quad (7)$$

$$P_{\max_{i,j}} = \sum_{k=1}^i \max_{ek} + (j-1) \sum_{k=1}^n \max_{ek} \quad (8)$$

La tabla siguiente se ha realizado para el grupo SEQ[1,3](a[1,1], b[1,1]). Esta tabla da para cada método de codificación, y cada elemento del grupo, los números de las posiciones posibles, y el número de bits necesario para la codificación de la posición del elemento.

TABLA 3

Elemento	Sin optimización		Método 1		Método 2	
a	1..6	3 bits	1..5	3 bits	1, 3, 5	2 bits
b	1..6	3 bits	2..6	3 bits	2, 4, 6	2 bits

Esta tabla muestra que el segundo método de optimización permite ganar un bit sobre el código de posición de un elemento de un grupo secuencia. Por otro lado, en el caso en que la posición de los nodos “hijos” relacionados con un nodo “padre” en una estructura, se define de manera que sólo autoriza una posibilidad, los métodos de optimización expuestos anteriormente de la codificación de la posición permiten conseguir una supresión completa de este código de posición en el código del segmento correspondiente. Se trata por ejemplo del caso de una secuencia de elementos en la que todos los elementos aparecen una sola vez:

SEQ[1, 1](e1[1,1], e2[1,1], ..., en[1,1])

En el caso de un grupo de tipo CHOICE, esta codificación puede también optimizarse calculando el límite máximo de la posición de cada elemento e_i del grupo. Este límite máximo $P_{\max}$ para un elemento e_i ($1 \leq i \leq n$, siendo n el número de elementos del grupo) puede obtenerse con ayuda de la fórmula siguiente:

$$P_{\max_i} = (\max_{ch} - 1) \cdot \max_{k=1}^n (\max_{ek}) + \max_{ej} \quad (9)$$

ES 2 300 429 T3

En la figura 2, la definición de un segmento de un trayecto en un árbol de esquema de estructura comprende un campo que contiene un código 12 de nodo, es decir un número de pareja (elemento, tipo), y un código 13 de posición del nodo de destino del segmento, con respecto a los otros nodos relacionados con el nodo de origen T del segmento, es decir con los otros elementos contenidos en el elemento.

Debe observarse que la codificación de la posición de un nodo se realiza independientemente del tipo del nodo. En la norma XML por el contrario, esta posición se marca con respecto al tipo del nodo. En el ejemplo .../T/b[1], b es el primer nodo b del nodo T, pero no necesariamente el primer elemento del nodo T.

Un trayecto 10 en un diagrama de árbol de esquema de estructura se define por tanto mediante una sucesión de segmentos 11, comprendiendo cada segmento al menos un código 12 de nodo y eventualmente un código 13 de posición.

A este respecto, en determinados casos, puede ser ventajoso retirar códigos 11 de segmento, los códigos 13 de posición de todos los nodos referenciados en un código 10 de trayecto, y colocados a parte en una zona prevista para ello en el código del trayecto.

Es necesario a continuación insertar un código 14' delimitador que marca el fin de la secuencia de segmentos que define un trayecto en la estructura del documento, y por tanto el principio de la información codificada relativa al elemento del documento, referenciada por el trayecto.

Por otro lado, el lenguaje XML permite asociar atributos a los diferentes elementos de información de un documento. En este contexto, si se quiere además poder definir un trayecto hasta un atributo de un elemento, se prevé asociar cada código 11 de segmento a un código 14 de tipo de segmento (figura 2) que permite distinguir si el objeto de destino del segmento es otro elemento denominado elemento "hijo" del nodo de origen del segmento o un atributo del nodo de origen.

Al igual que anteriormente, el código 11 de un segmento entre un elemento de información y un atributo de este elemento comprende un código de atributo obtenido numerando todos los atributos posibles del elemento. Por el contrario, como los atributos de un elemento no están ordenados, no es necesario prever un campo de posición en el código de segmento entre un elemento y un atributo.

Ventajosamente, los códigos de segmento hacia un elemento o hacia un atributo de elemento se definen por la tabla siguiente:

TABLA 4

Código	Significado
00	ir al padre
01	ir al índice de los atributos
10	ir al índice de los elementos
11	Indicador de fin de trayecto

En el ejemplo anterior (figuras 1a, 1b), el segmento entre el elemento T y el tercer elemento b está completamente definido por el código siguiente:

"10 100 10"

Según la invención tal como se ilustra en la figura 2, un trayecto en un diagrama de árbol está por tanto constituido por una sucesión de códigos 11 de segmentos tales como los definidos anteriormente, que se termina por un código 14' de tipo fin de trayecto es decir "11" según la Tabla 4.

Además, en la Tabla 4, el código "00" permite definir la posición de un elemento en un documento estructurado de manera relativa con respecto a un elemento tratado anteriormente. Permite así introducir un código de segmento de otro elemento unido al nodo de origen del segmento anterior o de un atributo de este nodo. Este código puede también ir seguido de otros códigos idénticos para remontar varios nodos en el diagrama de árbol del esquema de estructura del documento.

La figura 3 muestra un organigrama que ilustra el tratamiento realizado por un ordenador programado para realizar la codificación de trayecto, según la invención. En esta figura, el tratamiento de codificación comprende una etapa previa de análisis de la estructura del documento para determinar, para cada uno de los elementos de información

de la estructura, el contenido de la Tabla 2, la lista de los atributos del elemento y el número máximo de elementos “hijo” comprendidos en el elemento. A partir del trayecto que debe codificarse, que puede presentarse en forma de un trayecto XML, tal como se mencionó anteriormente, el ordenador de codificación según la invención ejecuta la etapa 21 que consiste en leer el nombre del elemento de origen del primer segmento del trayecto que debe codificarse. En la etapa 22, el ordenador de codificación determina si el objeto de destino del segmento en curso es un atributo o un elemento de información. En la etapa 23, el ordenador de codificación inserta en el código 10 de trayecto que debe determinarse, el código 14 de tipo de segmento que equivale a “01” o a “10”, según si el objeto de destino del segmento en curso es un atributo o un elemento. El ordenador de codificación ejecuta a continuación la etapa 24 de inserción del código de atributo o código 12 de pareja (elemento, tipo) leído en el Tabla 2 correspondiente al elemento de origen del segmento que está codificándose. Si el objeto de destino del segmento en curso es un atributo, el tratamiento de codificación se termina.

Si el objeto de destino es un elemento de información, el ordenador de codificación determina a partir del trayecto que debe codificarse, la posición del elemento de destino del segmento en curso, y determina el código binario de esta posición en función del número máximo de elementos unidos al elemento de origen del segmento. En la etapa 26, inserta el código 13 de posición así determinado en el código del trayecto, a continuación del código 12 de la pareja (elemento, tipo).

Si en la etapa 27, el trayecto que debe codificarse comprende otro segmento, el ordenador de codificación ejecuta las etapas 21 a 27 en el segmento siguiente, es decir considerando que el nodo de origen del segmento que debe codificarse es el nodo de destino del segmento anteriormente codificado. En el caso contrario, inserta el código 14’ de tipo de segmento “11” marcando el fin del código del trayecto (etapa 28).

Como se mencionó anteriormente, el trayecto que debe codificarse puede definirse de manera relativa, con respecto a un elemento de información de destino de un trayecto codificado anteriormente. En este caso, el nuevo trayecto que debe codificarse en modo relativo comprende en primer lugar uno o varios códigos de tipo de segmento iguales a “00”, indicando el número de estos códigos cuántos niveles es necesario remontar en la estructura jerárquica del esquema de estructura para llegar al nodo que debe referenciarse por el nuevo trayecto que debe codificarse.

La figura 4 muestra un organigrama que ilustra el tratamiento realizado por un ordenador programado para descodificar trayectos, según la invención. Un ordenador de este tipo realiza igualmente un análisis previo del esquema de estructura del documento para obtener, para cada elemento de información de la estructura, la Tabla 2, un índice de atributos y el número máximo de elementos “hijo” comprendidos en el elemento. En la etapa 31, el ordenador de descodificación lee los dos primeros bits del trayecto 10 codificado, dando un código 14 de tipo de segmento tal como se define por la Tabla 4. Si el código de segmento equivale a “10” indicando que el objeto siguiente en el trayecto es un elemento de información (etapas 32 a 34), el ordenador de descodificación lee en la etapa 38 la Tabla 2 correspondiente al primer elemento, para determinar sobre cuántos bits se codifican las parejas (elemento, tipo) del elemento. En el caso de un trayecto absoluto, el primer elemento es el elemento raíz de la estructura del documento. En la etapa 39, lee en el código de trayecto el código 12 del primer elemento sobre el número de bits así determinado, y determina con ayuda del código leído y de la Tabla 2 correspondiente al primer elemento, el nombre y el tipo del elemento que corresponde al elemento de destino del primer segmento. A partir del número máximo de elementos “hijo” contenidos en el primer elemento, determina en la etapa 40 el número de bits que deben leerse a continuación en el código 10 de trayecto que debe descodificarse y lee (etapa 41) el código 13 de posición del elemento en el código de trayecto sobre el número de bits así determinados. A continuación, el ordenador de descodificación ejecuta las etapas 31 a 41 para el código 11 de segmento siguiente en el código 10 de trayecto que debe descodificarse, convirtiéndose el nodo de destino del segmento anteriormente descodificado en el nodo de origen del nuevo segmento que debe descodificarse.

Si en el curso de las etapas 32 a 34, el código 14 de tipo de segmento leído en el código de trayecto que debe descodificarse equivale a “01”, el objeto de destino del segmento en curso de descodificación es un atributo del elemento en curso. En este caso, el ordenador de descodificación lee el índice de los atributos del elemento en curso para determinar sobre cuántos bits se codifica el número de atributo en el código de trayecto (etapa 36), y lee el número de bits así determinado en el código de trayecto para obtener el número de atributo (etapa 37), lo que permite determinar el atributo de destino del segmento en curso utilizando el índice de los atributos del elemento en curso. El tratamiento de descodificación del trayecto se termina entonces.

Si en el curso de las etapas 32 a 34, el código 14 de segmento leído en el código de trayecto que debe descodificarse equivale a “11”, la descodificación del código de trayecto también se termina. Si el código de segmento equivale a “00”, esto significa que el trayecto que debe descodificarse se ha codificado en modo relativo y que es necesario remontar el elemento de información de origen del segmento que acaba de descodificarse (etapa 35). Si este código aparece de nuevo, el ordenador de descodificación remonta más de un nivel en el diagrama de árbol para situarse al nivel del nodo situado por encima del nodo en curso. En otras palabras, cada vez que aparece el código “00”, el segmento que debe descodificarse siguiente tiene como elemento de información de destino el nodo de origen del segmento que debe descodificarse anterior.

El código 14’ de fin de trayecto del código 10 de trayecto marca el principio de la información codificada contenida en el elemento de información de destino del último segmento así descodificado.

ES 2 300 429 T3

Se puede prever además un código particular situado al principio de un código 10 de trayecto para indicar si el trayecto que sigue está codificado en modo relativo o en modo absoluto. Si está en modo absoluto, el elemento de información del primer segmento es el nodo raíz de la estructura en árbol del documento. Si el trayecto está codificado en modo relativo, el ordenador de descodificación se posiciona sobre el elemento “padre” del elemento en curso.

5

10

15

20

25

30

35

40

45

50

55

60

65

REIVINDICACIONES

1. Procedimiento de codificación de un trayecto en una estructura jerárquica de documento estructurado, definida por un esquema de estructura de documento, estando este trayecto definido por una sucesión de segmentos, uniéndose cada segmento un nodo de origen y un nodo de destino, representando cada nodo un elemento de información del documento, estando cada elemento de información asociado en el esquema de estructura a al menos un tipo de información, **caracterizado** porque comprende:
 - una fase previa que comprende una etapa que consiste en asociar a cada nodo en cuestión del esquema de estructura, una lista de parejas constituidas por un nombre y un tipo de elemento de información, representadas por todos los nodos susceptibles de relacionarse directamente con el nodo en cuestión, y en asociar un código binario respectivo a cada pareja de nombre y tipo de elemento de información, y
 - una fase de codificación del trayecto que comprende una etapa que consiste, para cada segmento del trayecto que debe codificarse, en determinar e insertar en el código de trayecto un código (12) binario de nodo asociado a la pareja nombre - tipo del nodo de destino del segmento.
2. Procedimiento de codificación según la reivindicación 1, **caracterizado** porque la fase de codificación de trayecto comprende además una etapa que consiste en determinar un código (13) binario de posición del nodo de destino del segmento, con respecto a los otros nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.
3. Procedimiento de codificación según la reivindicación 1 o 2, **caracterizado** porque la fase de codificación de trayecto comprende además una etapa que consiste en generar un código (10) de trayecto que comprende una sucesión de códigos (11) de segmentos, comprendiendo cada código de segmento un código (12) binario de nodo del nodo de destino del segmento, y un código (13) binario de posición del nodo de destino del segmento.
4. Procedimiento de codificación según la reivindicación 1 o 2, **caracterizado** porque la fase de codificación de trayecto comprende además una etapa que consiste en generar un código (10) de trayecto que comprende una sucesión de códigos (11) de segmentos, comprendiendo cada código de segmento un código (12) binario de nodo del nodo de destino del segmento, y una secuencia de códigos (13) de posición que dan la posición de todos los nodos referenciados en la sucesión de código de segmentos.
5. Procedimiento de codificación según una de las reivindicaciones 1 a 4, **caracterizado** porque la fase previa comprende además una etapa que consiste en determinar un número máximo de nodos susceptibles de relacionarse directamente con el nodo en cuestión, para determinar el tamaño del código (13) binario de posición de nodo.
6. Procedimiento de codificación según una de las reivindicaciones 1 a 5, **caracterizado** porque al menos uno de los elementos de información de la estructura del documento comprende atributos, teniendo el trayecto que debe codificarse como elemento de destino un atributo, comprendiendo la fase de codificación además una etapa de inserción en el código (11) de cada segmento de un código (14) de tipo de segmento que indica si el nodo de destino del segmento es un atributo o un elemento de información.
7. Procedimiento de codificación según una de las reivindicaciones 1 a 6, **caracterizado** porque la fase de codificación comprende además una etapa de inserción en el código (10) de trayecto de un código (14') de fin de trayecto.
8. Procedimiento de codificación según la reivindicación 7, **caracterizado** porque el código (14') de fin de trayecto es un código (14) de tipo de segmento que tiene un valor predefinido.
9. Procedimiento de codificación según una de las reivindicaciones 6 a 8, **caracterizado** porque el nodo de origen de cada segmento está situado en un nivel jerárquico superior al del nodo de destino en el esquema de estructura del documento, y porque la fase de codificación comprende además una etapa de inserción en el código de trayecto de al menos un código (14) de tipo de segmento que tiene un valor predefinido que indica que el nodo de origen del segmento que debe codificarse siguiente es el nodo de destino del segmento que debe codificarse anterior.
10. Procedimiento de codificación según una de las reivindicaciones 1 a 9, **caracterizado** porque la fase de codificación comprende además una etapa de inserción en el código (10) de trayecto de un código que indica si el trayecto codificado es un trayecto absoluto desde el nodo raíz del documento, o un trayecto relativo desde un nodo cualquiera del esquema de estructura del documento.
11. Procedimiento de decodificación de un código (10) de trayecto en una estructura jerárquica de documento estructurado, definida por un esquema de estructura de documento, comprendiendo este código de trayecto una sucesión de códigos (11) de segmentos, uniéndose cada segmento un nodo de origen a un nodo de destino que constituye el nodo de origen del segmento siguiente, representando cada nodo un elemento de información del documento, estando cada elemento de información asociado en el esquema de estructura a al menos un tipo de información, **caracterizado** porque cada segmento está definido en el código (10) de trayecto al menos por un código (12) binario de nodo que

ES 2 300 429 T3

representa una pareja nombre - tipo, constituida por un nombre y un tipo de elemento de información del elemento de información representado por el nodo de destino del segmento, comprendiendo el procedimiento:

5 - una fase previa que consiste en asociar a cada nodo en cuestión del esquema de estructura, una lista de parejas nombre - tipo de elemento de información constituidas por un nombre y un tipo de elementos de información, representadas por todos los nodos susceptibles de relacionarse directamente con el nodo en cuestión, y en asociar un código binario respectivo a cada pareja nombre - tipo de elemento de información, y

10 - una fase de decodificación del código de trayecto que consiste, para el segmento del código (10) de trayecto que debe decodificarse, en decodificar el código (12) de nodo que representa la pareja nombre, tipo del nodo de destino del código de segmento con ayuda de la lista de parejas nombre - tipo del nodo de destino.

15 12. Procedimiento de decodificación según la reivindicación 11, **caracterizado** porque cada segmento comprende además, en el código (10) de trayecto que debe decodificarse, un código (13) de posición del nodo de destino con respecto a los otros nodos susceptibles de relacionarse directamente con el nodo de origen del segmento, comprendiendo la fase de decodificación además una etapa que consiste, para cada segmento, en decodificar el código (13) binario de posición del nodo de destino del segmento, en función de las posiciones respectivas de todos los nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.

20 13. Procedimiento de decodificación según la reivindicación 11 o 12, **caracterizado** porque la decodificación del código (12) binario de nodo que representa la pareja nombre-tipo de elemento de información comprende una etapa de determinación del tamaño en número de bits de este código y de búsqueda de este código en la lista de parejas nombre - tipo del nodo de origen del segmento.

25 14. Procedimiento de decodificación según una de las reivindicaciones 11 a 13, **caracterizado** porque el decodificación del código (13) binario de posición del nodo de destino del segmento comprende la determinación del tamaño en número de bits de este código en función del número máximo de nodos susceptibles de relacionarse directamente con el nodo de origen del segmento.

30 15. Procedimiento de decodificación según una de las reivindicaciones 11 a 14, **caracterizado** porque cada código (11) de segmento comprende un código (14) de tipo de segmento, comprendiendo la fase de decodificación del trayecto además para cada segmento, la decodificación del código de tipo de segmento.

35 16. Procedimiento de decodificación según la reivindicación 15, **caracterizado** porque el código (14) de tipo de segmento de cada código (11) de segmento del código (10) de trayecto permite determinar si el segmento tiene como nodo de destino un elemento de información o un atributo del nodo de origen del segmento.

40 17. Procedimiento de decodificación según la reivindicación 15 o 16, **caracterizado** porque comprende la determinación del fin del código de trayecto, que está marcado por un código (14') de tipo de segmento que tiene un primer valor predefinido.

45 18. Procedimiento de decodificación según la reivindicación 15 o 17, **caracterizado** porque si el código (14) de tipo de segmento tiene un segundo valor predefinido, el código (11) de segmento que debe decodificarse siguiente en el código (10) de trayecto tiene como nodo de destino el nodo de origen del segmento que debe decodificarse anterior.

50

55

60

65

70

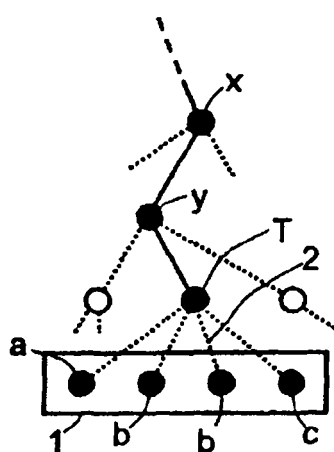


Fig. 1a

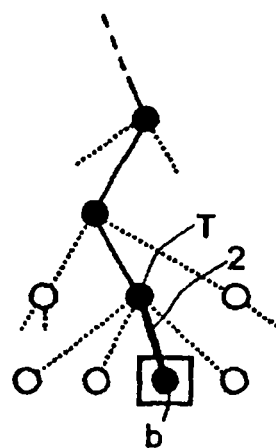


Fig. 1b

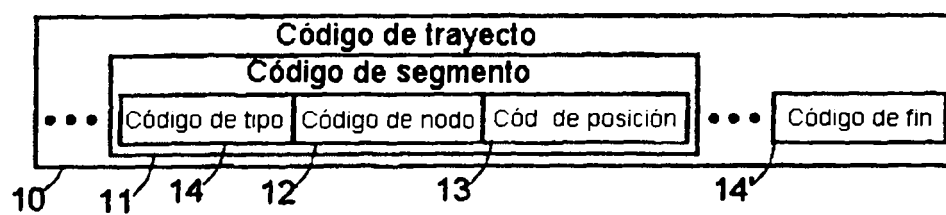


Fig. 2

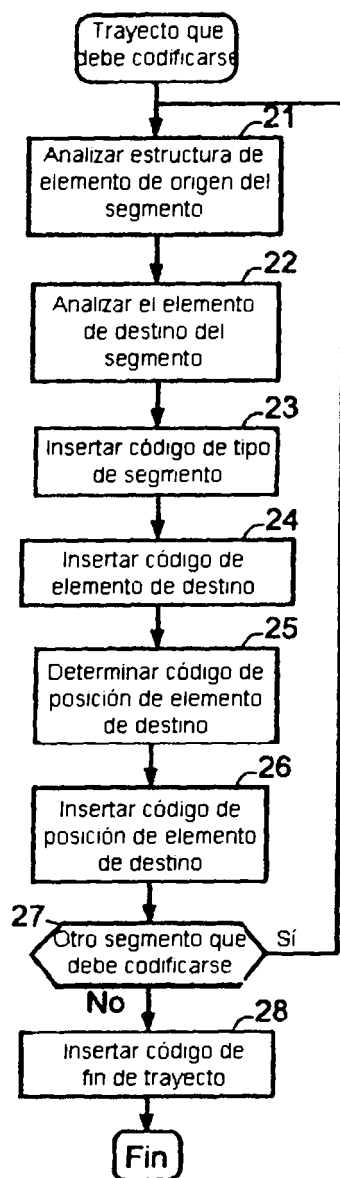


Fig. 3

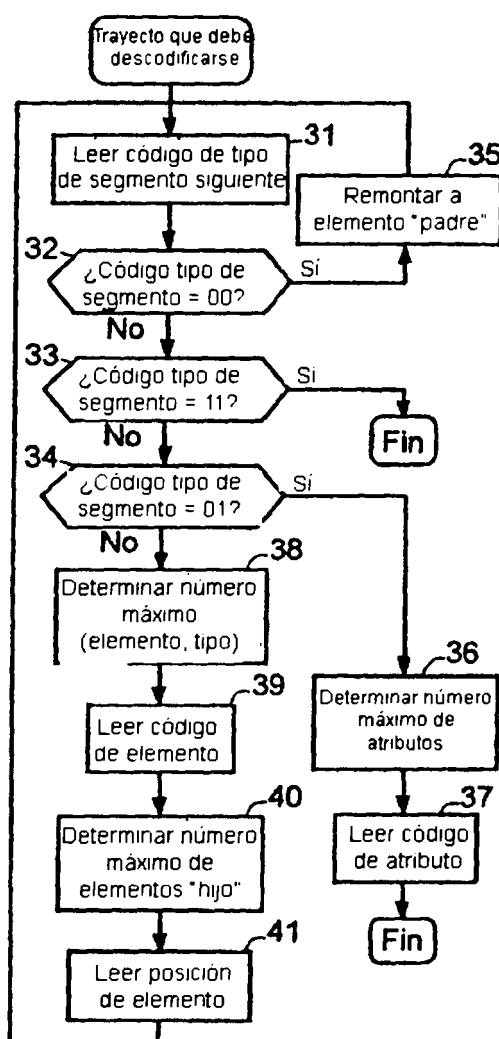


Fig. 4