República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e do Comércio Exterior
Instituto Nacional da Propriedade Industrial

(11) **PI 0316335-0 B1**

(22) Data do Depósito: 30/10/2003

(45) Data de Concessão: 10/02/2016

(RPI 2353)

(54) Título: PROCESSO PARA A GERAÇÃO DE UMA CORRENTE DE BITS A PARTIR DE UMA ÁRVORE DE INDEXAÇÃO

(51) Int.Cl.: H03M 7/00; G11B 27/10

(30) Prioridade Unionista: 15/11/2002 DE 102 53 275.3, 18/08/2003 DE 103 37 825.1

(73) Titular(es): SIEMENS AKTIENGESELLSCHAFT

(72) Inventor(es): HARALD KOSCH, JÖRG HEUER, ANDREAS HUTTER, ANDREA KOFLER-VOGT

Relatório Descritivo da Patente de Invenção para **"PROCESSO PARA A GERAÇÃO DE UMA CORRENTE DE BITS A PARTIR DE UMA ÁRVORE DE INDEXAÇÃO".**

A presente invenção refere-se a um processo para a geração de uma corrente de bits a partir de uma árvore de indexação, bem como a um processo de codificação e um processo de decodificação, que utilizam o processo para a geração da corrente de bits. Além disso, a invenção refere-se a um dispositivo de codificação e decodificação.

Freqüentemente, é necessário que determinados conteúdos de uma corrente de bits sejam lidos segundo uma busca efetuada por um usuário, respectivamente pesquisar, quanto a determinados conteúdos, se esses conteúdos estão contidos na corrente de bits. Nesse caso, uma solicitação definida por um usuário pode ser formulada por meio de uma linguagem de busca, como SQL (ver referência [1]) ou XPATH (ver referência [2]). Nesse caso, é vantajoso quando não for preciso pesquisar toda a corrente de bits à procura dos conteúdos desejados, senão quando as informações estiverem depositadas em uma lista de indexação ou em uma árvore de indexação, de tal modo que seja preciso pesquisar somente a parte da corrente de bits na qual esteja armazenada a árvore de indexação, respectivamente a lista de indexação.

O problema da leitura de dados a partir de uma corrente de bits ocorre, por exemplo, em um documento elaborado com ajuda da linguagem XML (XML = extensible markup language), o qual é representado no formato MPEG7-BiM. No que se refere ao formato MPEG7-BiM de um documento XML toma-se aqui como referência o documento [3]. No caso de uma representação desse tipo, a corrente de bits gerada é dividida em várias unidades (access units), as quais, por sua vez, são constituídas por muitos fragmentos (fragment update units). As unidades são codificadas e, em caso de necessidade, são enviadas como corrente de MPEG7-BiM a um ou mais receptores.

No que se refere à busca de informações a partir de documentos XML, já são conhecidas várias linguagens de busca que permitem buscar determinadas informações no documento. Por exemplo, nesse ponto cabe

remeter-se à linguagem de busca XPATH já mencionada (ver referência [2]). Por meio da linguagem de busca XPATH pode-se definir critérios de seleção para a filtragem de informações desejadas dentro de um documento XML. O objetivo de uma busca, nesse caso, pode ser, por um lado, a avaliação de se uma unidade da corrente de bits é importante para o receptor. Por outro lado, com a ajuda de uma busca pode-se recorrer planejadamente a informações desejadas no documento XML. Até agora, na geração da corrente de bits de um documento XML, o processo de codificação MPEG7 ainda não previu nenhum mecanismo que possibilite um acesso de livre escolha a determinados elementos do documento XML. Por isso, a corrente de bits MPEG7 tem que ser decodificada para a busca por elementos. Obtém-se, então, novamente, um documento no formato XML, o qual pode ser pesquisado por meio da linguagem de busca XPATH. A decodificação e o subseqüente processamento de um documento XML para a busca por informações específicas, conseqüentemente, demanda muito tempo e é inaceitável para determinadas aplicações críticas quanto ao tempo. Além disso, pode surgir o problema de que a memória no decodificador seja limitada, de tal modo que a corrente de bits não seja decodificada totalmente. Além disso, o esforço da decodificação teria sido desnecessário, caso a busca XPATH, executada no documento XML decodificado, tenha decorrido negativamente.

No âmbito de TV-Anytime (TVA), que se acha inscrito no documento [4], emprega-se uma estrutura de índex que permita um acesso de livre escolha a determinados elementos de um fragmento de dados. A estrutura de índex consiste em várias partes e abrange uma assim chamada lista de Key-Index, na qual são depositados todos os caminhos indexados de um documento. No caso de uma busca, esses caminhos são comparados, segundo a sequência, com a busca até que seja encontrado um determinado registro na lista de Key-Index. Por meio das informações que são armazenadas na lista de Key-Index para esse registro é possível determinar os locais em uma corrente descritiva nos quais encontra-se codificado o registro indexado. Por meio do emprego da lista de Key-Index não é mais necessário decodificar fragmentos de dados desinteressantes, de tal modo que durante

uma busca necessita-se de menos espaço de memória. No entanto, migração linear da lista de Key-Index gasta muito tempo e a transmissão de todos os caminhos indexados demanda muito esforço.

Por isso, a presente invenção pretende criar um processo para a geração de uma corrente de bits a partir de uma árvore de indexação, no qual a busca por dados indexados seja possível de modo fácil e eficiente.

Esse objetivo é alcançado de acordo com as características das reivindicações independentes. Desdobramentos da invenção resultam das reivindicações dependentes.

A árvore de indexação, a partir da qual uma corrente de bits é gerada por meio do processo de acordo com a invenção, abrange uma multiplicidade de planos hierárquicos, sendo que a cada plano hierárquico acha-se combinado um ou mais nós de índex e os nós de índex contêm dados de índex que se acham classificados na árvore de indexação de acordo com um ou mais critérios predeterminados. No processo de acordo com a invenção, os dados de índex dos nós de índex são inseridos na corrente de bits e, além disso, para um nó de índex insere-se respectivamente a informação na corrente de bits, em cujo local na corrente de bits se localizam os dados de índex de um ou mais nós de índex do plano hierárquico, que se localiza abaixo do plano de hierarquia do respectivo nó. Devido à armazenagem da informação adicional quanto aos nós de índex em um plano hierárquico situado abaixo, torna-se mais fácil uma busca por determinados dados de índex, pois desse modo tornam-se possíveis saltos para os nós de índex que são relevantes para a busca. Desse modo, assegura-se uma pesquisa e uma busca essencialmente mais eficiente por dados de índex.

Em uma forma preferida de execução, a árvore de indexação acha-se estruturada como uma assim chamada B-Tree (= Balanced Tree), a qual assegura uma distribuição equilibrada dos dados pelos nós da árvore. Uma descrição exata da B-Tree encontra-se na referência [5].

Os dados de índex na árvore de indexação podem ser classificados por quaisquer critérios, como, por exemplo, de modo lexicográfico.

Em uma forma de execução particularmente preferida, os dados de

índex acham-se inseridos na corrente de bits segundo a ordem Depth-First. Devido ao emprego da ordem Depth-Firts, os dados de índex na árvore de indexação são inseridos inicialmente segundo a profundidade na corrente de dados, e devido a isso as informações relevantes para uma pesquisa são dispostas na corrente de bits uma ao lado da outra e salta-se por cima eficientemente das informações não relevantes. Uma descrição exata da ordem Depth-First encontra-se na referência [6].

Em uma outra forma de execução preferida da invenção, os dados de índex abrangem caminhos de uma árvore de estrutura de documento, a qual compreende um nó de raiz e uma multiplicidade de nós de folhas. Além disso, os dados de índex abrangem, de preferência, as características de valor dos caminhos e as posições das características de valor no documento, que se acha representado pela árvore de estrutura de documento. Além disso, em uma forma preferida de execução, os dados de índex também abrangem a quantidade dos caminhos em um nó de índex.

Nesse caso, os caminhos podem ser caminhos absolutos que começam dos nós de raiz da árvore de estrutura de documento e conduzem a um nó de folhas da árvore de estrutura de documento. No entanto, em uma forma de execução da invenção particularmente preferida, os caminhos são caminhos relativos, sendo que um caminho relativo de um respectivo nó de índice é um caminho relativamente a um caminho, anteriormente inserido na corrente de bits, do respectivo nó de índex ou de um nó de índex de um plano hierárquico situado sobre o plano hierárquico do respectivo nó de índex. Por meio do emprego de caminhos relativos aproveita-se aquilo que há em comum nos caminhos, pois os caminhos de nós adjacentes possuem, em sua maioria, uma fração em comum. Desse modo, pode-se reduzir a necessidade de memória requerida para os dados de índex na corrente de bits. Uma outra redução da necessidade de memória pode ser obtida na medida em que os caminhos do nó de índex, cujos dados de índex são inseridos como primeiros de um plano hierárquico na corrente de bits, sejam inseridos na corrente de bits em uma seqüência inversa à seqüência na qual os dados de índex ficam dispostos no nó de índex. Nesse caso, leva-se em conta o

fato de que os dados de índex ao final do primeiro nó de índex de um plano hierárquico apresentam uma semelhança maior em relação ao nó de índex do plano hierárquico imediatamente superior do que os dados de índex ao início do primeiro nó de índex. Conseqüentemente, em determinados casos resulta uma codificação particularmente efetiva por meio de caminhos relativos.

Em uma forma de execução particularmente preferida da invenção, os caminhos abrangem elementos descritivos de um documento XML (XML = extensible markup language), sendo que os caminhos são particularmente caminhos XPATH do documento XML.

Em uma outra forma de execução da invenção, os dados de índex estão codificados binariamente por meio de um processo de codificação, especialmente por meio de um processo de codificação MPEG. Em uma forma de execução particularmente preferida, o processo de codificação empregado é o processo de codificação MPEG7.

Além do processo acima descrito para a geração de uma corrente de bits, a invenção abrange ainda um processo para a codificação de uma estrutura de dados, no qual os elementos de dados da estrutura de dados ficam indexados em uma árvore de indexação, sendo que uma corrente de bits é gerada de acordo com o processo de acordo com a invenção e essa corrente de bits é parte da corrente de dados codificada. Além disso, a invenção abrange um processo para a decodificação de uma estrutura de dados, sendo que o processo é configurado de tal forma que seja decodificada uma estrutura de dados codificada por meio do processo de codificação acima descrito.

Além disso, a invenção abrange um processo para a codificação e decodificação de uma estrutura de dados, o qual abrange o processo de codificação e o processo de decodificação acima descritos.

A invenção abrange também um dispositivo de codificação, com o qual pode-se executar o processo de codificação de acordo com a invenção, bem como um dispositivo de decodificação com o qual se pode executar o processo de decodificação de acordo com a invenção. Além disso, a invenção refere-se a um dispositivo correspondente para a codificação e decodifi-

cação, com o qual se pode executar o processo combinado de codificação e decodificação acima descrito.

Exemplos de execução da invenção serão representados e explicados a seguir com base nos desenhos.

Mostra-se:

Fig. 1: um exemplo de uma árvore de indexação empregada no processo de acordo com a invenção;

Fig. 2: um recorte da árvore de indexação mostrada na figura 1;

Fig. 3: um recorte de uma corrente de bits, que foi gerada a partir da árvore de indexação da figura 1 por meio do processo de acordo com a invenção;

Fig. 4: uma árvore de indexação simplificada, na qual estão ordenados lexicograficamente os dados de índex, os quais representam caminhos de uma árvore de estrutura de documento;

Fig. 5: a árvore de indexação da figura 4, sendo que os caminhos do segundo plano hierárquico acham-se representados parcialmente como caminhos relativos;

Fig. 6: a árvore de indexação da figura 4, sendo que todos os caminhos do segundo plano hierárquico acham-se representados como caminhos relativos; e

Fig. 7: uma variação da árvore de indexação segundo a figura 4.

A figura 1 mostra, como exemplo, uma árvore de indexação empregada no processo de acordo com a invenção. No caso dessa árvore, trata-se de uma assim chamada árvore B (B significa Balanced), que é freqüentemente empregada na indexação de dados. A estrutura de uma árvore B acha-se descrita, por exemplo, na referência [5] já mencionada. A árvore B apresenta uma estrutura equilibrada e possibilita assim uma busca apenas com um esforço logarítmico medido pela quantidade de registros, enquanto que no caso de busca com uma lista é necessário um esforço linear.

A árvore de indexação contém uma série de nós de 1 a 10, sendo que em cada nó acha-se armazenado um ou mais dados de índex sob a forma de assim chamadas chaves. No presente caso, os dados de índex são

assim chamados caminhos XPATH de uma árvore de descrição de um documento XML. Uma descrição de caminhos XPATH encontra-se na referência [2] já mencionada. Os caminhos XPATH levam dos nós de raiz da árvore de descrição do documento XML para cada um dos nós de folhas da árvore de descrição do documento XML. Na árvore de descrição empregada nessa forma de execução, emprega-se como nós de folhas os elementos XML com conteúdo simples e todos os atributos XML. Conseqüentemente, os dados de índex da árvore de indexação mostrada na figura 1 abrangem caminhos raiz-folha de uma árvore de descrição XML.

Os caminhos XPATH estão representados na árvore de indexação da figura 1 como uma série de letras grandes e pequenas estão ordenados lexicograficamente, sendo que cada letra corresponde a um nome de elemento na árvore de descrição XML. Por exemplo, o caminho MDMVUFACN no nó 1 significa MPEG7/Description/MultimediaContent/Video/UsageInformation/FinancialResults/AccountItem/Costtype/Name. Além do caminho, nos nós também estão armazenadas as características de valor, nas quais esse caminho ingressa no documento XML. No presente caso, o caminho MDMVUFACN no nó de índex 1 entra duas vezes com as características "Total for Production" e "Broadcast". Diferentemente deste, o caminho MDMVUAFAc, por exemplo, no nó de índex 2 entra apenas uma vez com a característica "EUR".

A numeração dos nós de índex na árvore de indexação da figura 1 corresponde à seqüência em que os dados de índex são inseridos na corrente de bits com a ajuda do processo de acordo com a invenção. Nesse caso, a seqüência corresponde a uma assim chamada ordem Depth-First, que se acha descrita, por exemplo, na referência [6] já mencionada. Por meio da inserção dos dados de índex na corrente de bits segundo a ordem Depth-First, obtém-se uma menor complexidade no acesso aos dados de índex, pois na busca por dados de índex salta-se por cima das informações não relevantes.

Na figura 2 mostra-se um recorte de uma árvore de indexação da figura 1, no qual, por meio de um exemplo, explica-se a pesquisa de dados de

índex nessa árvore. Nesse caso, trata-se da pesquisa para saber se o caminho XPATH MPEG7/Description/Multimediacontent/Video/UsageInformation/Availability/Dissemination/Disseminator/Agent/Name contém a característica de valor "Discovery" (resumidamente, MDMVUADDAN = "Discovery". Já que a linha MDMVUADDAN, do ponto de vista lexicográfico, é menor do que o caminho XPATH "MDMVUFACN" do nó de índex 1, então cabe remeter-se ao nó de índex 2. A linha de busca também é menor lexicograficamente do que o registro "MDMVUAFAC" do nó de índex 2. Por isso, passa-se para o nó de índex 3 localizado à esquerda do nó de índex 2. Esse nó de índex contém dois caminhos XPATH, "MDMVUAAt" e "MDMVUADFh". A linha de busca localiza-se entre esses dois caminhos XPATH, de tal modo que, no plano hierárquico seguinte, passa-se para o nó de índex 5 localizado entre os nós de índex 4 e 6. Esse nó de índex contém agora, como primeiro elemento, o caminho XPATH buscado, "MDMVUADDAN", bem como a característica de valor buscada "Discovery". Com isso descobriu-se no nó de índex 5 a característica de valor adequada e obtém-se, então, a posição do elemento correspondente no documento XML indexado. Devido ao fato de que os nós de índex são inseridos na corrente de bits conforme a ordem Depth-First, então todos os demais nós da árvore de indexação são ignorados ou pulados durante a busca, de tal modo que a busca por elementos na árvore de indexação apresente pouca complexidade.

A seguir será tratada detalhadamente a inserção dos dados de índex na corrente de bits por meio do processo de acordo com a invenção. Isso é feito com base na figura 3, a qual representa um recorte de uma corrente de bits gerada por meio do processo de acordo com a invenção. Na corrente de bits mostrada na figura 3 são expostos, para melhor compreensão, os números dos nós de índex, sendo que, no entanto, esses números não são remetidos junto com a corrente. No detalhe são mostrados, na figura 3, os conteúdos dos nós de índex 1 e 3. Os nós contêm como primeiro elemento inicialmente a quantidade dos registros, isto é, a quantidade dos caminhos XPATH armazenados no nó. Nesse caso, para o nó de índex 1

resulta o número 1 e para o nó de índex 3 resulta o número 2. Em seguida, para cada entrada segue-se a chave, isto é, o respectivo caminho XPATH, bem como a quantidade de instâncias, isto é, a quantidade das características de valor que esse caminho XPATH tem no documento XML. Para o nó de índex 1 resulta, por exemplo, para o caminho XPATH, "MDMVUAFCN", a quantidade de instâncias 2, pois o caminho com as duas características de valor "Total for Production" e "Broadcast" aparece no documento XML. Além disso, para cada característica de valor, a posição dessa característica de valor no documento XML é inserida na corrente de bits. No processo de acordo com a invenção, é essencial que, além dessas informações, também seja inserida na corrente de bits a informação de em que posição na corrente de bits – particularmente em qual posição em relação à posição momentânea (offset) – encontram-se os dados de índex, do plano hierárquico inferior, de nós de índex ligados com o respectivo nó de índex. Desse modo, torna-se possível passar por cima de nós que não sejam relevantes para uma pesquisa de busca predeterminada.

A seguir, o nó que se bifurca de um nó para um plano hierárquico inferior é chamado de nó filho. No processo de acordo com a invenção é preciso notar que a informação quanto à posição do primeiro nó filho de um plano hierárquico não precisa ser armazenada, pois esse nó filho na corrente de bits é lido como sendo o próximo. Como se pode deduzir da figura 3, desse modo, para o nó de índex 1 armazena-se apenas a posição do nó de índex 10 (chamado de "offset (10)") e para o nó de índex 3 armazena-se apenas as posições dos nós de índex 5 e 6 (chamados de "offset (5)" e "offset (6)"). Resultando, por exemplo, que a informação buscada em uma pesquisa seja lexicograficamente maior do que o XPATH no nó 1, então, salta-se imediatamente para o nó 10, de tal modo que não seja mais necessário buscar o nó de índex 2 e o nó de índex que se bifurca do nó de índex 2. Desse modo torna-se possível uma busca efetiva.

Na tabela que se segue expõe-se novamente quais informações por nó são armazenadas na corrente de bits.

Tabela

Nó {
Quantidade dos registros
para cada registro:
chaves
quantidade das instâncias
para cada instância:
característica de valor
posição
para cada nó filho, exceto o primeiro
offset na corrente

A corrente de bits gerada com processo de acordo com a invenção é remetida ao receptor e pode ser lida correspondentemente a uma pesquisa que, por exemplo, tenha sido explicada em relação à figura 2. Por meio de comparações entre o XPATH a ser pesquisado e os registros nos nós de índex, determinadas informações de nós são lidas a partir da corrente até que o correspondente XPATH buscado com a característica de valor buscada tenha sido encontrado ou até que não seja mais possível nenhum caminho XPATH adequado devido à classificação. Neste último caso, a informação desejada não está contida no documento indexado.

Outras formas de execução do processo de acordo com a invenção serão explicadas a seguir, com base nas figuras de 4 a 7, que apresentam árvores de indexação simplificadas com registros ordenados lexicograficamente. Na figura 4 mostra-se uma árvore de indexação a ser codificada por meio do processo de acordo com a invenção, árvore esta que abrange nós de K1 a K4. Nesse caso, o nó K1 é chamado de nó pai e os nós K2-K4, ramificados do nó K1, são chamados de nós filhos. No caso dos registros, trata-se de caminhos de uma árvore de estrutura de documento. Em vez de transmitir na corrente de bits os caminhos com seus comprimentos totais, existe a possibilidade de armazenar os assim chamados caminhos relativos no nó de índex a ser transmitido, sendo que esses caminhos são relativos em relação a um caminho do nó pai transmitido anteriormente, respectiva-

mente em relação ao caminho transmitido anteriormente do nó de índex momentaneamente a ser transmitido.

A inserção de caminhos relativos na corrente de bits é apresentada na figura 5. Nesse caso, a representação "." significa um passo para trás no caminho a que se refere o caminho relativo. No nó K1, para o registro "AE" transmite-se o caminho relativo "..E" em relação ao registro anterior ACB. Nos nós filhos K1, K2 e K3, respectivamente no segundo e no terceiro registro são transmitidos os caminhos relativos em relação ao primeiro caminho do respectivo nó filho.

Na forma de execução da figura 5, para os primeiros caminhos AA, ACC, respectivamente AEF, dos nós filhos K2, K3, respectivamente K4, transmite-se o caminho absoluto completo. Também é possível adicionalmente transmitir os primeiros caminhos dos nós filhos como sendo caminhos relativos em relação a um caminho do nó pai K1. Isso é apresentado na figura 6. Em vez dos caminhos AA e ACC do primeiro e do segundo nó filho, nesse caso são transmitidos os caminhos relativos "..A" e ".C" em relação ao primeiro caminho do nó pai. Em vez do caminho "AEF" do terceiro nó filho, transmite-se o caminho relativo "F" em relação ao segundo caminho do nó pai. Nesse caso, os caminhos, de preferência em sua representação em Byte, são depositados na corrente de bits.

Devido ao emprego de caminhos relativos, reduz-se bastante o volume de dados na corrente, pois ao se armazenar os caminhos aproveita-se aquilo que há em comum nos caminhos, de tal modo que se necessita de menos espaço de memória. Particularmente, no caso de caminhos relativos, as respectivas frações em comum dos caminhos são remetidas apenas uma vez.

Em uma outra forma de execução da invenção obtém-se uma economia de dados na medida em que na disposição dos registros dos nós de índex para os primeiros nós filhos K2 é empregada uma seqüência inversa à da figura 4. Desse modo, consegue-se que o registro que apresenta a maior semelhança com o nó pai K1 anterior fique ao início do nó filho K2. Desse modo, no caso de configurações especiais da árvore de indexação

obtêm-se caminhos relativos essencialmente reduzidos e uma menor necessidade de memória.

Os caminhos podem ser os caminhos XPATH, mencionados anteriormente, de uma árvore de descrição XML. Além disso, os caminhos podem ser codificados binariamente, como, por exemplo, com um processo de codificação MPEG7. Particularmente, para a codificação binária podem ser empregados os processos especificados na referência [3] já mencionada. O emprego de caminhos codificados binariamente pode economizar na quantidade dos bits a serem transmitidos, em comparação com representações textuais.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] http://dx1.hrz.unidortmund.de:8001/doc1/hrz/sqlref/sqloracle.html

[2] http://www.w3.org/TR/xpath

[3] ISSO/IEC 15938-1 Multimedia Content Description Interface – Part 1: Systems, Genebra 2002.

[4] TV-Anytime Specifications Series S-3 on Metadata, Part-B, versão 13.

[5] http://www.public.asu.edu/~peterjn/btree/

[6] http://www.generation5.org/simple_search.shtml

REIVINDICAÇÕES

1. Processo para a geração de uma corrente de bits a partir de uma árvore de indexação, sendo que a árvore de indexação abrange uma multiplicidade de planos hierárquicos e a cada plano hierárquico acha-se combinado um ou mais nó de índex (K1, K2, K3, K4), sendo que os nós de índex (K1, K2, K3, K4) contêm dados de índex que estão classificados na árvore de indexação segundo um ou mais critérios predeterminados, no qual:

- os dados de índex dos nós de índex (K1, K2, K3, K4) são inseridos na corrente de bits;

- para um nó de índex, insere-se respectivamente a informação na corrente de bits, naquele ponto na corrente de bits em que se encontram os dados de índex de um ou mais nós de índex (K1, K2, K3, K4) do plano hierárquico que se localiza abaixo do plano hierárquico do respectivo nó.

2. Processo de acordo com a reivindicação 1, no qual a árvore de indexação é uma B-Tree (=Balanced Tree).

3. Processo de acordo com a reivindicação 1 ou 2, no qual os dados de índex na árvore de indexação estão classificados lexicograficamente.

4. Processo de acordo com uma das reivindicações anteriores, no qual os dados de índex são inseridos na corrente de bits segundo a ordem Depth-First.

5. Processo de acordo com uma das reivindicações anteriores, no qual os dados de índex abrangem caminhos de uma árvore de estrutura de documento constituída por pelo menos um nó de raiz e uma multiplicidade de nós de folhas.

6. Processo de acordo com a reivindicação 5, no qual os dados de índex contêm as características de valor dos caminhos e as posições das características de valor no documento que se acha representado pela árvore de estrutura de documento.

7. Processo de acordo com a reivindicação 5 ou 6, no qual os dados de índice abrangem a quantidade dos caminhos em um nó de índex

(K1, K2, K3, K4).

8. Processo de acordo com uma das reivindicações de 5 a 7, no qual os caminhos abrangem caminhos absolutos, que começam no nó de raiz e levam a um nó de folhas.

9. Processo de acordo com uma das reivindicações de 5 a 8, no qual os caminhos abrangem caminhos relativos, sendo que um caminho relativo de um respectivo nó de índex (K1, K2, K3, K4) é um caminho relativo a um caminho, anteriormente inserido na corrente de bits, do respectivo nó de índex (K1, K2, K3, K4) ou de um nó de índex (K1, K2, K3, K4) de um plano hierárquico localizado acima do plano hierárquico do respectivo nó de índex (K1, K2, K3, K4).

10. Processo de acordo com a reivindicação 9, no qual os caminhos do nó de índex, cujos dados de índex são inseridos na corrente de bits como sendo os primeiros de um plano hierárquico, são inseridos na corrente de bits em uma seqüência inversa à seqüência na qual os dados de índex ficam dispostos nos nós de índex (K1, K2, K3, K4).

11. Processo de acordo com uma das reivindicações de 5 a 10, no qual os caminhos abrangem elementos de descrição de um documento XML (XML = Extensible Markup Language).

12. Processo de acordo com a reivindicação 11, no qual os caminhos são caminhos XPATH do documento XML.

13. Processo de acordo com uma das reivindicações anteriores, no qual os dados de índex acham-se codificados binariamente por meio de um processo de codificação, particularmente por meio de um processo de codificação MPEG.

14. Processo de acordo com a reivindicação 13, no qual o processo de codificação é um processo de codificação MPEG7.

15. Processo para a codificação de uma estrutura de dados, no qual os elementos de dados da estrutura de dados estão indexados em uma árvore de indexação, sendo que é gerada uma corrente de bits de acordo com uma das reivindicações anteriores e a corrente de bits é parte da corrente de dados codificados.

16. Processo para a decodificação de uma estrutura de dados, sendo que o processo é configurado de tal modo que a estrutura de dados codificada de acordo com a reivindicação 12 é decodificada.

17. Processo para a codificação e decodificação de uma estrutura de dados abrangendo o processo de acordo com a reivindicação 15 e o processo de acordo com a reivindicação 16.

18. Dispositivo de codificação, com qual se pode executar o processo como definido na reivindicação 15.

19. Dispositivo de decodificação, com o qual se pode executar um processo como definido na reivindicação 16.

20. Dispositivo para a codificação e decodificação de uma estrutura de dados, com o qual se pode executar um processo como definido na reivindicação 17.

FIG 1

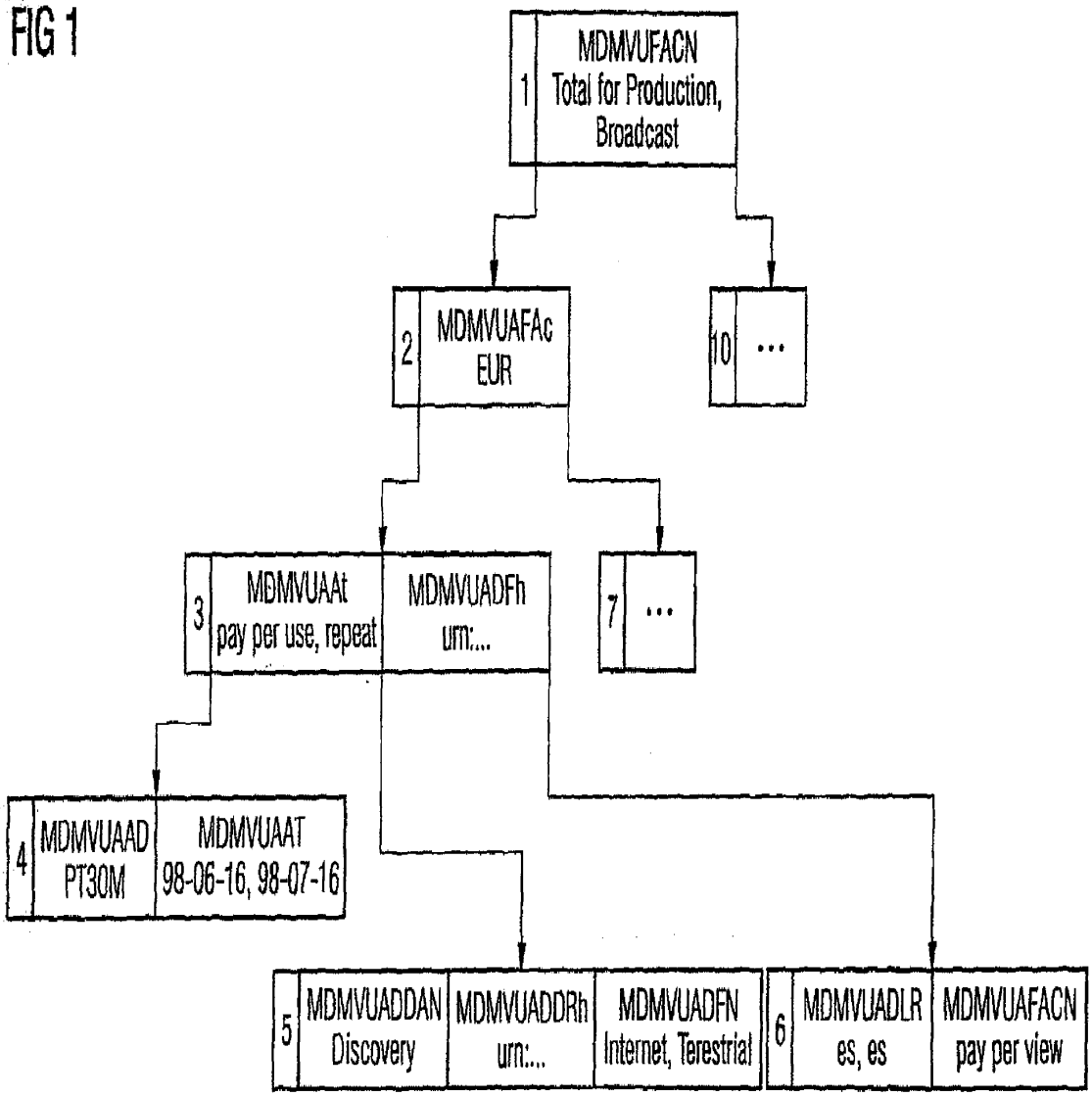

1
MDMVUFACN
Total for Production, Broadcast
2
MDMVUAFAc
EUR
10
...
3
MDMVUAAt
pay per use, repeat
MDMVUADFh
urn:...
7
...
4
MDMVUAAD
PT30M
MDMVUAAT
98-06-16, 98-07-16
5
MDMVUADDAN
Discovery
MDMVUADDRh
urn:...
MDMVUADFN
Internet, Terestrial
6
MDMVUADLR
es, es
MDMVUAFACN
pay per view

FIG 2

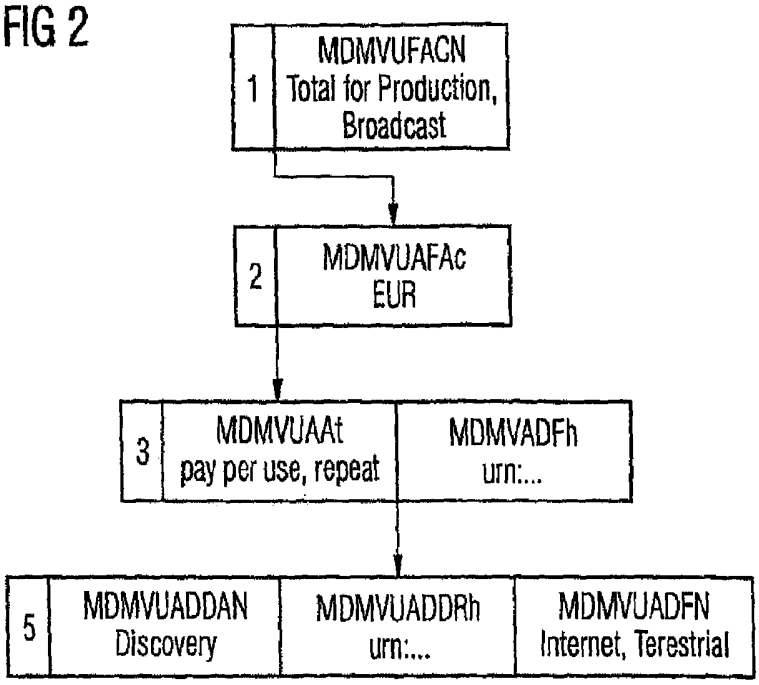


FIG 3

1	1, MDMVUAFCN, 2, Total for Production, position, Broadcast, postion, offset (10)	2	...
3	2, MDMVUAAt, 2, pay per use, position, repeat, position, MDMVUADFh, 1, urn:..., position, offset (5), offset (6)	4	...

FIG 4

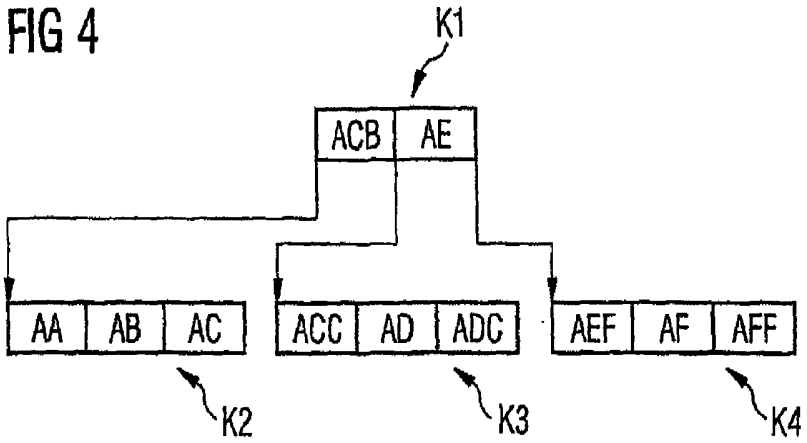

K1
ACB
AE
AA
AB
AC
ACC
AD
ADC
AEF
AF
AFF
K2
K3
K4

FIG 5

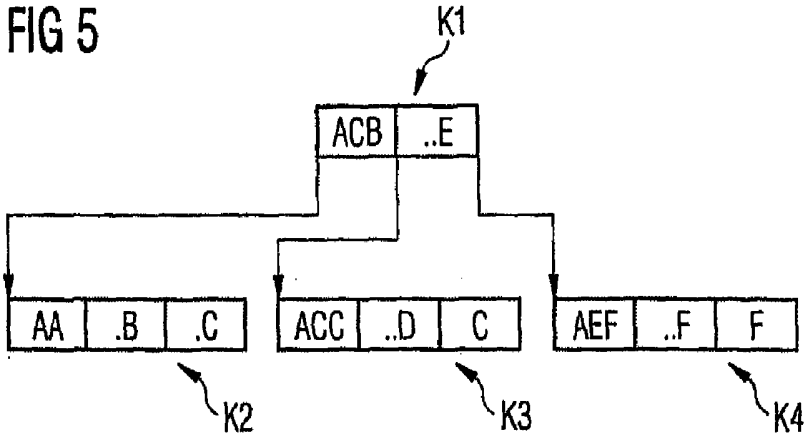

K1
ACB
..E
AA
.B
.C
ACC
..D
C
AEF
..F
F
K2
K3
K4

FIG 6

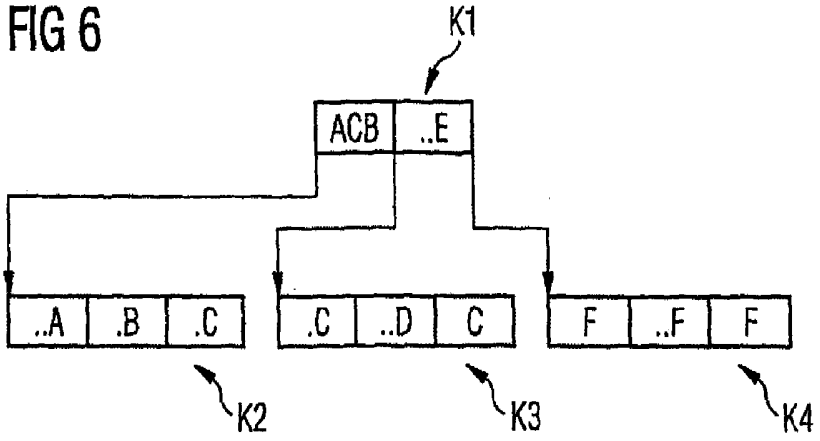
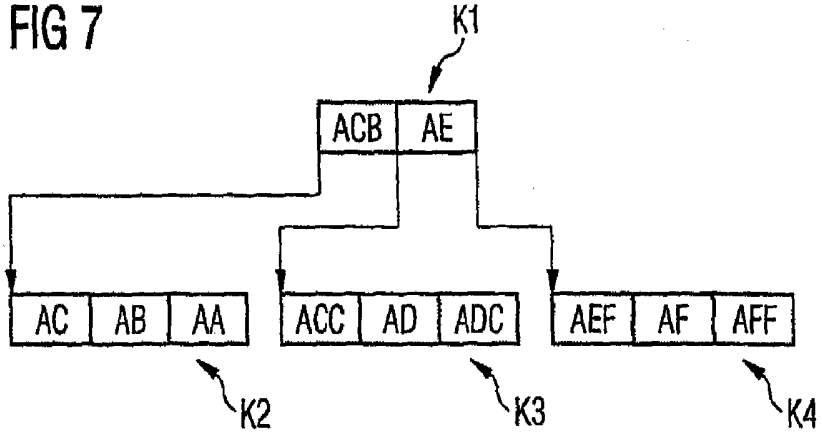

K1
ACB
..E
..A
.B
.C
.C
..D
C
F
..F
F
K2
K3
K4

FIG 7


K1
ACB
AE
AC
AB
AA
ACC
AD
ADC
AEF
AF
AFF
K2
K3
K4

RESUMO

Patente de Invenção: **"PROCESSO PARA A GERAÇÃO DE UMA CORRENTE DE BITS A PARTIR DE UMA ÁRVORE DE INDEXAÇÃO".**

A presente invenção refere-se a um processo para a geração de uma corrente de bits a partir de uma árvore de indexação, sendo que a árvore de indexação abrange uma multiplicidade de planos hierárquicos e a cada plano hierárquico acha-se combinado um ou mais nó de índex, sendo que os nós de índex contêm dados de índex que estão classificados na árvore de indexação segundo um ou mais critérios predeterminados. No processo, os dados de índex dos nós de índex são inseridos na corrente de bits, e para um nó de índex, insere-se respectivamente a informação na corrente de bits, naquele ponto na corrente de bits em que se encontram os dados de índex de um ou mais nós de índex do plano hierárquico que se localiza abaixo do plano hierárquico do respectivo nó.