

**(12) FASCÍCULO DE PATENTE DE INVENÇÃO**

(22) Data de pedido: <b>2003.07.16</b>	(73) Titular(es): <b>SAMSUNG ELECTRONICS CO., LTD.</b> <b>416 MAETAN-DONG, YEONGTONG-GU SUWON-</b> <b>SI GYEONGGI-DO</b> KR
(30) Prioridade(s): <b>2002.07.23 KR 2002043097</b> <b>2002.10.15 KR 2002062913</b>	
(43) Data de publicação do pedido: <b>2005.03.16</b>	(72) Inventor(es): <b>HYOSEOP SHIN</b> KR
(45) Data e BPI da concessão: <b>2007.11.07</b> <b>137/2007</b>	(74) Mandatário: <b>ANTÓNIO JOÃO COIMBRA DA CUNHA FERREIRA</b> <b>R DAS FLORES 74 4 AND 1249-235 LISBOA</b> PT

(54) Epígrafe: **MÉTODO PARA FORNECIMENTO DE ÍNDICES DE METADADOS**

(57) Resumo:  
MÉTODO PARA FORNECIMENTO DE ÍNDICES DE METADADOS



## DESCRIÇÃO

### **"Método para fornecimento de índices de metadados"**

#### CAMPO TÉCNICO

O presente invento refere-se a uma estrutura de índices de metadados fornecida para pesquisa de informação sobre conteúdos e um método para fornecimento de índices dos metadados e um método e um aparelho para pesquisa dos metadados através da utilização da estrutura de índices dos metadados. De forma mais particular, o presente invento refere-se a uma estrutura de índices de metadados que contém informação sobre uma chave, pelo menos, uma parte da qual é cifrada para permitir que informação sobre conteúdo seja pesquisada de forma mais eficiente, quando os metadados XML para os conteúdos digitais definidos no fórum TV-Anytime (daqui em diante referido como "TVA") (daqui em diante referido como "metadados TVA") são divididos em fragmentos numa unidade independente e transmitidos numa base de fragmentos, um método para proporcionar índices dos metadados e um método e um aparelho para pesquisar os metadados através da utilização dos índices dos metadados.

#### ANTECEDENTES DA TÉCNICA

O Fórum "TV-Anytime" é uma organização de padronização privada criada em Setembro de 1999 com a finalidade de desenvolver padrões para proporcionar serviços relacionados com o audiovisual num ambiente amigável do utilizador tal como um gravador digital pessoal (PDR) que tem um dispositivo de armazenagem pessoal de volume alto. Especificamente, o objectivo dos serviços é possibilitar que todos os utilizadores visionem e ouçam vários tipos de programas (tais como serviços de radiodifusão convencionais, serviços interactivos através de linha e outros semelhantes) numa altura desejada e de uma maneira desejada, apoiadas no dispositivo de armazenagem pessoal.

O Fórum "TV-Anytime" tinha feito funcionar Grupos de Trabalho para modelos de negócio, referência de sistema/ligação de transmissão/conteúdos, descrições,

metadados, protecção e administração de direitos e coisas semelhantes, de modo a estabelecer a padronização. Em relação aos metadados a que respeita o presente invento, foi publicado o "1<sup>st</sup> Draft of Metadata Specification SP003v1.3" em Junho de 2002.

Uma configuração do PDR será descrita de resumidamente com referência à FIG. 1. O PDR 100 recebe sinais e metadados vídeo/áudio através de uma variedade de redes tais como ondas rádio, ondas de satélite, redes Internet e semelhantes, provenientes de um fornecedor 200 para fornecer sinais vídeo/áudio, recebe padrões de visualização e audição e as preferências pessoais de utilizadores, se necessário, e transmite os mesmos para o fornecedor 200 para fornecimento dos sinais vídeo/áudio. O PDR 100 compreende um dispositivo de armazenagem de grande volume para armazenar os sinais e os metadados vídeo/áudio recebidos. O PDR 100 também compreende suporte lógico para armazenagem e reprodução dos sinais vídeo/áudio e uma aplicação de guia electrónico de programas (EPG) para recuperar e exibir metadados para os sinais vídeo/áudio. O utilizador verifica os metadados para os dados vídeo/áudio, isto é, títulos dos programas, tempos de reprodução de programas e semelhantes, através de um ecrã guia de grelha da aplicação EPG mostrada na FIG. 2, selecciona um programa desejado e recebe o mesmo através da rede em tempo real ou reproduz os dados vídeo/áudio armazenados previamente no dispositivo de armazenagem de grande volume.

Os metadados referem-se a dados que descrevem conteúdos tais como títulos e sinopses de programas e são definidos como "dados sobre dados". Nas especificações de metadados TVA do fórum TV-Anytime, a sua estrutura é definida pela utilização de linguagem de esquema XML (ver XML 1.0 de W3C), a norma pelo W3C (um consórcio para promover normas para XML) e as semânticas e atributos dos respectivos elementos de metadados também são definidos. Os metadados TVA relevantes para difundir conteúdos são configurados com um documento XML que tem um nó de raiz, "TVAMain (300)" como mostrado na FIG. 3. Os metadados TVA relevantes para programas são configurados com, por exemplo, nós tais como a "Tabela ProgramInformation", "Tabela GroupInformation", "Tabela

ProgramLocation”, “Tabela SegmentInformation” e outras semelhantes, sob o nó de “ProgramDescription”.

No fórum TV-Anytime, os metadados TVA são transmitidos numa base de fragmentos como uma unidade independente, de modo a transmitir um grande volume de metadados TVA num formato de sequência. O conceito de fragmentos será descrito de forma breve com referência à FIG. 4. Os fragmentos são obtidos pela divisão dos metadados TVA configurados com os documentos XML mostrados na FIG. 3 em estruturas de árvore predeterminadas. Por exemplo, onde todos os metadados TVA são divididos numa estrutura de árvore (fragmento de rede TVA) que inclui um nó superior de “TVAMain” e nós subordinados predeterminados sob este nó superior, uma estrutura de árvore (fragmento ProgramInformation) que inclui um nó superior de “Tabela ProgramInformation” e nós subordinados sob este nó superior, uma estrutura de árvore (fragmento BroadcastEvent) que inclui um nó superior de “BroadcastEventInformation” e nós subordinados sob este nó superior, cada uma das estruturas de árvore dividida torna-se um fragmento. Os fragmentos podem ser transmitidos de forma independente dos outros fragmentos e os fragmentos podem ser acedidos de forma individual.

Para acesso individual aos fragmentos, é necessário conhecer um nó referenciado por um fragmento de metadados TVA transmitido, isto é, um nó corresponde ao nó superior do fragmento de metadados TVA, em toda a estrutura de árvore de metadados, e para descrever trajectos relativos nos fragmentos de metadados TVA de chaves contidas no fragmento de metadados TVA transmitido. Para esta finalidade, é utilizado XPath, que é uma sintaxe para descrever um trajecto para um ou mais nós num documento XML definido pelo W3C. O termo ‘chave’ refere-se a um campo específico dos metadados utilizado para indexar e também significa nós subordinados de um nó referenciado por um fragmento. Campos (para condições de pesquisa) introduzidos pelo utilizador, tais como ‘Service ID’ e ‘Published Time’, correspondem às chaves.

A fim de proporcionar pesquisa eficiente e acesso a fragmentos, uma estrutura de índices para as chaves incluídas nos fragmentos de metadados também é necessária e informação

sobre a estrutura de índices, isto é, informação de índice, também é transmitida de forma independente dos fragmentos de metadados.

Sob o ambiente fornecido pelo fórum TV-Anytime, se um utilizador deseja recuperar informação sobre um programa que satisfaça uma condição de 'Published Time' predeterminada, a informação de índice transmitida para o mesmo de forma independente dos fragmentos é utilizada para identificar a localização (identificador) de um fragmento de metadados que satisfaça uma condição de 'Published Time' desejada e um acesso ao fragmento de metadados relevante é, então, concretizado com base na localização (identificador), de modo a extrair metadados que satisfaçam a condição de Hora publicada.

A especificação TV-Anytime TV145, J.P. Evain, "1st Draft of Metadata Specification SP003v1.3", 17ª reunião do fórum TV-Anytime, Montreal, Canada, Junho de 2002; daqui em diante, referido como "Referência de técnica de índice de chave" propõe uma estrutura de sequência de dados de índice de chave para um índice de fragmentos de metadados.

A noção de um contentor definido pelo fórum TV-Anytime será descrita antes da descrição da estrutura de índices.

O fórum TV-Anytime define um contentor como um armazém de nível de topo para o qual são transmitidos todos os dados, que cobrem a informação de índice referida atrás e os fragmentos de metadados, o que é designado por um tipo de transmissão de nível superior. Descrevendo de forma breve o contentor, cada contentor compreende uma pluralidade de secções, cada uma armazenando a informação de índice ou os fragmentos de metadados. O contentor pode ser classificado num contentor de índice e num contentor de dados de acordo com a informação assim transportada: o contentor de índice transporta secções de informação de índice tais como uma secção de lista de índice de chave (`key_index_list`), uma secção de índice de chave (`key_index`), uma secção de índice de subchave (`sub_key_index`), uma secção de repositório de sequência de caracteres (`string-repository`) e uma secção de repositório de dados de fragmentos (`fragment_data_repository`), visto que um

contentor de dados transporta secções de fragmentos de metadados tais como uma secção de tabela de elementos (elements\_table), uma secção de repositório de sequência de caracteres (string-repository) e uma secção de repositório de dados de fragmentos (fragment\_data\_repository). A classificação acima é dada com base nos conteúdos da informação incluída nos contentores. Tanto o contentor de índice como o contentor de dados têm configuração idêntica.

Referindo o contentor definido pelo fórum TV-Anytime como ilustrado na FIG. 5, o contentor compreende um campo de dados identificador do contentor (container\_id) (não mostrado) e um grande número de secções. Em cada secção, os conteúdos armazenados no 'section\_body' são idênticos de acordo com um valor cifrado no 'section\_id'. Por exemplo, uma secção 10 cujo valor cifrado no 'section\_id' é '0X0004' é identificada como uma secção de lista de índice de chave (key\_index\_list), uma secção 20 cujo valor cifrado em 'section\_id' é '0X0005' é identificada como uma secção de índice de chave (key\_index), uma secção 30 cujo valor cifrado no 'section\_id' é '0X0006' é identificada como uma secção de índice de subchave (sub\_key\_index), uma secção 40 cujo valor cifrado no 'section\_id' é '0X0001' é identificada como uma secção de tabela de elementos (elements\_table) e uma secção 50 cujo valor cifrado no 'section\_id' é '0X0003' é identificada como uma secção de repositório de dados de fragmentos (fragment\_data\_repository).

Os fragmentos de metadados TVA são armazenados na secção 50 de repositório de dados de fragmentos (fragment\_data\_repository) do contentor de dados e depois transmitidos. A informação de identificador (handle\_value) para os fragmentos de metadados TVA no contentor de dados é incluída na secção de tabela de elementos 40 do contentor de dados.

Em conclusão, o fragmento de metadados TVA é unicamente identificado pela informação de identificador de contentor (container\_id) e pela informação de identificador de fragmento de metadados (handle\_value) do contentor que inclui o fragmento de metadados TVA.

A referência da técnica de índice de chave descrita acima propõe a estrutura de índice de chave para indexar os fragmentos de metadados TVA, armazenados no contentor de dados acima referido, isto é, uma estrutura composta pela secção de lista de índice de chave (`key_index_list`) 10, pela secção de índice de chave (`key_index`) 20 e a pela secção de índice de subchave (`sub_key_index`) 30. Uma vez que a sintaxe da estrutura é descrita em detalhe na referência da técnica de índice de chave descrita acima, a descrição detalhada da mesma será omitida. Daqui em diante, a estrutura será descrita com referência à FIG. 6, que ilustra a estrutura por segmentos da informação de índice.

A secção de lista de índice de chave (`key_index_list`) 10 definida na estrutura de índice de chave fornece uma lista de todas as chaves transmitidas. A lista inclui a informação de chave que define cada informação de chave e de identificação sobre a secção de índice de chave (`key_index`) 20 a ser descrita posteriormente. A informação de chave compreende (1) informação de localização do fragmento de metadados relevante para a chave e (2) informação de localização da chave no fragmento de metadados. A informação de localização do fragmento de metadados é expressa em XPath (`fragment_xpath_ptr`) no TVA. A informação de localização da chave é expressa em XPath (`key_xpath_ptr`) para o trajecto relativo no fragmento relevante dos nós utilizados como a chave no TVA.

O XPath do fragmento de metadados é um trajecto para nó de raiz do documento XML de metadados TVA, isto é, um trajecto absoluto e o XPath dos nós utilizados como as chaves, isto é, o XPath das chaves, representa um trajecto relativo da chave para o fragmento de metadados relevante. O XPath para o fragmento de metadados e o XPath para a chave são armazenados num segmento '`fragment_xpath_ptr`' 11 e num segmento '`key_xpath_ptr`' 12, respectivamente.

Além disso, a secção de lista de índice de chave (`key_index_list`) 10 inclui a informação de identificação sobre a secção de índice de chave (`key_index`) 20 de cada chave a ser descrita posteriormente (isto é, a informação de identificador de contentor (`container_id`) do contentor que

armazena a secção de índice de chave (key\_index) 20 e a informação de identificador de índice de chave). A informação de identificador de contentor e a informação de identificador de índice de chave são armazenadas num segmento 'index\_container' da secção de lista de índice de chave (key\_index\_list) 10 e num segmento 'key\_index\_identifier', respectivamente e depois transmitidas.

A secção de índice de chave (key\_index) 20 definida na estrutura de índice de chave fornece uma lista de informação que representa as gamas de valores da chave incluída na respectiva secção de índice de subchave (sub\_key\_index) 30, isto é, o valor mais elevado da chave de entre os valores da chave na respectiva gama (daqui em diante referido como um 'valor de chave representativo') e informação de identificação na secção de índice de subchave (sub\_key\_index) 30 relevante para cada valor de chave representativo (isto é, a informação de identificador de contentor (container\_id) do contentor que armazena a secção de índice de subchave (sub\_key\_index) e a informação de identificador de índices de subchave).

Por conseguinte, a secção de índice de chave (key\_index) 20 inclui um segmento 'key\_index\_identifier' para armazenar a informação de identificador de índice de chave definida na secção de lista de índice de chave (key\_index\_list) 10, segmentos 'high\_key\_value' 13 para armazenar os valores de chave representativos das respectivas gamas de valores da chave incluída na secção de índice de subchave (sub\_key\_index) 30 e segmentos 'sub\_index\_container' e segmentos 'sub\_index\_identifier' para a informação de identificação sobre a secção de índice de subchave (sub\_key\_index) 30 (isto é, para a informação de identificador de contentor (container\_id) do contentor onde a secção de índice de subchave (sub\_key\_index) 30 está armazenada e a respectiva informação de identificador de índices de subchave). A secção de índice de subchave (sub\_key\_index) 30 definida na estrutura de índice de chave fornece uma lista dos valores da chave. A lista também inclui informação de identificação sobre os fragmentos de metadados que correspondem aos valores da chave (isto é, a informação de identificador de contentor (container\_id) dos contentores

que armazenam os fragmentos de metadados e a informação de identificador (`handle_value`) dos fragmentos de metadados).

Por conseguinte, a secção de índice de subchave (`sub_key_index`) 30 inclui um segmento `'sub_index_identifier'` para armazenar a informação de identificador de índice de subchave definida na secção de índice de chave (`key_index`) 20, segmentos `'key_value'` 14 para armazenar as respectivas gamas de valores da chave, segmentos `'target_container'` para armazenar a informação de identificador de contentor respectiva (`container_id`) dos contentores onde os fragmentos de metadados são armazenados e segmentos `'target_handle'` para armazenar nos mesmos a informação de identificador de dados de fragmento respectiva (`handle_value`). A estrutura de índice de chave pode ser compreendida de forma mais fácil com referência à FIG. 7 que ilustra a informação de índice.

A FIG. 7 mostra a secção de lista de índice de chave (`key_index_list`) que inclui chaves relevantes para o Service ID, a `'Published Time'` e a `'Published Duration'`. O nó superior do fragmento de metadados que inclui as chaves relevantes para o Service ID, a `'Published Time'` e a `'Published Duration'` é o `'BroadcastEvent'` 310 como mostrado na FIG. 3, identificado pelo bloco sombreado. Em consequência, o XPath `'/TVAMain/ProgramDescription/ProgramLocation Table/BroadcastEvent'` para o fragmento `'BroadcastEvent'` é armazenado no segmento `'fragment_xpath_ptr'` 11a e os XPaths para as chaves do `'Service ID'` a `'Published Time'` e a `'Published Duration'` para o fragmento `'BroadcastEvent'`, isto é, `'@ServiceId'` (311a na FIG. 3), `'EventDescription/ PublishedTime'` (311b na FIG. 3) e `'EventDescription/PublishedDuration'` (311c na FIG. 3) são armazenados no segmento `'key_xpath_ptr'` 12a.

A estrutura de índices será mais compreensível com referência à FIG. 7 que ilustra a informação de índice.

A FIG. 7 mostra a secção de lista de índice de chave (`key_index_list`) que inclui chaves para `'Service ID'`, `'Published Time'` e `'Published Duration'`, em que um nó superior dos metadados relativos ao Service ID, à `'Published Time'` e à `'Published Duration'` é o `'BroadcastEvent'` 310

indicado como uma porção sombreada na FIG. 3. Por conseguinte, o XPath para o fragmento 'BroadcastEvent', '/TVAMain/ProgramDescription//ProgramLocationTable/ /BroadcastEvent' é armazenado no segmento 'fragment\_xpath\_ptr' e os respectivos XPaths para as chaves de Service ID, 'Published Time' e de 'Published Duration' para o fragmento 'broadcastEvent', '@ServiceId' (ver 311a da FIG. 3), 'EventDescription/Published Time' (ver 311b da FIG. 3) e 'EventDescription/ PublishedDuration' (ver 311c da FIG. 3) são armazenados no segmento 'key\_xpath\_ptr'.

A FIG. 7 mostra também a secção de índice de chave (key\_index) 20 e a secção de índice de subchave (sub\_key\_index) 30 para o 'Service ID' (o XPath da chave: @ServiceId) da secção de lista de índice de chave (key\_index\_list) 10.

Numa estrutura de índices deste tipo, quando uma condição de pesquisa para pesquisar os metadados é introduzida, informação de localização sobre um campo da condição de pesquisa introduzida nos metadados é determinada e a informação de localização determinada é comparada com a informação de chave na lista de índice de chave de modo a pesquisar a chave com a informação de localização determinada na lista de índice de chave, uma carga adicional é originada uma vez que é necessária a comparação dos dois XPaths. O mesmo problema ocorre quando as chaves que indicam trajectos relativos dos fragmentos entre a informação de chave são comparadas em termos de informação de localização. Em particular, este problema torna-se mais grave quando fragmentos, que são mais complexos do que as chaves, são comparados em termos de informação de localização. Uma vez que o XPath do fragmento que representa informação de localização entre a informação de chave descreve um trajecto para um nó relevante a partir do nó de raiz sobre o documento XML, custos de transmissão são ineficientes e custos de interpretação do XPath no terminal são elevados. Por exemplo, o XPath do fragmento de evento de difusão que indica informação de localização de um programa de entre os fragmentos TV-Anytime pode ser expresso como '/TVAMain/ProgramDescription//ProgramLocationTable/ /BroadcastEvent'. Entretanto, a fim de representar um nó sobre o documento XML, o XPath pode ser expresso de uma forma alternativa. No caso de um evento difundido, para além da representação normal

supra referida, o XPath pode ser expresso de forma alternativa, tal como '/TVAMain/BroadcastEvent' ou '//BroadcastEvent', etc. Aqui, '/' significa um nó subordinado na estrutura de um documento XML. Por conseguinte, uma operação para inspeccionar se fragmentos são os mesmos através da utilização do XPath não é aquela simples que meramente faz corresponder sequências de caracteres simples entre si. Em particular, custo adicional é originado na análise/comparação do trajecto relevante, se o trajecto XPath for expresso num formato abreviado.

#### APRESENTAÇÃO DO INVENTO

Um aspecto do presente invento é proporcionar um método de fornecimento de um índice dos metadados que permita pesquisar a informação sobre conteúdos de uma forma rápida, um método de pesquisar os metadados através da utilização do índice de metadados e um aparelho de pesquisa com a utilização do mesmo. Aspectos e/ou vantagens adicionais do presente invento serão expostos em parte na descrição que se segue e, em parte, será óbvio da descrição ou pode ser aprendido pela prática do invento.

Para alcançar o acima e/ou outros aspectos do presente invento é proporcionado um método de fornecimento de uma estrutura de índices para metadados dividida em fragmentos, compreendendo o método o fornecimento de uma lista de chaves que correspondem a campos dos metadados e informação de localização para definir uma chave, em que, pelo menos, parte da informação de localização é expressa como um código predeterminado.

O método também pode compreender o fornecimento de valores da chave e de informação de identificação sobre os metadados que correspondem aos valores da chave.

O método pode também compreender o fornecimento de uma sub-secção que inclui gamas de valores da chave e de informação de identificação sobre aqueles dos fragmentos dos metadados que correspondem aos valores da chave e fornecimento de uma secção que inclui valores de chave representativos que representam as respectivas gamas de valores da chave.

A informação de localização pode compreender informação de localização de um fragmento que inclui a chave e informação de identificação da chave no fragmento.

O fornecimento da lista pode compreender o fornecimento da lista com alguma da informação de localização do fragmento e informação de localização da chave cifrada como o código predeterminado.

O código predeterminado pode compreender XPath como informação adicional onde o fragmento/chave respectivo corresponde a um tipo definido de utilizador.

Para alcançar o acima e/ou outros aspectos do presente invento é fornecido outro método de fornecimento de uma estrutura de índices para metadados dividida em fragmentos, compreendendo o método o fornecimento de uma secção de lista de índice de chave que compreende uma lista de chaves que corresponde a campos dos metadados e informação de localização que define as chaves, em que, pelo menos, parte da informação de localização é expressa como um código predeterminado, que fornece uma secção de índice de chave e fornece uma secção de índice de subchave, em que para uma chave da lista de índice de chave, a secção de índice de subchave compreende gamas de valores da chave e informação de identificação sobre aqueles dos fragmentos dos metadados que correspondem aos valores da chave e a secção de índice de chave compreende valores de chave representativos que representam as respectivas gamas de valores da chave.

Para alcançar o acima e/ou outros aspectos do presente invento é proporcionado ainda outro método de fornecimento de uma estrutura de índices para metadados dividida em fragmentos, compreendendo o método o fornecimento de uma lista de chaves que correspondem a campos dos metadados e informação de localização para definir as chaves, em que, pelo menos, parte da informação de localização é expressa como um código predeterminado e que fornece valores das chaves e informação de identificação sobre os metadados que correspondem aos valores das chaves.

A informação de identificação pode compreender informação de identificação sobre os fragmentos dos metadados que correspondem aos valores das chaves.

Para alcançar o acima e/ou outros aspectos do presente invento, é proporcionado para cada um dos métodos descritos acima, um meio que pode ser lido por computador que compreende instruções que podem ser executadas por computador para realizar a(s) operação(ões) citada(s) no método.

#### BREVE DESCRIÇÃO DOS DESENHOS

O acima e outros aspectos e funcionalidades do presente invento ficarão evidentes da descrição que se segue de concretizações preferidas dadas em conjugação com os desenhos em anexo, em que:

a FIG. 1 é um diagrama esquemático que ilustra um conceito de um PDR geral;

a FIG. 2 mostra um ecrã guia de grelha numa aplicação EPG geral;

a FIG. 3 é um diagrama de blocos que ilustra uma estrutura de metadados geral definida pelo fórum TV-Anytime;

a FIG. 4 é um diagrama esquemático que ilustra um conceito de um fragmento geral definido pelo fórum TV-Anytime;

a FIG. 5 é um diagrama esquemático que ilustra um conceito de um contentor geral definido pelo fórum TV-Anytime;

a FIG. 6 é um diagrama de blocos que ilustra uma estrutura de índices de metadados que utiliza o esquema de chave convencional;

a FIG. 7 é um diagrama de blocos que ilustra uma estrutura de índices de metadados e um processo de pesquisa que utiliza o esquema de chave convencional;

a FIG. 8 é um diagrama de blocos que ilustra uma estrutura de índices de metadados de acordo com uma concretização do presente invento;

a FIG. 9 é um diagrama esquemático que ilustra uma estrutura de índices de metadados e um processo de pesquisa de acordo com uma concretização do presente invento;

a FIG. 10 é um diagrama que ilustra um método de fornecimento de índices de metadados de acordo com uma concretização do presente invento;

a FIG. 11 é um diagrama que ilustra um método de pesquisar os metadados de acordo com uma concretização do presente invento; e

a FIG. 12 é um diagrama esquemático que ilustra um aparelho para pesquisar os metadados de acordo com uma concretização do presente invento.

#### MELHOR MODO PARA CONCRETIZAR O INVENTO

Daqui em diante, uma estrutura de índices de metadados fornecida para pesquisar informação sobre conteúdos e um método para fornecimento de índices dos metadados e um método e um aparelho para pesquisar os metadados através da utilização da estrutura de índices dos metadados serão descritos em detalhe com referência aos desenhos em anexo.

As concretizações serão descritas na base de metadados TVA neste fascículo por questão de descrição; no entanto, o mesmo não será interpretado ou compreendido como limitação da cobertura de protecção do presente invento.

A FIG. 8 mostra uma estrutura de índices de metadados para pesquisar os metadados de acordo com uma concretização do presente invento, em que a estrutura de índices inclui informação para definição de uma chave de modo a indexar fragmentos de metadados TVA armazenados no contentor de dados como descrito acima. Daqui em diante, uma secção de lista de índice de chave (key\_index\_list) 110, uma secção de índice de chave (key\_index) 120 e uma secção de índice de subchave

(sub\_key\_index) 130 serão descritas e a estrutura de índices incluindo informação de chave cifrada definida pela sintaxe será então descrita.

A sintaxe que define a estrutura de índices dos metadados de acordo com uma concretização do presente invento, em particular, incluindo a informação de chave cifrada, é diferente em conceito da sintaxe definida numa referência da técnica de índice de chave convencional por a mesma compreender estruturas introduzidas recentemente para um conceito de cifra da informação de chave, tais como `fragment_descriptor()` e `key_descriptor()` e reorganiza estruturas da secção de lista de índice de chave (`key_index_list`) 110, a secção de índice de chave (`key_index`) 120 e a secção de índice de subchave (`sub_key_index`) 130.

A secção de lista de índice de chave (`key_index_list`) 110 compreende informação de chave que define a respectiva chave e informação de identificação sobre a secção de índice de chave (`key_index`) 120 a descrever posteriormente.

A informação de chave serve para definir as chaves, isto é, informação de localização nos metadados, cujos campos predeterminados dos metadados que constituem as chaves têm. A informação de chave compreende informação de localização de um fragmento de metadados a cujos campos que constituem as chaves pertencem nos metadados (daqui em diante referida como "informação de localização de um fragmento", que é expresso como XPath do fragmento em TVA (`fragment_xpath_ptr`) e informação de localização de campos que constituem as chaves que estão no respectivo fragmento de metadados (daqui em diante referida como, por exemplo, "informação de localização de uma chave", que é, um XPath para um trajecto relativo do nó no fragmento relevante, que é expressa como XPath da chave em TVA (isto é, `key_xpath_ptr`).

#### 1. Secção de lista de índice de chave (`key_index_list`)

A secção de lista de índice de chave (`key_index_list`) proporciona uma lista de todas as chaves transmitidas.

Numa concretização do presente invento, um 'fragment\_xpath\_ptr' que indica informação de localização do fragmento na secção de lista de índice de chave (key\_index\_list) convencional (expressa como XPath do fragmento no TVA) é substituído com um fragment\_descriptor().

Tabela 1

Sintaxe	N.º de bits (alterável)
key_index_list() {	
para (j=0; j<key_index_count; j++) {	
fragment_descriptor()	16
key_descriptor()	16
index_container	16
key_index_identifier	8
}	
}	

**key\_index\_count**: especifica o número de todas as chaves transmitidas, isto é, o número de índices para todo o documento XML.

**fragment\_descriptor()**: corresponde à localização XPath de um ou vários fragmentos) alvo a serem indexados. De acordo com uma concretização do presente invento, a informação de localização do fragmento é expressa como um código predeterminado, como mostrado na Tabela 3 abaixo para o tipo de fragmento padrão. O tipo do fragmento não é limitado ao tipo de fragmento padrão da Tabela 3 e o fragmento pode ser conformado de forma tão aleatória quanto possível tanto quanto a sua forma possa indicar o XPath do fragmento para definir as chaves.

**key\_descriptor()**: corresponde a XPaths das chaves na localização XPath do fragmento alvo a ser indexado. Onde a informação de localização da chave é expressa como um código predeterminado, de forma semelhante ao tipo de fragmento descrito acima, o tipo de chave padrão pode ser descrito. Como descrito acima com referência ao fragment\_descriptor(), o tipo de chave não está limitado ao tipo de chave padrão.

**index\_container:** identifica o contentor em que existe uma secção de índice de chave (key\_index) especificada.

**key\_index\_identifier:** identifica a secção de índice de chave (key\_index) no contentor especificado pelo index\_container. A secção de índice de chave (key\_index) pode ser identificada de uma forma única na combinação do index\_container e do key\_index\_identifier.

## 2. Descritor de fragmento (fragment\_descriptor)

O 'fragment\_descriptor()' proporciona uma estrutura de cifrar bits específicos (que podem ser cifrados com bits arbitrários tal como 8 bits, 16 bits, etc.) relativamente ao tipo de fragmento padrão utilizado de forma frequente e ao mesmo tempo, uma estrutura capaz de descrever XPath como informação adicional relativa ao tipo de fragmento de metadados definido pelo utilizador. Isto é, onde o fragment\_descriptor é '0xFF', o mesmo indica um fragmento definido pelo utilizador e, deste modo, XPath para o fragmento definido pelo utilizador relevante é imediatamente descrito.

Tabela 2

Sintaxe	N.º de Bits (alterável)
Fragment_descriptor() {	
fragment_type	8
se(fragment_type == 0xFF) {	
fragment_xpath_ptr	16
}	
}	

**fragment\_type:** representa o tipo de fragmentos a indexar. Valores cifrados são atribuídos aos tipos de fragmento padrão utilizados com frequência. Se o fragment\_type tiver um valor cifrado de 0xFF, o fragment\_xpath\_ptr é adicionado como informação adicional.

A Tabela 3 ilustra valores cifrados para informação de localização dos tipos de fragmentos utilizados com frequência quando uma pesquisa é conduzida no TV-Anytime. No entanto, os tipos de fragmentos padrão e os valores cifrados nesta concretização não estão limitados aos ilustrados na Tabela 3 mas podem ser estendidos de acordo com aplicações.

Tabela 3

Valor	Descrição
0x00	Não designado
0x01	Fragmento de informação de programa
0x02	Fragmento de informação de grupo
0x03	Fragmento de informação de créditos
0x04	Fragmento de análise de programa
0x05	Fragmento de informação de segmento
0x06	Fragmento de informação de serviço
0x07	Fragmento de evento de difusão
0xFF	Fragmento designado pelo utilizador
0x08-0x0E	Reservado
0x10-0xFF	

### 3. Descritor de chave (key\_descriptor)

6 O 'key\_descriptor()' proporciona uma estrutura de cifra de informação de localização das chaves com uma frequência elevada de utilização para especificar bits quando uma pesquisa é realizada e, ao mesmo tempo, uma estrutura de descrição do tipo de chave definido pelo utilizador no XPath. Por exemplo, se o key\_descriptor for '0xFF', o mesmo indica uma chave definida pelo utilizador. Assim, o XPath é descrito como informação adicional para a chave definida pelo utilizador.

Tabela 4

Sintaxe	N.º de Bits (alterável)
key_descriptor() {	
key_type	8
se (key_type == 0xFF) {	
key_xpath_ptr	16
}	
}	

**key\_type:** representa o tipo de chaves a indexar. Valores cifrados são atribuídos a informação de localização dos tipos de chave padrão utilizados com frequência quando uma pesquisa é conduzida. Se o key\_type tem um valor cifrado de '0xFF', o key\_xpath\_ptr é adicionado como informação adicional.

**key\_xpath\_ptr:** refere-se ao trajecto relativo envolvido no XPath de fragmento do nó utilizado como a chave.

Apesar dos valores cifrados para as chaves padrão não terem sido especificados, será entendido que os valores cifrados para os tipos de chave padrão podem ter uma estrutura semelhante para a cifra dos tipos de fragmentos da Tabela 3.

Uma vez que as definições da secção de índice de chave (key\_index) e da secção de índice de subchave (sub\_key\_index) são as mesmas como as definidas na referência de técnica de índice de chave, a descrição detalhada da mesma será omitida.

#### 4. Secção de índice de chave (key\_index)

Tabela 5

Sintaxe	N.º de Bits(alterável)
key_index() {	
key_index_identifier	8
para (j=0; j<contador_sub_indice; j++) {	
high_key_value	16
sub_index_container	16
sub_index_identifier	8
}	
}	

## 5. Secção índice de subchaves (sub\_key\_index)

Tabela 6

Sintaxe	N.º de Bits (alterável)
sub_key_index() {	
sub_index_identifier	8
for (j=0; j<reference_counte; j++) {	
key_value	16
target_container	16
target_handle	16
}	
}	

Daqui em diante, a estrutura de metadados definida pela sintaxe descrita acima será discutida com referência à FIG. 8, na qual os metadados são expressos como segmentos da informação de índice.

A secção de lista de índice de chave (key\_index\_list) 110 definida na estrutura de índices fornece uma lista de todas as chaves transmitidas. A lista inclui informação de chave que define cada chave (isto é, informação de localização do fragmento (fragment\_descriptor) e/ou informação de localização das chaves (key\_descriptor); a informação de localização do fragmento ou a informação de localização da chave pode ser cifrada de forma selectiva ou as mesmas podem ser cifradas em simultâneo em função das concretizações do presente invento) e informação de identificação sobre a secção de índice de chave (key\_index) 120 a descrever posteriormente. O XPath do fragmento de metadados é um trajecto para o nó de raiz do documento XML de metadados TVA, isto é, um trajecto absoluto, da mesma forma que na estrutura de índices convencional e o XPath do nó utilizado como a chave, isto é, o XPath da chave, representa um trajecto relativo da chave para o fragmento de metadados. O XPath do fragmento de metadados e o XPath da chave em combinação representa informação de localização da chave para todo o documento XML.

Na presente concretização, o valor cifrado correspondente ao XPath para o fragmento de metadados (isto é, informação de localização do grupo de fragmentos) e o valor cifrado correspondente ao XPath da chave (isto é, informação de localização da chave) são, respectivamente, armazenados no segmento 'fragment\_descriptor' 111 e no segmento 'key\_descriptor' 112.

Como descrito acima, onde a informação de localização do fragmento entre a informação de chave é do tipo de fragmento padrão que é utilizada com frequência, é fornecido um valor cifrado (fragment\_descriptor) que expressa o XPath para o fragmento de metadados (fragment\_xpath\_ptr) com um código predeterminado. Como os tipos de fragmento padrão utilizados com frequência, existem por exemplo, informação de programa (ProgramInformation), informação de grupo de programa (GroupInformation), informação de crédito (CreditInformation), revisão de programa (ProgramReview), informação de segmento (SegmentInformation), evento de difusão (BroadcastEvent), informação de serviço (ServiceInformation) e semelhantes. Se o XPath do fragmento de metadados para estes tipos de fragmentos poder ser expresso simplesmente como um valor cifrado, a carga adicional na pesquisa dos metadados pode ser reduzida.

Por conseguinte, na estrutura de índices de acordo com a presente concretização, o XPath do fragmento de metadados padrão é cifrado para um valor cifrado predeterminado e depois armazenado. Para além disso, nem todos os valores cifrados são atribuídos aos fragmentos e alguns dos valores cifrados (por exemplo, '0XFF') são atribuídos aos fragmentos de metadados como definido pelo utilizador, para deste modo permitir ao utilizador definir, de forma adicional, informação de localização no fragmento de metadados por meio do XPath. Quanto a isto, uma área adicional ('fragment\_xpath\_ptr'), por exemplo, através da qual o XPath para o fragmento de metadados pode ser designado é fornecida.

Na concretização em que fragmentos são cifrados de acordo com a Tabela 3, a informação de localização sobre o fragmento de metadados de entre a informação de chave tem estes valores cifrados como '0x01', '0x02' e '0x03'. A

informação de localização sobre o fragmento de metadados cifrada para '0x01' indica o XPath do 'fragmento de informação de programa (ProgramInformation)'. Além disso, onde a informação de localização sobre o fragmento de metadados é '0xFF', significa que o fragmento de metadados definido pelo utilizador, e, deste modo, uma área adicional para permitir que o XPath do fragmento de metadados seja designado é fornecido.

Apesar da concretização acima ter sido descrita em relação apenas com o fragmento de metadados, a mesma pode ser aplicada em relação à(s) chave(s) para o fragmento de metadados. Isto é, valores cifrados podem ser designados e utilizados para as chaves utilizadas com frequência em vez do XPath convencional para as chaves. Além disso, se o valor cifrado compreende um valor predeterminado, o utilizador pode, de forma adicional, designar o XPath para a chave. A cifra do XPath do fragmento de metadados supra referido e a cifra do XPath da chave podem ser utilizadas em simultâneo ou de forma independente.

Para além disso, a secção de lista de índice de chave (key\_index\_list) 110 compreende a informação de identificação sobre a secção de índice de chave (key\_index) 120 de cada chave a ser descrita posteriormente (isto é, a informação de identificador de contentor (container\_id) do contentor que armazena a secção de índice de chave (key\_index) 120 e a informação de identificador de índice de chave). A informação de identificador de contentor e a informação de identificador de índice de chave são, respectivamente, armazenadas num segmento 'index\_container' e num segmento 'key\_index\_identifier' na secção de lista de índice de chave (key\_index\_list) 110.

Uma vez que a secção de índice de chave (key\_index) 120 e a secção de índice de subchave (sub\_key\_index) 130 são idênticas como descrito na referência de técnica de índice de chave, a descrição das mesmas será omitida.

A estrutura de índices que inclui a informação de chave cifrada será descrita em detalhe com referência à FIG. 9, a

qual ilustra a informação de índice de acordo com uma concretização do presente invento.

A FIG. 9 mostra a secção 110 de lista de índice de chave na qual o XPath de fragmento 'BroadcastEvent' para o 'Service ID' está cifrado '0x07'. Aqui, a secção de índice de chave (key\_index) 120 e a secção de índice de subchave (key\_index) 130 são idênticas como descrito em relação à FIG. 7.

A estrutura de índices descrita acima é muito eficiente quando as chaves relativas aos tipos de fragmentos utilizados de forma frequente, por exemplo, ProgramInformation, GroupInformation e BroadcastEvent, etc. são utilizadas, o que, deste modo, reduz toda a carga adicional no aparelho para pesquisa de metadados.

A FIG. 10 ilustra um método de fornecimento de um índice de metadados com uma estrutura de acordo com uma concretização do presente invento como descrito acima.

Índices dos metadados de acordo com uma concretização do presente invento podem ser gerados pelo fornecedor 200 que fornece, por exemplo, sinais áudio/visuais.

A informação sobre conteúdos, isto é, metadados, é processada primeiro numa base de fragmento como descrito acima (S100). Pelo menos uma porção (informação de localização do fragmento ou informação de localização da chave) de informação sobre os campos que serão incluídos no índice de metadados, isto é, informação sobre a chave (por exemplo, informação de localização do fragmento e informação de localização da chave) é cifrada (S200). Por outras palavras, onde a informação de localização do fragmento de metadados cujos campos que constituem chaves pertencem ou informação de localização da chave é do tipo de fragmento padrão ou do tipo de chave padrão, ambas podem ser cifradas, a informação de localização do fragmento de metadados ou a informação de localização da chave, isto é, o XPath do fragmento de metadados ou o XPath da chave é cifrada para o valor de código predeterminado (por exemplo, o 'fragmento de evento de difusão (BroadcastEvent) é cifrado para '0X07' na FIG. 9). Onde a informação de localização do fragmento de

metadados ou a informação de localização da chave não são identificadas pelo valor cifrado, a informação de chave expressa com XPath pode ser utilizada como na técnica convencional.

Uma chave é fornecida através da utilização de informação que constitui o fragmento, por exemplo, informação sobre um 'Service ID' (S300). Então, uma secção de índice de subchave (sub\_key\_index) 130 é fornecida para a chave como fornecida acima (S400). A secção de índice de subchave (sub\_key\_index) 130 inclui segmentos 114 que têm gamas de valores da chave e informação de identificação de fragmentos de metadados que correspondem aos valores da chave (isto é, a informação de identificador de contentor (container\_id) e informação de identificador de dados de fragmento (handle\_value) respectivamente armazenadas no segmento 'target\_container' e no segmento 'target\_container' da FIG. 8).

Uma secção de índice de chave (key\_index) 120 com valores de chave representativos que representam as respectivas gamas de valores da chave é fornecida (S500). Por exemplo, um valor de chave representativa (por exemplo, 509) que indica uma gama predeterminada (por exemplo, 500~509) do 'Service ID' está incluído. A secção de índice de chave (key\_index) 120 inclui informação de identificação para a secção de índice de subchave (sub\_key\_index) 130, em que a informação de identificação compreende a informação de identificador de contentor (container\_id) do contentor em que a secção de índice de subchave (sub\_key\_index) 130 é armazenada e a informação de identificador de índice de subchave como mostrado na FIG. 8.

Uma secção de lista de índice de chave (key\_index\_list) 110 configura informação de chave como fornecida acima, isto é, informação de localização do fragmento e informação de localização da chave, baseada na chave, é fornecida (S600). Nesta altura, se a informação de localização cifrada do fragmento ou a informação de localização cifrada da chave no passo de S200 existir, a informação de localização acima é expressa como um código cifrado quando a secção de lista de índice de chave (key\_index\_list) 110 é fornecida. Por outras

palavras, por exemplo, o fragmento de 'broadcastEvent (BroadcastEvent)' na FIG. 9 é expresso como '0X07'. Onde a informação de localização do fragmento ou a informação de localização da chave não podem ser distinguidas por um valor cifrado, a informação de chave expressa no XPath como na técnica convencional pode ser utilizada.

A secção de lista de índice de chave (key\_index\_list) 110 também compreende informação de identificação sobre a secção de índice de chave (key\_index) 120, para além da informação de chave.

Os passos descritos acima podem prosseguir na ordem inversa noutras concretizações do presente invento e o passo S500 de proporcionar a secção de índice de chave (key\_index) 120 incluindo os valores de chave representativos pode ser omitido em função da(s) concretização(ões).

Aqui abaixo, um método de pesquisa de metadados que satisfaçam uma condição de pesquisa através da utilização do índice de metadados com uma estrutura de acordo com uma concretização do presente invento descrita acima, será descrito com referência à FIG. 11.

Uma condição de pesquisa é introduzida, por exemplo, por um utilizador (S1100) e informação de localização de metadados relativa a um campo da condição de pesquisa introduzida é determinada (S1210). Uma chave correspondente à informação de localização do campo é pesquisada na secção de lista de índice de chave (key\_index\_list) 110 (S1300), em que, pelo menos, uma parte de informação de localização, por exemplo, informação de localização de um fragmento que inclui a chave ou informação de localização da chave no fragmento, está definida com um código predeterminado e os metadados respectivos são extraídos através da utilização da chave pesquisada (S1400).

O passo de extracção dos metadados respectivos, S1400, compreende os passos de pesquisa de um valor de chave representativo que satisfaça a condição de pesquisa, em comparação do valor de chave representativo e da gama de valores da chave da condição de pesquisa, na secção de índice

de chave (key\_index) 120 e a pesquisa da secção de índice de subchave (sub\_key\_index) 130 para um segmento 114 que inclui os valores da chave na gama representada pelo valor (S1410) de chave representativo pesquisado, pesquisa de um valor da chave que satisfaça a condição de pesquisa no segmento 114 da secção de índice de subchave (sub\_key\_index) 130 (S1420) pesquisada e extracção dos metadados respectivos através da utilização da informação de identificação do fragmento de metadados correspondente ao valor pesquisado da chave, de modo que o fragmento de metadados que satisfaz a condição de pesquisa seja extraído. Entende-se que, por exemplo com referência às FIGS. 2 e 9, quando é introduzida uma condição de pesquisa que corresponda a uma chave de 'Service ID' na gama de 507-514, sejam pesquisados valores de chave representativos 509 e 519, sejam pesquisados valores das respectivas gamas de valores da chave e sejam extraídos fragmentos correspondentes à condição de pesquisa através da utilização da informação de identificação sobre os fragmentos correspondente aos valores da chave.

A informação de localização do fragmento refere-se a um trajecto absoluto do fragmento de metadados, cujas chaves se destinam a serem indexadas como descrito acima, isto é, o XPath do fragmento de metadados (fragment\_xpath\_ptr) e a informação de localização da chave refere-se a um trajecto relativo da chave para o fragmento de metadados (trajecto relativo na localização XPath do fragmento), isto é, o XPath (key\_descriptor) dos nós utilizados como chaves.

Nos passos de S1410, S1420 e S1430, os passos de pesquisa da secção de índice de chave (key\_index) 120 respectiva e da secção de índice de subchave (sub\_key\_index) 130 e de extracção do fragmento respectivo prosseguem através da utilização da informação de identificação da secção de índice de chave (key\_index) 120, da secção de índice de subchave (sub\_key\_index) e do fragmento de metadados, respectivamente.

A FIG. 12 representa um aparelho para pesquisar os metadados de acordo com uma concretização do presente invento. O aparelho concretiza um método de pesquisar os

metadados de acordo com o presente invento descrito com referência à FIG. 11.

O aparelho 1000 compreende uma unidade de entrada 1100 que permite que um utilizador introduza uma condição de pesquisa, uma unidade de recepção 1200 que recebe conteúdos, metadados sobre conteúdos ou um índice dos metadados, uma unidade de armazenagem 1300 que armazena os conteúdos recebidos, os metadados sobre os conteúdos ou o índice dos metadados, uma unidade de controlo 1400 que determina informação de localização dos metadados correspondentes ao campo da condição de pesquisa introduzida a partir da unidade de entrada 1100, pesquisa da chave que contém o código predeterminado como informação de localização, onde, pelo menos, uma parte da informação de localização está definida como o código predeterminado e extracção dos metadados respectivos através da utilização da chave pesquisada e uma unidade de saída 1500 que fornece o resultado da pesquisa pela unidade de controlo 1400.

A unidade de controlo 1400 compara a condição de pesquisa introduzida a partir da unidade de entrada 1100 com o valor da chave contida no índice de metadados armazenado na unidade de armazenagem 1300.

De entre os passos de pesquisa dos metadados de acordo com uma concretização do presente invento, o passo de determinar informação de localização do campo da condição de pesquisa introduzida nos metadados (S1210), o passo de pesquisa da chave que contém o código predeterminado como informação de localização, onde, pelo menos, uma parte da informação de localização está definida como o código predeterminado (S1300) e o passo de extrair os metadados respectivos através da utilização da chave pesquisada (S1400) são realizados na unidade de controlo 1400. Descrições destes passos foram descritas com referência à FIG. 11.

O presente invento propõe uma estrutura de índices que proporciona uma indexação simplificada para fragmentos de metadados para pesquisar os fragmentos de metadados de uma forma rápida, no ambiente em que os metadados estão estruturados sobre uma base de fragmento, um método para

pesquisar a informação de índice e um aparelho para pesquisar a informação de índice.

#### APLICABILIDADE INDUSTRIAL

De acordo com o presente invento, uma pesquisa rápida de metadados está disponível e a carga para o aparelho pesquisar os metadados é reduzida, o que, deste modo, encurta o tempo de pesquisa e aumenta a eficiência do aparelho para pesquisar os metadados. No entanto, entende-se que embora concretizações ilustrativas, não limitativas do presente invento ultrapassem as desvantagens descritas acima e outras desvantagens não descritas acima, o presente invento não é necessário para ultrapassar as desvantagens descritas acima e concretização ilustrativas, não limitativas do presente invento podem não ultrapassar qualquer dos problemas descritos acima. É também entendido que um sistema que utilize o presente invento também inclui a armazenagem permanente ou amovível, tal como discos ópticos e magnéticos, RAM, ROM, um meio de onda portadora, etc., no qual o processo e as estruturas de dados do presente invento podem ser armazenados e distribuídos. As operações também podem ser distribuídas através, por exemplo, de descarga através de uma rede tal como a Internet.

Apesar de terem sido mostradas e descritas algumas concretizações preferidas, será apreciado por quem for especializado na técnica que várias alterações e modificações podem ser realizadas como definido nas reivindicações em anexo.

Lisboa,

## REIVINDICAÇÕES

1 - Método de fornecimento de uma estrutura de índices para metadados relativos a conteúdos de difusão, dispostos num esquema semi-estruturado predeterminado e divididos em fragmentos como unidades dos metadados transmitidas de forma independente e acedidas de forma individual, compreendendo o método o fornecimento de uma lista (110) das chaves correspondentes a campos dos metadados e informação de localização (111, 112) para definição de uma chave relativa ao esquema semi-estruturado, caracterizado por a informação de localização (111, 112) compreender informação de localização (111) de um fragmento que inclui a chave e informação de localização (112) da chave dentro do fragmento e por, pelo menos, parte da informação de localização (111, 112) ser expressa como um código predeterminado.

2 - Método de acordo com a reivindicação 1, que compreende ainda o fornecimento de valores da chave e informação de identificação sobre os metadados correspondentes aos valores da chave.

3 - Método de acordo com a reivindicação 1, que compreende ainda:

o fornecimento de uma sub-secção (130), que inclui gamas de valores da chave e informação de identificação sobre os fragmentos dos fragmentos de metadados, que correspondem aos valores da chave; e

o fornecimento de uma secção (120) que inclui os valores de chave representativos que representam as respectivas gamas dos valores da chave.

4 - Método de acordo com a reivindicação 1, em que o fornecimento da lista (110) compreende o fornecimento da lista que tem uma da informação de localização (111) do fragmento e informação de localização (112) da chave cifrada como o código predeterminado.

5 - Método de acordo com a reivindicação 4, em que o código predeterminado compreende XPath como informação

adicional onde o respectivo fragmento/chave corresponde a um tipo definido de utilizador.

6 - Método de acordo com a reivindicação 4, em que a outra da informação de localização (111) do fragmento e a informação de localização (112) da chave é expressa como um outro código predeterminado ou XPath.

7 - Método de acordo com qualquer reivindicação anterior, que compreende:

o fornecimento de uma secção de lista de índice de chave (110) que compreende a lista das chaves que correspondem a campos dos metadados e a informação de localização (111, 112), que define as chaves, em que, pelo menos parte da informação de localização (111, 112) é expressa como um código predeterminado;

o fornecimento de uma secção de índice de chave (120); e

o fornecimento de uma secção de índice de subchave (130);

em que para uma chave da lista de índice de chave (110):

a secção de índice de subchave (130) compreende gamas de valores da chave e informação de identificação sobre os fragmentos dos metadados que correspondem aos valores da chave, e

a secção de índice de chave (120) compreende valores de chave representativos que representam as respectivas gamas dos valores da chave.

8 - Método de acordo com a reivindicação 7, que compreende ainda o fornecimento de uma correspondente secção de índice de chave (120) e uma correspondente secção de índice de subchave (130) para uma outra chave da lista de índice de chave (110).

9 - Método de acordo com qualquer reivindicação anterior, que compreende:

o fornecimento da lista (110) das chaves, que correspondem a campos dos metadados e informação de localização (111, 112) para definição das chaves, em que, pelo menos parte da informação de localização é expressa como um código predeterminado; e

o fornecimento de valores das chaves e informação de identificação sobre os metadados que corresponde aos valores das chaves.

10 - Método de acordo com a reivindicação 9, em que a informação de identificação compreende a informação de identificação sobre os fragmentos dos metadados que correspondem aos valores das chaves.

Lisboa,

FIG 1

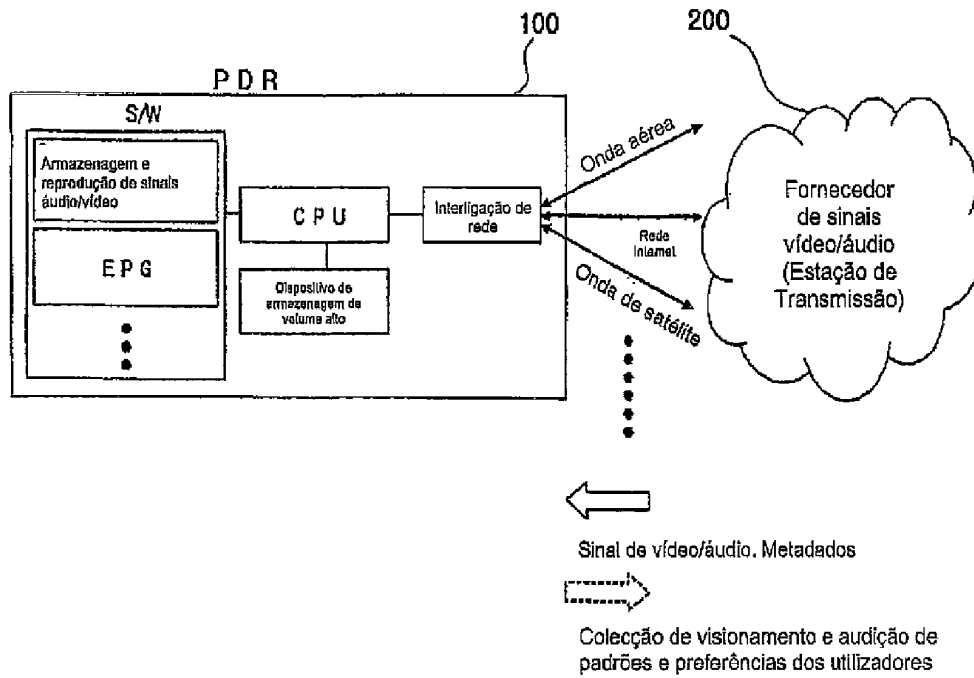


FIG. 2

Today	9:00p	9:30p	10:00p
507 HBOF	Out of Africa - Movie - Spanish <b>HD</b>		
508 HBFW	Into the West - Movie	Hocus Pocus	
512 MAX	Vertical Limit - Movie <b>HD</b>	American Be.	
513 MMAX	The Gauntlet - Movie		
514 MaxW	The Bridges of Madison County - Movie <b>LB</b>		

FIG. 3

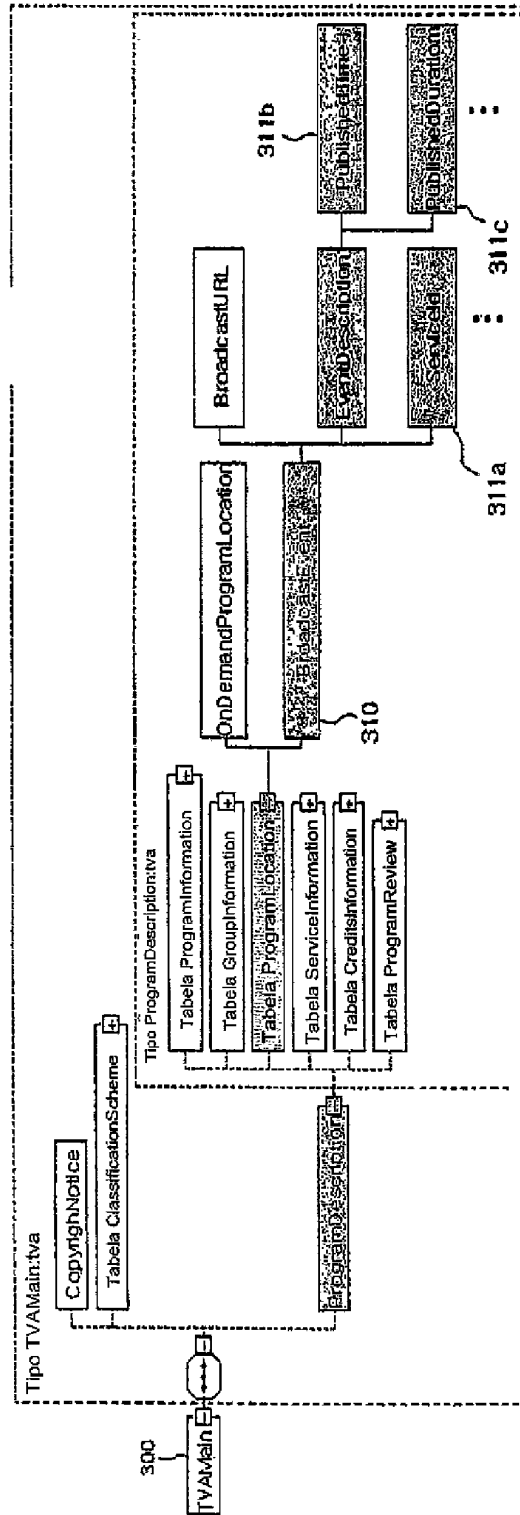


FIG. 4

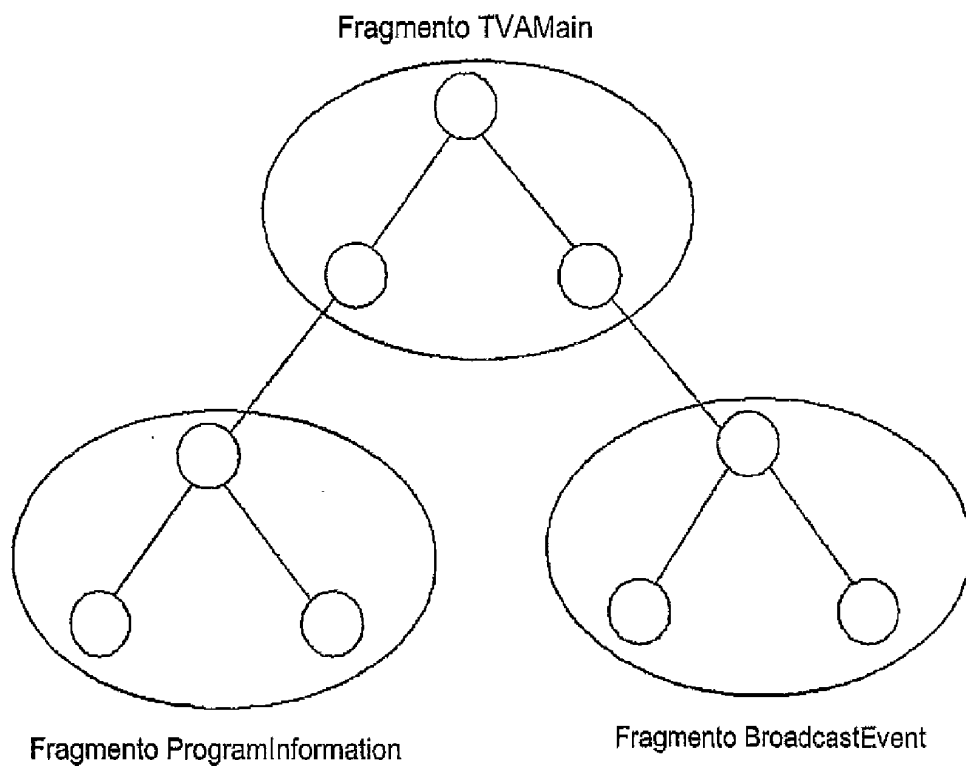
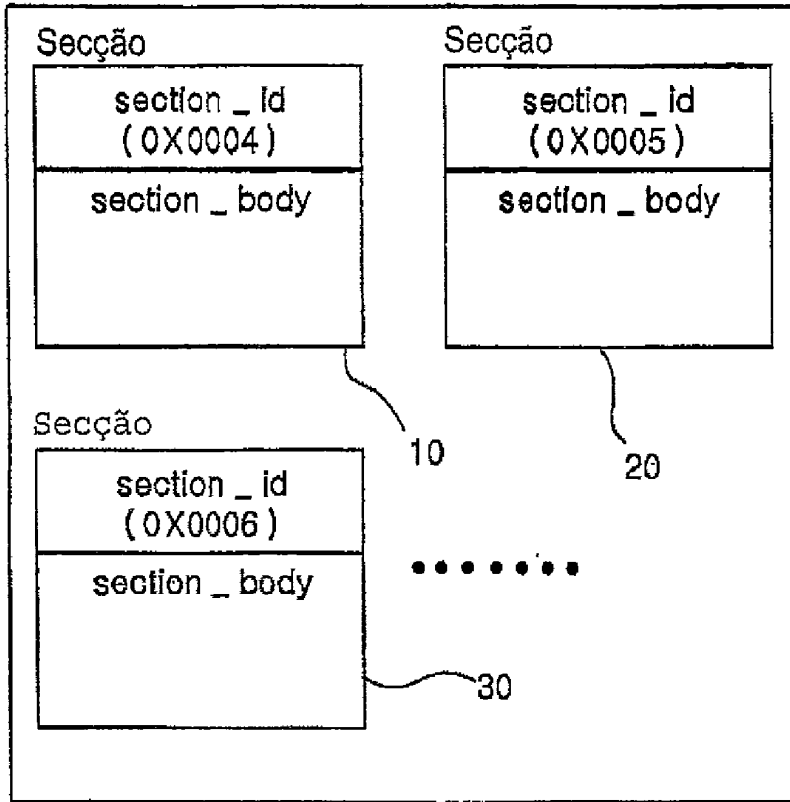


FIG. 5

Contentor de índice



Contentor de dados

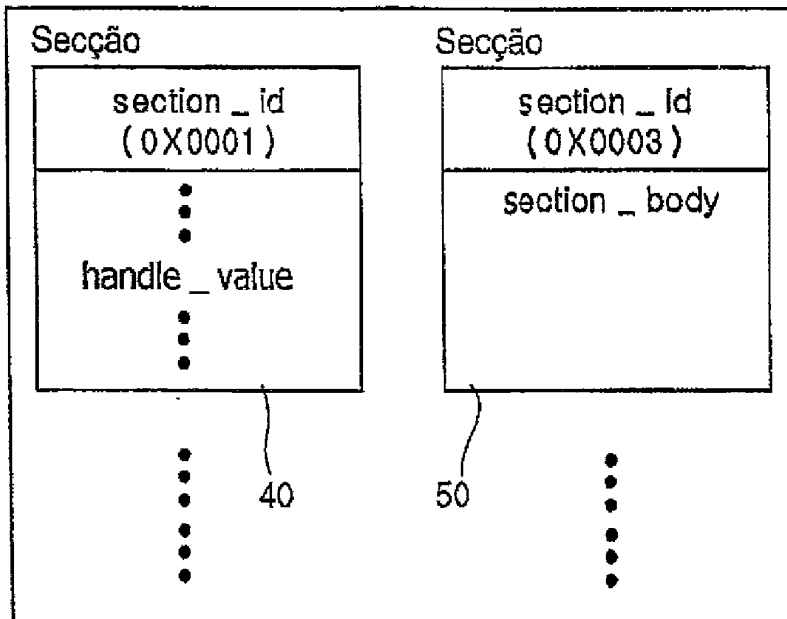
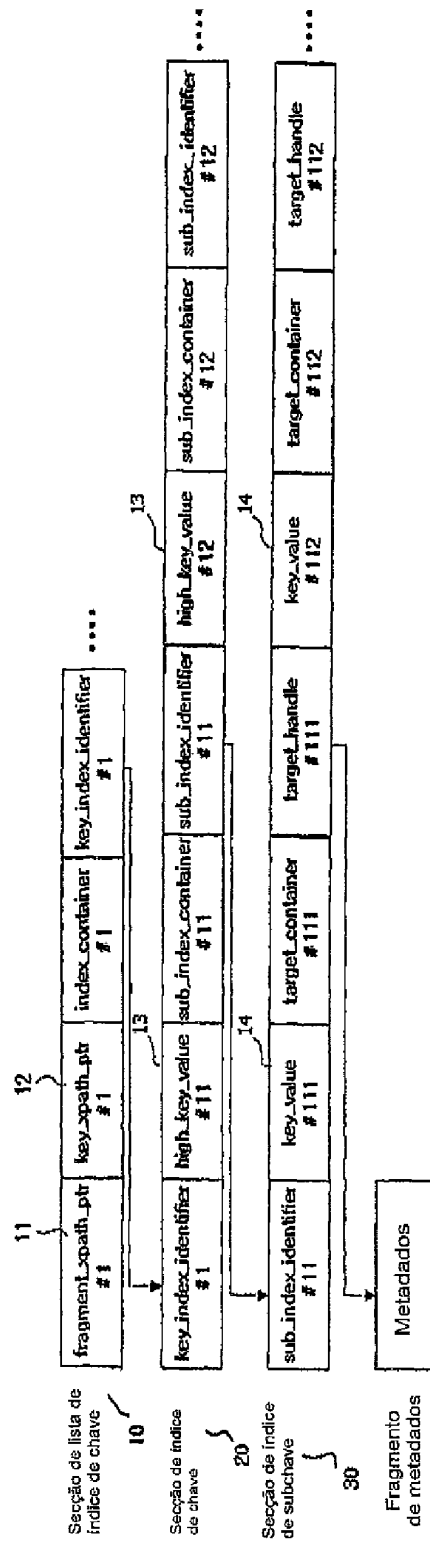


FIG 6



10  
Seção de lista de índices de chave

20  
Seção de índice de chave

30  
Seção de índice de subchave

30  
Fragmento de metadados

FIG. 7

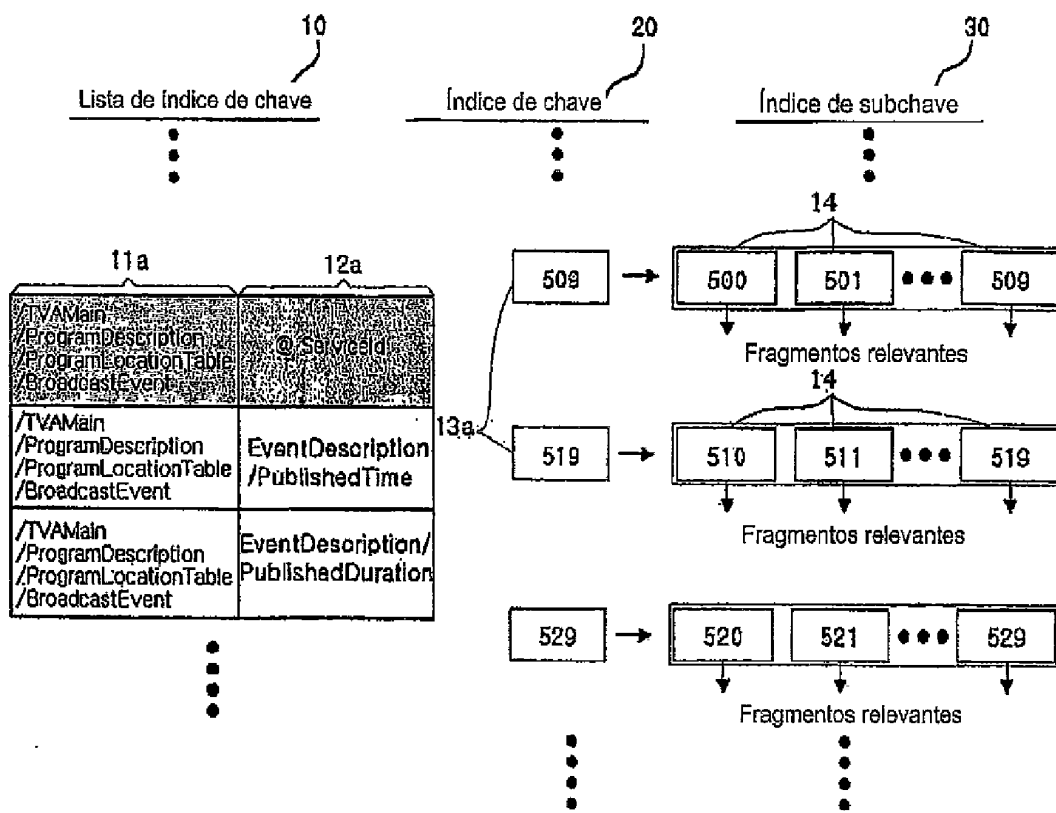


FIG. 8

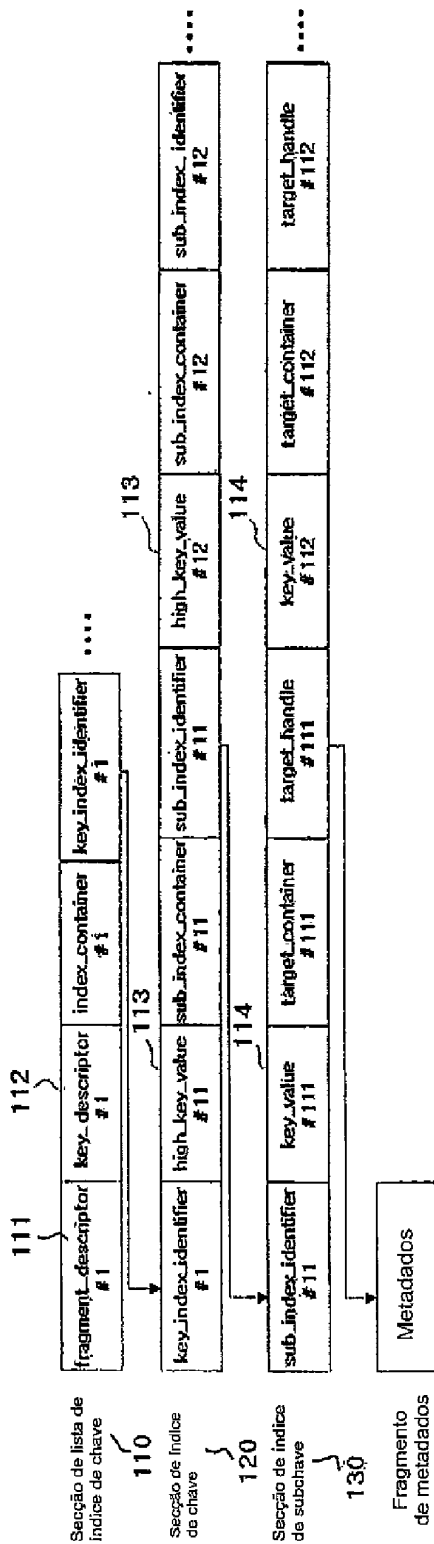


FIG. 9

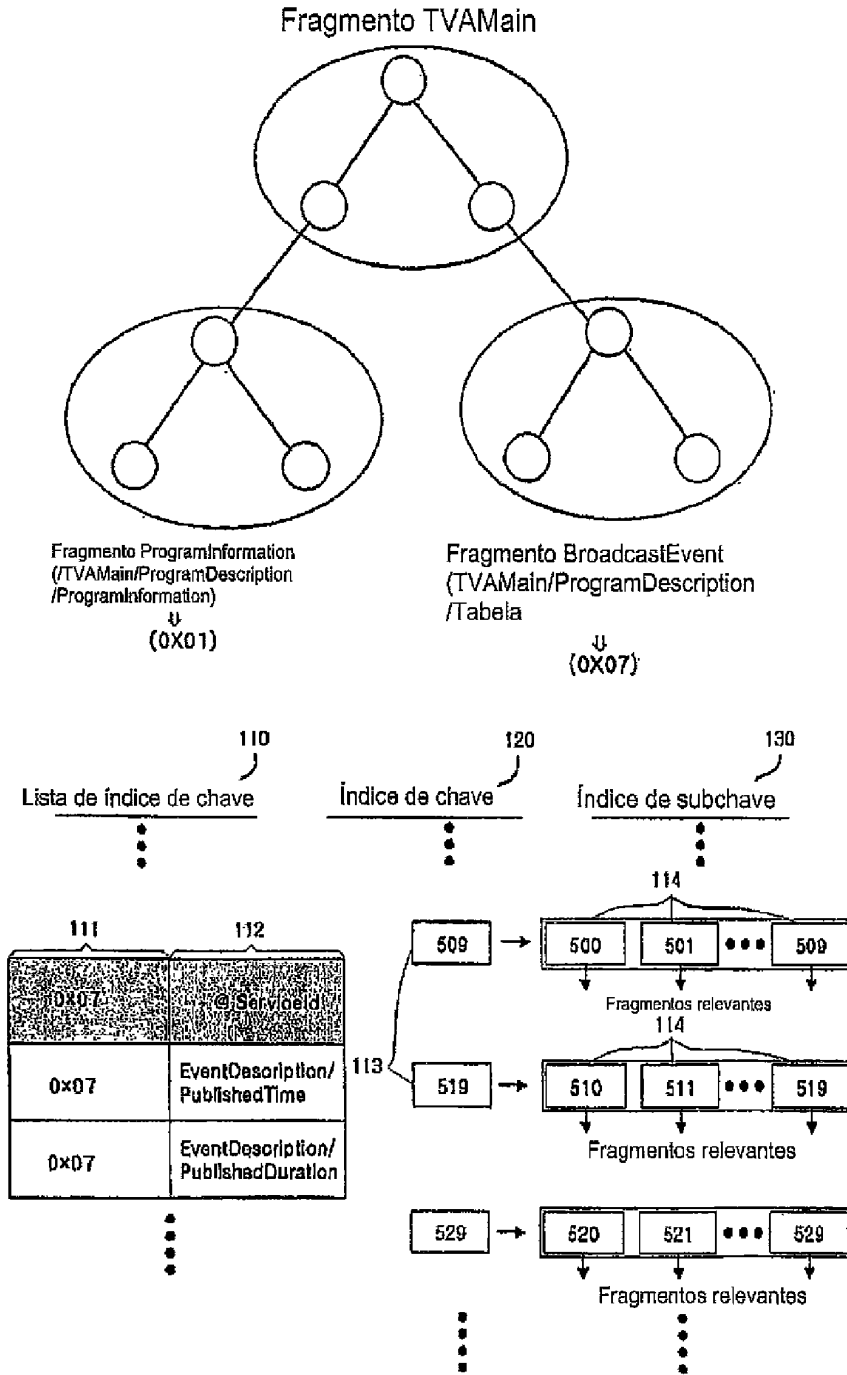


FIG. 10

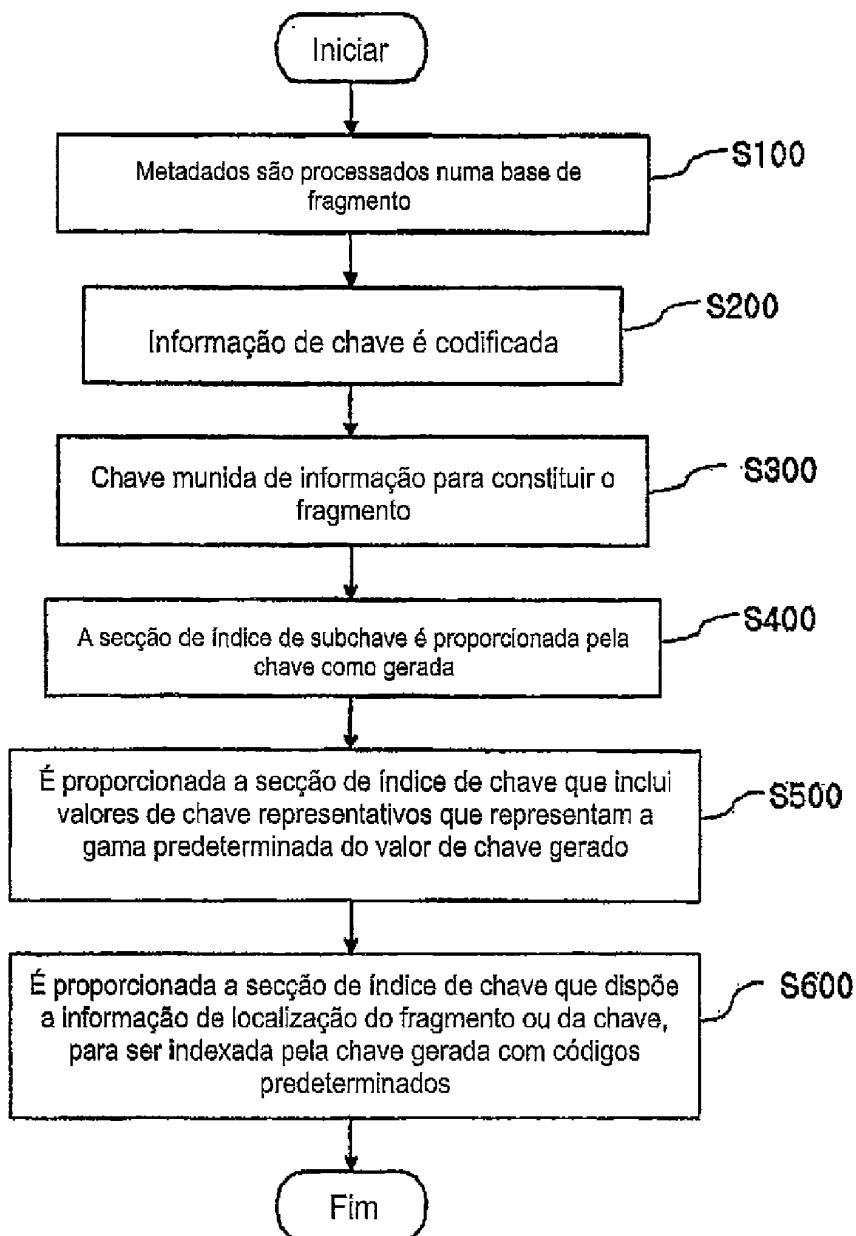


FIG. 11

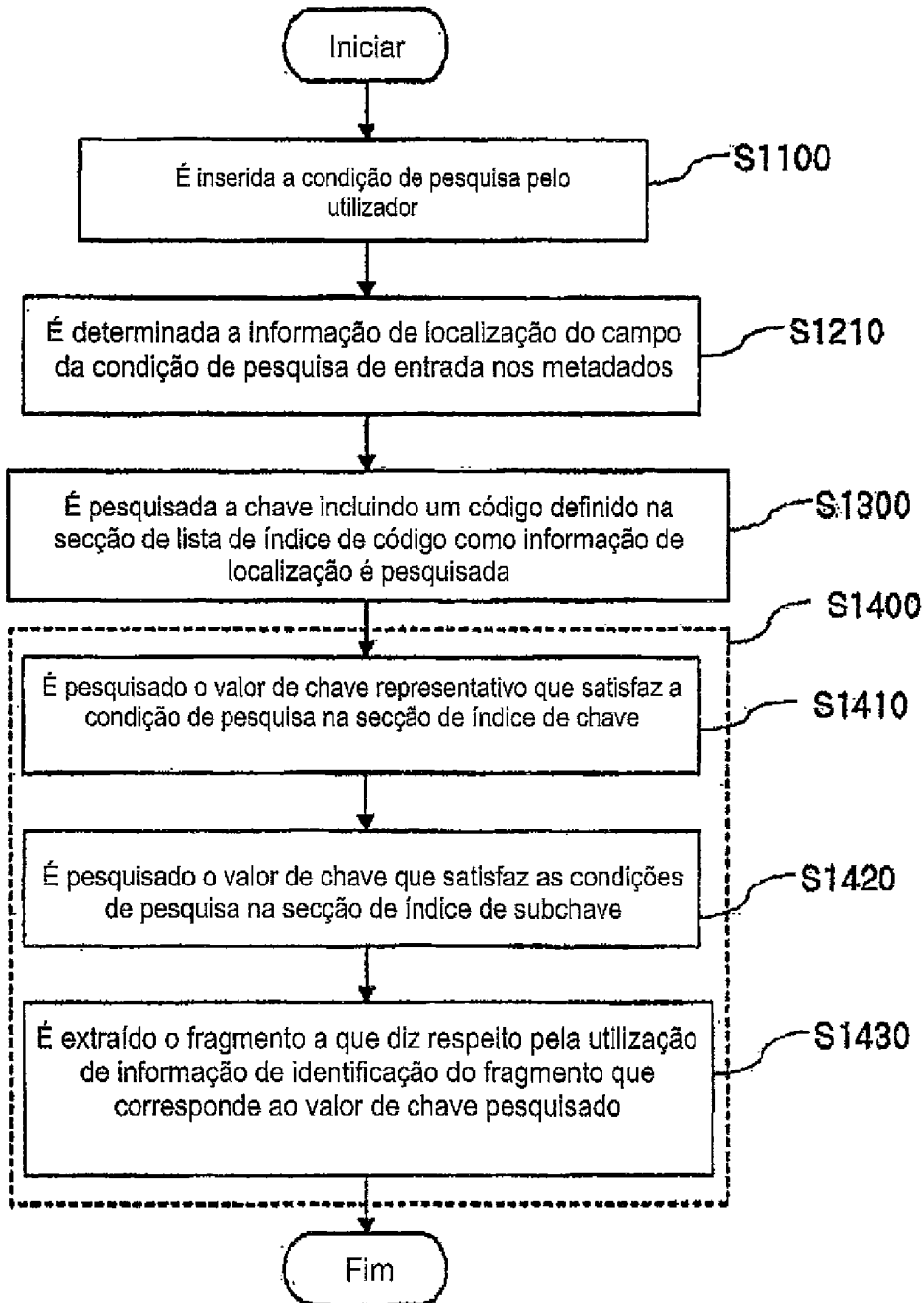


FIG. 12

