

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5001510号  
(P5001510)

(45) 発行日 平成24年8月15日(2012.8.15)

(24) 登録日 平成24年5月25日(2012.5.25)

(51) Int. Cl. F I  
**G 0 6 F 17/30 (2006.01)** G O 6 F 17/30 2 2 O C  
**G 0 6 F 12/00 (2006.01)** G O 6 F 12/00 5 4 7 Z

請求項の数 6 (全 17 頁)

(21) 出願番号	特願2002-555536 (P2002-555536)	(73) 特許権者	390023711
(86) (22) 出願日	平成13年12月21日(2001.12.21)		ローベルト ボツシュ ゲゼルシャフト
(65) 公表番号	特表2004-517422 (P2004-517422A)		ミット ベシユレンクテル ハフツング
(43) 公表日	平成16年6月10日(2004.6.10)		ROBERT BOSCH GMBH
(86) 国際出願番号	PCT/EP2001/015194		ドイツ連邦共和国 シュツツガルト (
(87) 国際公開番号	W02002/054775		番地なし)
(87) 国際公開日	平成14年7月11日(2002.7.11)		Stuttgart, Germany
審査請求日	平成16年12月20日(2004.12.20)	(74) 代理人	100061815
審査番号	不服2009-8164 (P2009-8164/J1)		弁理士 矢野 敏雄
審査請求日	平成21年4月16日(2009.4.16)	(74) 代理人	100135633
(31) 優先権主張番号	01100441.3		弁理士 二宮 浩康
(32) 優先日	平成13年1月8日(2001.1.8)	(74) 代理人	100114890
(33) 優先権主張国	欧州特許庁 (EP)		弁理士 アインゼル・フェリックス=ライ ンハルト

最終頁に続く

(54) 【発明の名称】 マルチメディアデータのバイナリ記述用拡張コードを供給する方法

(57) 【特許請求の範囲】

【請求項 1】

マルチメディアデータのバイナリ記述のための拡張コードをエンコーダからデコーダに供給する方法において、

前記マルチメディアデータは記述要素として記述子(D)と記述スキーム(DS)を有しており、

前記エンコーダにより、拡張ツリーブランチコード(TBC)が前記マルチメディアデータのバイナリフォーマット(BiM)に挿入され、

該拡張ツリーブランチコード(TBC)によって、拡張された前記記述スキーム(DS)が2つのパートに分割され、一方のパートは前記まえて定義されベースとなる記述スキーム(DS)と同一であり、他方のパートは前記拡張ツリーブランチコード(TBC)以降に続くことになる拡張が記述されていることが、前記エンコーダから前記デコーダに通知されることを特徴とする、

マルチメディアデータのバイナリ記述のための拡張コードをエンコーダからデコーダに供給する方法。

【請求項 2】

前記まえて定義されベースとなる記述スキーム(DS)はMPEG-7の記述スキーム(DS)である、請求項1記載の方法。

【請求項 3】

前記エンコーダによって、少なくとも前記一方のパートがバイナリフォーマット(Bi

M)における1つまたは複数のフラグメントにより表され、該フラグメントにより拡張記述スキーム(DS)のうちMPEG-7と合致したパートが、符号変換されることなく前記デコーダにより抽出される、請求項1または2記載の方法。

【請求項4】

前記エンコーダにより、前記拡張ツリーブランチコード(TBC)がポジションコードといっしょに使用されて、1つの記述スキーム(DS)のノードにおいて2つ以上の拡張を可能にする、請求項1から3のいずれか1項記載の方法。

【請求項5】

前記エンコーダにより、MPEG-7固有の拡張ツリーブランチコード(TBC)が挿入され、MPEG-7標準の記述子(D)および記述スキーム(DS)の拡張が定義されて前方互換を可能にする、請求項1から4のいずれか1項記載の方法。

10

【請求項6】

拡張を伴うMPEG-7のデータのためのバイナリフォーマットは、ナビゲーションコマンドとナビゲーションパスとコンテンツマニピュレーションコマンドとサブツリーバイナリ表現であって、

前記ナビゲーションコマンドは、記述スキーム(DS)のどのパートを次に受け取ることになるのかの情報をデコーダに提供し、

前記ナビゲーションパスは、デコーダがMPEG-7データフォーマット中に含まれる情報を受け取るツリーノードのアドレスを提供し、

前記コンテンツマニピュレーションコマンドは、サブツリーに対しどのような操作を行うのかの情報をデコーダに提供する、

請求項1から5のいずれか1項記載の方法。

20

【発明の詳細な説明】

【0001】

従来の技術

MPEG-7標準[1]のコンテキストにおいてマルチメディアデータの記述は記述子(D)と記述スキーム(DS)によって構成されており、この場合、記述要素(DまたはDS)の個数は可変である。MPEG-7におけるすべてのDおよびDSは、いわゆる記述定義言語(DDL)を使用して定義される[3]。また、MPEG-7はテキストフォーマット(XML)でもバイナリフォーマット(BiM)でも表現可能である[1]。MPEG-7を使用したアプリケーション間の共働を可能にするためには、MPEG-7において構成されるDおよびDSのセットを適正な時点で所定のポイントにおいて決定する必要がある。DDLによってMPEG-7のDSをまえもって定められたセット以上にさらに拡張し続けることができる一方、これら新たな(拡張された)DSの意味はそれを設計した人以外の誰にもわからない。

30

【0002】

本発明の目的および利点

請求項1記載の方法によれば、バイナリフォーマットにおけるマルチメディア記述スキームのユーザ固有またはアプリケーション固有の拡張が提供され、その際、まえもって定義されているたとえばMPEG-7の記述スキームおよび拡張された記述スキームの双方に共通の記述スキームの部分におけるバイナリフォーマットが保存される。

40

【0003】

従来技術によるマルチメディア記述スキームのためのバイナリフォーマットすなわちいわゆるBiMにより、テキストによる記述をバイナリ形式で表すことができる。DおよびDSのセットがたとえばMPEG-7のような標準で決定されたならば、定義されたそれらのDおよびDSを原則的にはさらに拡張することができるけれども、拡張されたDまたはDSのセマンティクスはその作成者にしかわからない。それにもかかわらず固有のDやDSをたとえばMPEG-7などのDやDSといっしょに使用する多数のアプリケーションのために、両方の種類に対し同じバイナリフォーマットを使用するのが非常に重要となる。基本的にBiMは、MPEG-7のDDLをベースとしているかぎりいずれの種類

50

およびDSにも使用することができる。しかし現行のB i MをM P E G - 7のDまたはDSに適用し、かつM P E G - 7のDまたはDSをベースとする拡張されたDSに適用すると、両方のDSのバイナリフォーマットは両方の事例において同一のDSの部分であっても異なるものになってしまう。

【0004】

本発明の利点は、まえもって定義されたたとえばM P E G - 7のDSと拡張されたDSの同一部分のバイナリフォーマットを同じままに保つことのできる手段が提供されることであり、その結果、拡張部分に対するバイナリフォーマットだけが異なるようになる。このことはマルチメディアデータバイナリフォーマットに拡張ツリーブランチコードを含めることによって達成される。

10

【0005】

本発明のさらに別の観点として挙げられるのは、拡張ツリーブランチコード ( tree branch code ) T B C をポジションコードといっしょに使用できることである。

【0006】

本発明のさらに別の観点によれば、M P E G - 7固有の拡張ツリーブランチコード ( T B C ) をツリーブランチコード ( T B C ) テーブルに付加的にまたは択一的に含めることができる。このM P E G - 7固有の拡張T B CをM P E G - 7標準のDおよびDSの拡張に使用することができ、それらは将来のバージョン2, 3等で定義されようである。この種のM P E G - 7固有の拡張T B Cを含めることによって、所定の範囲まではM P E G - 7標準のフォワードコンパチビリティ ( forward compatibility ) が許容される。ここでいうフォワードコンパチビリティとは、M P E G - 7バージョン1のデコーダは、M P E G - 7バージョン2+のDSのうち少なくともM P E G - 7バージョン1のDSと合致している部分をデコーディングできるということである。さらに既存のM P E G - 7バージョン1の内容を、バージョン1のDSから派生された将来のM P E G - 7のDSに非常にたやすく組み込むことができる。

20

【0007】

図面

図面には従来技術を引き合いに出しながら本発明の実施形態が描かれており、以下でそれらの実施形態について詳しく説明する。

【0008】

図1は、ある特定の例に関するM P E G - 7記述のツリー表現を示す図である。

30

【0009】

図2は、M P E G - 7記述のツリー表現の一例を示す図である。

【0010】

図3は、B i Mフラグメントの構造を示す図である。

【0011】

図4は、B i Mフラグメント内部の種々の状態に関するカレントモードのポジションを示す図である。

【0012】

図5は、各要素に関するポジションのビット表現をオカレンスとともに示す図である。

40

【0013】

図6 . aは、ルートノードに関して絶対パスを示す図である。

【0014】

図6 . bは、トップレベルノードに関して絶対パスを示す図である。

【0015】

図6 . cは、カレントノードに関して相対パスを示す図である。

【0016】

図7は、M P E G - 7のDSと拡張M P E G - 7のDSのツリー表現を示す図である。

【0017】

図8は、あるDSノードにおいて単一の拡張T B Cを使用する例を示す図である。

50

## 【 0 0 1 8 】

図 9 は、ある D S ノードにおいて複数の拡張 T B C を使用する例を示す図である。

## 【 0 0 1 9 】

本発明の詳細な説明

本発明の詳細について論じる前に、いくつかの定義殊に M P E G - 7 で使用される定義を示しておく。

## 【 0 0 2 0 】

現在開発中の M P E G - 7 標準のコンテキストにおいて、マルチメディア環境におけるオーディオビジュアルデータを記述するための記述構造が定義されている。したがって記述子 (Description, D) および記述スキーム (Description scheme, D S) が、いわゆる記述定義言語 (Description Definition Language, DDL) を用いて定義されている。本明細書の以降のコンテキストにおいて、以下の定義が使用される：

・データ Data：データはオーディオビジュアル情報であって、これは記憶やコーディング、表示、伝送、媒体あるいは技術にかかわらず、M P E G - 7 を使用して記述される。

## 【 0 0 2 1 】

・フィーチャ Feature：フィーチャは何かを誰かに知らせるデータにおいて他との区別を示す特徴である。

## 【 0 0 2 2 】

・記述子 Descriptor (D)：記述子はフィーチャの表現である。記述子によってフィーチャ表現のシンタックスおよびセマンティクスが定義される。

## 【 0 0 2 3 】

・記述子値 Descriptor Value (D V)：記述子値は、目下のデータを表す所定のデータセット (またはそのサブセット) に対する記述子のインスタンスーションである。

## 【 0 0 2 4 】

・記述スキーム Description Scheme (D S)：記述スキームは、記述子 (D) および記述スキーム (D S) の両方とすることのできる各コンポーネント間の関係の構造およびセマンティクスを指定する。

## 【 0 0 2 5 】

・記述：1つの記述は、D S (構造) とデータを記述する記述子値 (インスタンスーション) のセットから成る。

## 【 0 0 2 6 】

・コーディングされた記述：コーディングされた記述は、圧縮効率、フォールトトレランス、ランダムアクセス等のような関連する要求を満たすために エンコーディングされた記述 である。

## 【 0 0 2 7 】

・記述定義言語 (D D L)：記述定義言語は、新たな記述スキームおよび場合によっては記述子を生成することのできる言語である。さらにこれによって既存の記述スキームの拡張や変形も可能である。

## 【 0 0 2 8 】

記述の一番下のレベルは記述子である。これによって1つまたは複数のデータのフィーチャが定義される。これは個々の D V とともに、データの固有の部分を実際に記述するために使用される。次に上のレベルは記述スキームであり、これには少なくとも2つまたはそれよりも多くのコンポーネントおよびそれらの関係が含まれている。コンポーネントは記述子または記述スキームのどちらであってもよい。最も上のレベルは記述定義言語である。これは M P E G - 7 の D と D S を定義するために使用されるし、さらにたとえば M P E G - 7 のコンテキストで定義された既存の D S を拡張するなどしてユーザ固有またはアプリケーション固有の新たな D S を定義するためにも使用できる。

## 【 0 0 2 9 】

M P E G - 7 の記述を表現するために2つのオプションがある。最初のオプションは、extended Markup Language (XML) を使用したテキスト表現である。この表現は人間が読む

10

20

30

40

50

ことができるけれども、かなり冗長になる可能性がある。2番目のオプションはバイナリ表現であり、これはMPEG-7データのためのバイナリフォーマットである（Binary format for MPEG-7 Data, BiM）。BiMは人間には読めないが、帯域幅について効率的なMPEG-7データの記憶や伝送、DSの所定のコンポーネントへのランダムアクセスが可能となるし、さらにDSコンポーネントのフレキシブルな伝送順序が許容される。以下のセクションで、MPEG-7記述のテキストフォーマットとバイナリフォーマットについてさらに詳しく説明する。

【0030】

MPEG-7データのためのテキストフォーマットはXML言語をベースとしている。これはいわゆる「XMLタグ」から成り、タグ間でデータの構造と意味を定義する。記述は階層的に組み立てられており、DSコンポーネントの開始を定義する各オープニングXMLタグごとに、DSコンポーネントの終了を定義するそれに対応するクロージングタグが設けられている。非常に簡単なDSの一例を以下に示す。DSの定義はDDLを使用することで行われ、これはMPEG-7のMDSパートに示されている[4]。

【0031】

【外1】

```
<CreationInformation>
```

```
  <Creation>
```

```
    <Title type="original">
```

```
      <TitleText xml:lang="en">Help</TitleText>
```

```
    </Title>
```

```
    <Creator>
```

```
      <Role CSName="MPEG_roles_CS" CSTermID="47">
```

```
        <Label xml:lang="en">presenter</Label>
```

```
      </Role>
```

```
    <PersonGroup>
```

```
      <Name>The Beatles</Name>
```

```
    </PersonGroup>
```

```
  </Creator>
```

```
</Creation>
```

```
<Classification>
```

```
  <CountryCode>de</CountryCode>
```

```
  <Language>
```

```
    <LanguageCode>en</LanguageCode>
```

```
  <CountryCode>en</CountryCode>
```

```
  </Language>
```

```
  <Genre CSName="Escort2_4:Content" CSTermId="2">
```

```
    <Label xml:lang="en">Rock</Label>
```

```
  </Genre>
```

```
</Classification>
```

```
</CreationInformation>
```

10

20

30

40

50

## 【 0 0 3 2 】

括弧 ("<...>") 内のテキストは、オープニングXMLタグおよびクロー징XMLタグを表す。タグ間のテキストは記述のデータ値を表す。上述の例では、英国のグループ "The Beatles" の "Help" という名称のオーディオトラックのタイトル、プレゼンタならびにジャンルが記述されている。この記述も図1に示したように記述ツリーとして描くことができる。この場合、実際のデータはツリーのリーフ中に含まれる一方、個々のブランチノードによって構造が与えられる。このツリー表現は、MP EG - 7のバイナリフォーマットを理解するうえで非常に有用であり、これについて以下のセクションで説明する。

## 【 0 0 3 3 】

本発明において引き合いに出すマルチメディア記述のためのバイナリフォーマットに関する従来技術はMP EG - 7 Bi Mとしており、これについては[2]に詳しく記載されている。ここではバイナリフォーマットについて、本発明を理解するうえで必要な範囲で説明する。

10

## 【 0 0 3 4 】

一般に、各MP EG - 7記述はいわゆるMP EG - 7ルート要素からスタートし、これには1つまたは複数のMP EG - 7記述スキームを含めることができる。ルート要素の子であるMP EG - 7のDSを以下では「トップレベル要素」と称する。すべてのMP EG - 7記述は階層的に定義されているので、それらを図2に示されているように記述ツリーとして捉えることができる。

## 【 0 0 3 5 】

記述のツリー表現を使用しているので、以下の記述ではルート要素とトップレベル要素も「ルートノード」および「トップレベルノード」と称する。このツリー表現を、バイナリフォーマットのデコーディングの仕方をも記述するためにも使用する。ルートノード情報はデコーダによって受け取られるので、ビットストリームの残りはいわゆるBi Mフラグメントから成る。各フラグメントは基本的に図3に示されているようになる。

20

## 【 0 0 3 6 】

図示されているように、Bi Mフラグメントは主として4つのパートから成り、以下でこれらについて手短かに説明する。第1のパートはいわゆる「ナビゲーションコマンド」であって、これは記述スキームのどのインスタンス化パートを次に受け取るかをデコーダに通知する。3種類のアドレッシング記述要素が可能である：

30

- ・デコーダが目下存在するノードに対し相対的なアドレッシング
- ・デコーダが目下存在するDSの目下のトップレベルノードに対する絶対的なアドレッシング
- ・ルートノードに対し相対的な絶対アドレッシング

Bi Mフラグメントとの次のパートはナビゲーションパスであって、つまりこれはデコーダがBi Mフラグメント中に含まれる情報を受け取りたいツリーノードに関する絶対アドレスまたは相対アドレスである。トップレベルDSのアドレス(識別子またはIDとも称する)は8bitのコードから成り、これは以下のテーブルで指定されている。ルートノードを離れてMP EG - 7ビットストリームを終わらせることができるよう、トップレベルDS IDに加えて、「イクジットルート "exit root"」コマンドを通知する特定のビットコードが存在する。トップレベルDSの後続のサブ要素(子)のためのアドレスはローカルパスコードを使用して計算され、以下の説明ではこれをツリーブランチコードとも称する。

40

## 【 0 0 3 7 】

Bi Mフラグメントの第3のパートはいわゆる「コンテンツマニピュレーションコマンド "content manipulation command"」であって、これはデコーダに対し、どの種類のコンテンツマニピュレーションをナビゲーションによりたたいま送られたノードに対してデコーダが受け取るのかを通知する。この場合、3つの可能なコマンドが設けられており、つまりそれらのコマンドとはサブツリーの追加 "add"、サブツリーの更新 "update"、ならびにサブツリーの削除 "delete" である。サブツリーには、デコーダが目下存在するノ

50

ード要素が含まれている。そしてコマンドが実行される。追加 "add" または更新 "update" の場合、サブツリーのコンテンツは B i M フラグメントの第 4 のコンポーネント内に含まれる。削除 "delete" の場合、B i M フラグメントの第 4 のコンポーネントは空である。コマンドが実行された後（つまりサブツリー情報がデコーディングされるかまたは既存のサブツリーが削除された後）、デコーダは暗黙的にその親のノードに戻る。

【 0 0 3 8 】

B i M フラグメントのコンテンツマニピュレーションセクションは 2 つの部分から成り、すなわちそれらはコンテンツマニピュレーションコマンドとサブツリーバイナリ表現である。ナビゲーションパスを含むナビゲーションコマンドの終了後、コンテンツマニピュレーションコマンドを待つ。同様に、サブツリーのバイナリ表現を含むコンテンツマニピュレーションコマンドの終了後、ナビゲーションコマンドを待ち、これは新たな B i M フラグメントの開始である。デコーダの初期化時はカレントノードがルートノードにデフォルトとして設定され、ナビゲーションコマンドを待つ。B i M 中に定義されるコンテンツマニピュレーションコマンドに関するコードテーブルを表 1 として以下に示す。

【 0 0 3 9 】

【表 1】

コンテンツマニピュレーションコマンドのコードテーブル		
符号語	コマンド名	仕 様
000	---	予約
001	AddSubtree	送信されたサブツリーに関してカレントノードが空であるときのみサブツリーを含むカレントノードに対しコンテンツを追加
010	UpdateSubtree	既存のコンテンツを上書きすることになってもサブツリーを含むカレントノードに対しコンテンツを更新
011	DeleteSubtree	サブツリーを含むカレントノードを削除
100 - 110	---	予約
111	ESCAPE	将来の拡張用に予約

表 1 コンテンツマニピュレーションコマンドのコードテーブル

【 0 0 4 0 】

コマンドがカレントノードに適用される。ナビゲーション後のカレントノード（記述ツリー中の個々のポジション）は、ターミネーション T B C（図 4 参照）を除いて完全なナビゲーションパスにより参照されるノードである。この場合、コンテンツマニピュレーションコマンド（追加 add、更新 update、削除 delete）は、カレントノード自身を含むサブツリーに適用される。コンテンツマニピュレーションコマンドが発行された後、カレントノードがナビゲーション後のノードの父 father にセットされる。

【 0 0 4 1 】

コマンドコードテーブルには以下のコードが含まれている：

- ・ AddSubtree：ツリー中のカレントポジションを含むサブツリーを追加。このコマンドに続いてサブツリー全体のバイナリ表現がくる。AddSubtree コマンドが発行された後、カレントノードがナビゲーション後のカレントノードの父 father にセットされる。

【 0 0 4 2 】

- ・ UpdateSubtree：ツリー中のカレントポジションを含むサブツリーを更新。このコマンドに続いてサブツリー全体のバイナリ表現がくる。UpdateSubtree コマンドが発行された後、カレントノードがナビゲーション後のカレントノードの父 father にセットされる。

## 【 0 0 4 3 】

・DeleteSubtree：ツリー中のカレントポジションを含むサブツリーを削除。DeleteSubtree コマンド発行後、カレントノードがナビゲーション後のカレントノードの父にセットされる。

## 【 0 0 4 4 】

カレントノードが単純型の要素または属性であるならば、コマンド直後にバイナリコンテンツ表現が続く。コマンド実行後、デコーダはナビゲーションモードに戻る。

## 【 0 0 4 5 】

BiMフラグメントのナビゲーションセクションは2つのパートに分割され、それらはすなわちナビゲーションコマンドとナビゲーションパスである。ナビゲーションコマンドは2bitの固定ビット長であり、それに続いて可変長のナビゲーションパスが設けられる。ナビゲーションパスは連続的なツリーブランチコード(TBC)から成り、そこにおいて各TBCは記述ツリー内部のあるノードにおけるパスのフラクショナルを記述する。したがってナビゲーションパスはローカルナビゲーションパスのシーケンスであり、これはツリーを通るパス中に含まれる各ノードにおけるツリーブランチを指定する(図6参照)。ナビゲーションコマンドによって、表2中で指定されているようにナビゲーションパスをどのように解釈すべきかを指定する。

## 【 0 0 4 6 】

## 【表2】

ナビゲーションコマンドのコードテーブル		
コード	ナビゲーションパス	ナビゲーションコマンド
00	ID_of_TopLevelElement + <TBCs>	絶対アドレッシングモードでMPEG-7記述のルートノードからトップレベルノードのIDおよび後続のTBCにより指定されたノードへナビゲート
01	<TBCs>	絶対アドレッシングモードでカレントトップレベルノードからTBCにより指定されたノードへナビゲート
10	<TBCs>	相対アドレッシングモードでカレントノードからTBCにより指定されたノードへナビゲート
11	---	予約

表2 ナビゲーションコマンドのコードテーブル

## 【 0 0 4 7 】

ナビゲーションコマンドのパラメータでありビットストリーム中のナビゲーションコマンドコードに続く<TBC>は、インスタンス化された記述のベースとなるMPEG-7スキーマから生成される。スキーマ定義によって、種々のTBCによりアドレッシングされるべき記述中の1つのノードの可能な子childrenが指定される。スキーマ定義中、子の可能な型およびそれらの最大オカレンス数が指定される。個々の記述ノードに関してインスタンス化された要素内のツリーブランチに従い、TBCは子要素の仕様(SchemaBranch)および1回以上インスタンス化可能ならばポジションが含まれる。1つのナビゲーションパス全体は、連続的なツリーブランチコード(Tree Branch Code, TBC)の順序づけられたリストとして構成される。

## 【 0 0 4 8 】

コードテーブルおよびそのTBCは、複合型complexTypeの指定されたすべての要素または記述スキーム中の指定された複合型complexTypeについて固有のものである。したがってナビゲーションのために、インスタンス化された要素の型により使用されるコード

テーブルも決定される。したがって同じ複合型のノードに関するすべての要素は同じコードテーブルを有する。1つのノードにおける各々の子要素および属性はそれを参照する符号語に割り当てられる。これらの符号語はツリーブランチコード(Tree-Branch-Code, TBC)と呼ばれ、以下で説明する規範的アルゴリズムにより記述スキームから生成される。

【0049】

上述のようにTBCは2つのパートによって構成されている。すなわち1つはスキーマブランチ SchemaBranch であって、これはスキーマ中で指定された種々の子要素の定義に対応し、もうひとつはポジションであって、これはスキーマ定義に従いこの子の多数のオカレンスが可能であるときにのみ存在する。記述ツリー内のナビゲーションのため、父のノードを参照するためのコードも設けられている。さらに、パスをターミネートしコマンド

10

モードへ暗黙的にスイッチするためにコードが用いられる。表3にはこの種のコード

【表3】

要素または複合型の名前		
TBC		ツリーブランチ
#SchemaBranch	#position	
000	--	父への参照
001	Pos. Code	第1の子への参照
010	Pos. Code	第2の子への参照
011	...	
100		
...		
111	--	このTBCはナビゲーションパスのターミネーションを通知。このTBCの受信後、デコーダは以降のコンテンツマニピュレーションコマンドを受け取るためのコマンドモードにあるものとする。

20

30

表3 要素または指定された複合型のツリーブランチコードテーブルのプロトタイプ

【0051】

TBCの割り当ては下記のルールに従い実行される：

- ・種々の子の数は記述スキームから既知であり、これにより以下のルールに従いTBCのスキーマブランチ SchemaBranch ビットフィールドの長さが定義される

40

【0052】

【外2】

$$\lceil \lg(\# \text{ of child elements} + \# \text{ of attributes} + 2) \rceil,$$

ただし  $\lg$  は2を底とする対数、 $\lceil x \rceil$  は  $y \geq x$  が真である最小の整数を規定

【0053】

- ・オール0のスキーマブランチ SchemaBranch コードは、父のノードへの参照に常に割り当てられる。

50

【 0 0 5 4 】

・子を参照するスキーマブランチコードは、ISO/IEC 15938 の個々のパートにおいて指定された指定されたその要素または型のスキーマ定義中に定義された順序でシーケンシャルに割り当てられる。

【 0 0 5 5 】

・オール1のスキーマブランチコードは、パスをターミネートしコマンドモードにスイッチするために常に使用される。

【 0 0 5 6 】

・1つの子が多数のオカレンスを持つことができるならば、フィールド<#position>により以下のルールに従いインスタンス化された記述中の子の個数が指定される：フィールド #position のビット表現は、スキーマ定義におけるmaxOccur の個数を表すために必要とされる最小ビット数である。このビット数が3を超えるならば、最初のビット ( E x t ) は4つのビットの拡張が存在するのかを指定する ( 図5参照 )。この4つのビットは、以降で拡張が 8 b i t である点を除いて先行する4つのビットと同じセマンティクスをもち、その 8 b i t のうち最初のビットは拡張を通知するために予約されている。代替グループにグループ分けされた要素は付加的に、TBCの後の属性としてこの代替グループ内のアドレスをコーディングしなければならない (これは上述の例の図面には示されていない)。この付加的なコードは SubstGrpSelect と呼ばれる。代替グループ内の各要素は S ubstGrpSelect に割り当てられる。要素を参照する SubstGrpSelect コードは、ゼロから始まってスキーマ定義中に定義された順序でシーケンシャルに割り当てられる。

10

20

【 0 0 5 7 】

単純型 simpleType の属性、要素および指定された単純型 simpleType は、1つのリーフテーブルによって表される。この種のリーフテーブルのプロトタイプを表4として示す。

【 0 0 5 8 】

【表4】

要素または単純型の名前	
TBC	ツリーブランチ
	ナビゲーションのターミネーションおよびコマンドモードへのスイッチ

30

表4 要素または指定された単純型のツリーブランチコードテーブルのプロトタイプ

【 0 0 5 9 】

1つのパスはTBCを連結することによって作成される。リーフノードを除いてすべてのノードはその父への参照をもっているので、相対パスを使用するならば記述ツリー階層において上へ向かって動かすこともできる。

40

【 0 0 6 0 】

絶対ナビゲーションおよび相対ナビゲーションは、後続のコンテンツマニピュレーションコマンドを適用するツリー構造内のインスタンスノードの位置決定を目的としている。絶対パスは、インスタンス構造のMPEG-7記述ルートまたはカレントノードのトップレベルからターゲット要素までの経路を表すのに対し、相対パスは個々のノードのカレントポジションから始まるパスを指定しようというものである ( 図6参照 )。

【 0 0 6 1 】

本発明はナビゲーションモードの拡張に係わるものであって、殊にいわゆるツリーブランチコードの方法論を拡張することに係わる。以下のセクションでこのことについて詳しく

50

説明する。

【 0 0 6 2 】

マルチメディア記述スキームのための従来技術のバイナリフォーマットすなわちいわゆる B i M によって、テキスト形式の記述をバイナリ形態で存在させることができる。たとえば M P E G - 7 のような標準において D および D S のセットが確定されると、定義された D および D S は基本的にさらに拡張可能であるが、拡張された D または D S のセマンティクスはその作者しかわからない。それにもかかわらず、固有の D と D S をたとえば M P E G - 7 の D および D S といっしょに使用する多くのアプリケーションにとって、両方の種類ののために同じバイナリフォーマットを使用するのが非常に重要になる。原則的に B i M は、M P E G - 7 の D D L をベースとするかぎり、D と D S のどちらの種類にも使用することができる。しかし現行の B i M を M P E G - 7 の D または D S に適用し、かつ M P E G - 7 の D または D S をベースとする拡張された D S にも適用すると、両方の事例において同一である D S のパートであっても両方の D S のバイナリフォーマットが異なってしまう。図 7 にはその事例が示されている。この図には単純な M P E G - 7 の D S と拡張された D S が示されている。拡張された D S は、1 つのノード "DiCo" をノード "Lang" に加えることで M P E G - 7 の D S から派生している。しかしながら各ノードに関する T B C テーブルはその子の目下の個数に左右されるので、ノード "LaCo" のアドレスは両方の事例で異なる。つまり M P E G - 7 の事例では ( ノード "Clas" に対し相対的に ) "010 001 111" となる。拡張された D S についてアドレスは ( ノード "Clas" に対し相対的に ) "010 010 111" となる。ここでわかるように、拡張がなされているノードの子のアドレスは変わっており、したがって M P E G - 7 のバイナリフォーマットのその部分も変化することになる。

10

20

【 0 0 6 3 】

本発明の利点は、まえもって定義されている D S たとえば M P E G - 7 の D S と拡張された D S の同一のパートにおけるバイナリフォーマットを同じままにしておく手段が得られるので、拡張パートに対するバイナリフォーマットのみしか異ならないようになる。このことは B i M 中に「拡張ツリーブランチコード "extension tree branch code" を含ませることにより達成され、これによりこのコード以降はまえもって定義された D S についてのいくつかの拡張が続くことになることが通知される。これにより拡張された D S を 2 つのパートに分けることができる。この場合、一方のパートはベースの D S と同一であり、すなわちまえもって定義された (たとえば M P E G - 7 の) D S であり、拡張された D S はこれをベースとしており、他方のパートはベースの D S について D S の拡張を記述している。この場合、第 1 のパートは、1 つまたは複数の B i M フラグメントおよび第 2 のパートによって表すことができる。このようにすることで、拡張されたコードにおいて M P E G - 7 と合致したパートを符号変換する必要なく抽出するのが非常に簡単になる。さらに、拡張された D S のユーザにとって拡張 D S の一部分である M P E G - 7 の D S へのアクセスおよび利用も格段に簡単になり、この場合も内容全体を符号変換する必要がない。それぞれまったく異なるニーズをもつ多数のアプリケーションが存在するが、それらは部分的に M P E G - 7 の D S を使用するけれども、それらのニーズを満たす目的でおそらくは拡張を望むであろうから、M P E G - 7 と非 M P E G - 7 の D S の簡単ななコンビネーションは将来、非常に重要になる。

30

40

【 0 0 6 4 】

本発明の全般的な着想は、「拡張ツリーブランチコード "extension tree branch code" 」を B i M に含めることであり、これによってこのコード後には拡張された事前に定義された (たとえば M P E G - 7 などの) D S の拡張パートが続くことが通知される。したがって以下で示すように、項目「拡張ツリーブランチコード "extension tree branch code" 」が複合型のツリーブランチコードに関する表 3 に含まれる。

【 0 0 6 5 】

【表 5】

要素または複合型の名前		
TBC		ツリーブランチ
#SchemaBranch	#position	
000	--	父への参照
001	Pos. Code	第1の子への参照
010	Pos. Code	第2の子への参照
011	...	
100		
...		
110	--	このTBCは、このブランチ後にまえて定められたMPEG-7のDまたはDSの拡張が続くこと、それらのセマンティクスおよび意味は現在の標準では指定されておらず拡張DSのユーザまたはアプリケーションに依存することを通知。
111	--	このTBCはナビゲーションパスのターミネーションを通知。このTBCの受信後、デコーダは以降のコンテンツマニピュレーションコマンドを受け取るためのコマンドモードにあるものとする。

10

20

表5 まえて定められたDまたはDSに対するユーザまたはアプリケーションに固有の拡張に関する「拡張ツリーブランチコード」を含む要素または指定された複合型のツリーブランチコードテーブルのプロトタイプ

【0066】

30

指定された複合型または要素に対しこのようなTBCテーブルを形成する目的で、TBCを割り当てるための本来のルールを以下のとおり変更する：

・種々の子の個数は記述スキームから知られており、以下のルールに従いTBCにおけるSchemaBranch ビットフィールドの長さが定義される。

【0067】

【外3】

$$\lceil \lg(\# \text{ of child elements} + \# \text{ of attributes} + 3) \rceil,$$

ただし  $\lg$  は2を底とする対数、 $\lceil x \rceil$  は  $y \geq x$  が真である最小の整数を規定

40

【0068】

・オール0のSchemaBranchコードは常に父のノードへの参照に割り当てられる。

【0069】

・子を参照するSchemaBranchコードは、ISO/IEC 15938の個々のパートにおいてこの指定された要素または型のスキーマ定義中で定義されている順序でシーケンシャルに割り当てられる。

【0070】

・オール1のSchemaBranchコードは、パスのターミネーションおよびコマンドモードへ

50

のスイッチのために常に使用される。

#### 【 0 0 7 1 】

・ 1つの子が複数のオカレンスをもつことができるならば、あるいは複数のオカレンスをもつことのできる拡張 T B C であるならば、フィールド < #position > によって以下のルールに従いインスタンス化された記述中の子の数が指定される：フィールド #position のビット表現はスキーマ定義中の maxOccur の個数を表すのに必要とされる最小ビット数である。このビット数が 3 を超えたならば、最初のビット ( E x t ) は 4 ビットの拡張が存在するかを指定する ( 図 5 参照 )。これら 4 つのビットは、それ以降は拡張が 8 ビットであり最初のビットは拡張を通知するために予約されている点を除いて、先行の 4 つのビットと同じである。

10

#### 【 0 0 7 2 】

このように変更することで、まえもって定義された (たとえば M P E G - 7 などの) D および D S のためにも、M P E G - 7 をベースとする拡張された D および D S のためにも、B i M を使用することができるようになり、さらに両方の D S において M P E G - 7 と合致したパートについてバイナリフォーマットを同じままに保つことができる。しかも、まえもって定義された (たとえば M P E G - 7 などの) D および D S と拡張された D および D S のために、同じ B i M エンコーダおよびデコーダを使用することができる。図 8 には、M P E G - 7 ベースの D S と拡張されたパートの拡張された D S インスタンスのツリー表現を描くことで、拡張 T B C の利用に関する一例が描かれている。ここに示されているように、2 つのサブツリーの各々を 1 つまたは複数の B i M フラグメントによって表すことができる。しかし M P E G - 7 と合致した D S のパートの B i M フラグメントのフォーマットは M P E G - 7 の D S 事態のものと同じである。

20

#### 【 0 0 7 3 】

本発明の別の観点によれば、拡張ツリーブランチコード T B C をポジションコードといっしょに使用することができる (たとえば表 5 を参照)。このポジションコードは何度も現れてよい D S ノードの子の場合に使用される。表 5 に定義されているようにポジションコードがディスエーブル (表のエントリが "--") である拡張 T B C であれば、1 つの D S ノードにおいてただ 1 つの拡張 T B C だけしか現れてはならない。オプションとしてポジションコードをイネーブルにすることもでき、そうすることで 1 つの D S ノードにおいて 2 つ以上の拡張 T B C が許可されるようになる。これはごく小さい変更しか生じない D S の拡張の事例に対して殊に有利である。図 9 にはやはり、個々の拡張 D S パートのツリー表現を描くことで多数の拡張 T B C の一例が示されている。やはりすべてのサブツリーは 1 つまたは複数の B i M フラグメントによって表現されており、M P E G - 7 の D S の B i M フラグメントのフォーマットが保たれることになる。

30

#### 【 0 0 7 4 】

本発明の別の観点によれば付加的または択一的に、M P E G - 7 固有の拡張ツリーブランチコード ( T B C ) をツリーブランチコード ( T B C ) テーブルに含めることができる。この M P E G - 7 固有の拡張 T B C を、将来のバージョン 2 , 3 等で定義される可能性の高い M P E G - 7 標準の D および D S の拡張に使用することができる。このような M P E G - 7 固有の拡張 T B C を含めることによって、所定の範囲まで M P E G - 7 標準のフォワードコンパチビリティが許容される。ここでいうフォワードコンパチビリティとは、M P E G - 7 バージョン 1 のデコーダは、M P E G - 7 バージョン 2 + の D S のうち少なくとも M P E G - 7 バージョン 1 の D S と合致した部分をデコーディングできるということである。さらに、既存の M P E G - 7 バージョン 1 の内容をバージョン 1 の D S から派生した将来の M P E G - 7 の D S に組み入れるのが非常に容易になる。

40

#### 【 0 0 7 5 】

参考文献

[ 1 ] ISO/IEC, "Introduction to MPEG-7", Doc. ISO/IEC JTC1/SC29/WG11 N3751, La Baule, France, October 2000.

[ 2 ] ISO/IEC, "Text of ISO/IEC CD 15938-1 Information technology - Multimedia

50

content description interface: Systems", Doc. ISO/IEC JTC1/SC29/WG11 N3701, La Baule, France, October 2000.

[ 3 ] ISO/IEC, "Text of ISO/IEC CD 15938-2 Information technology - Multimedia content description interface: Description Definition Language", Doc. ISO/IEC JTC1/SC29/WG11 N3702, La Baule, France, October 2000.

[ 4 ] ISO/IEC, "Text of ISO/IEC CD 15938-5 Information technology - Multimedia content description interface: Description Definition Language", Doc. ISO/IEC JTC1/SC29/WG11 N3705, La Baule, France, October 2000.

【図面の簡単な説明】

【図 1】 ある特定の例に関する M P E G - 7 記述のツリー表現を示す図である。

10

【図 2】 M P E G - 7 記述のツリー表現の一例を示す図である。

【図 3】 B i M フラグメントの構造を示す図である。

【図 4】 B i M フラグメント内部の種々の状態に関するカレントモードのポジションを示す図である。

【図 5】 各要素に関するポジションのビット表現をオカレンスとともに示す図である。

【図 6】 6 . a はルートノードに関して絶対パスを示す図であり、6 . b はトップレベルノードに関して絶対パスを示す図であり、6 . c はカレントノードに関して相対パスを示す図である。

【図 7】 M P E G - 7 の D S と拡張 M P E G - 7 の D S のツリー表現を示す図である。

【図 8】 ある D S ノードにおいて単一の拡張 T B C を使用する例を示す図である。

20

【図 9】 ある D S ノードにおいて複数の拡張 T B C を使用する例を示す図である。

【図 1】

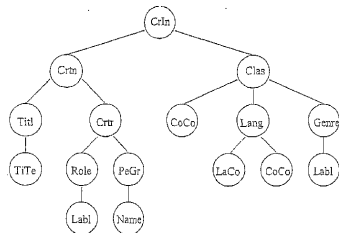
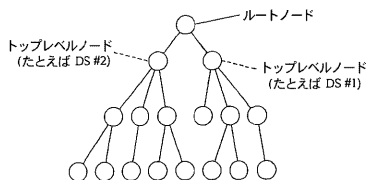
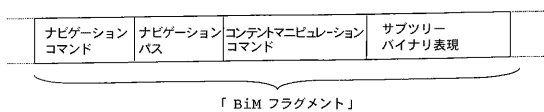


Figure 1

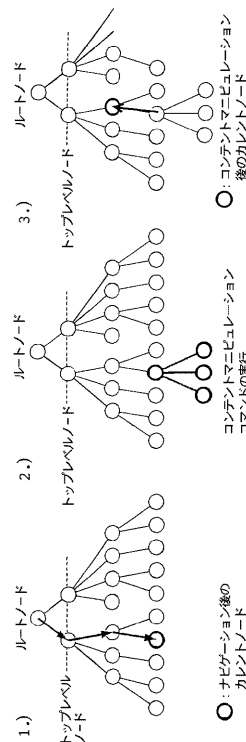
【図 2】



【図 3】



【図 4】



【 図 5 】

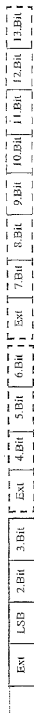
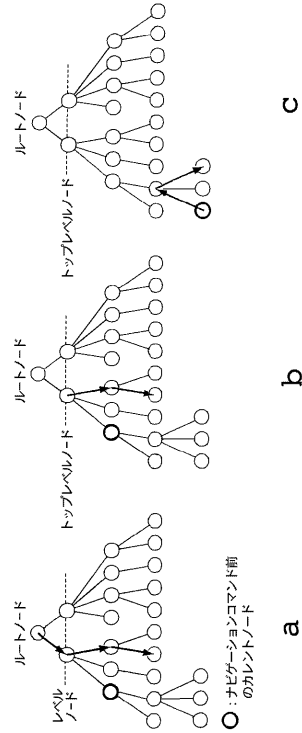


Figure 5

【 図 6 】



【 図 7 】

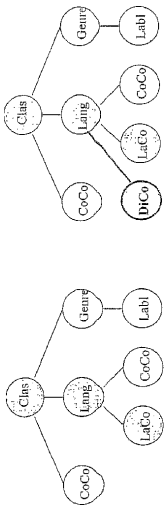


Figure 7

【 図 8 】

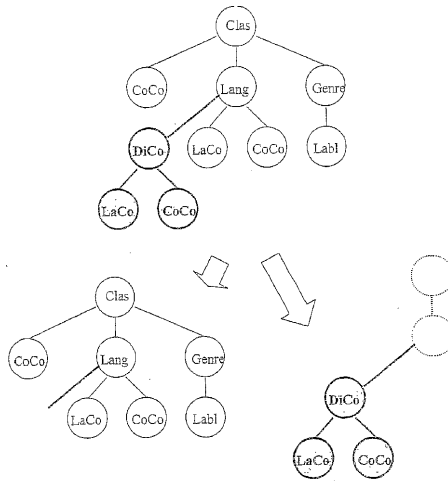


Figure 8

【 図 9 】

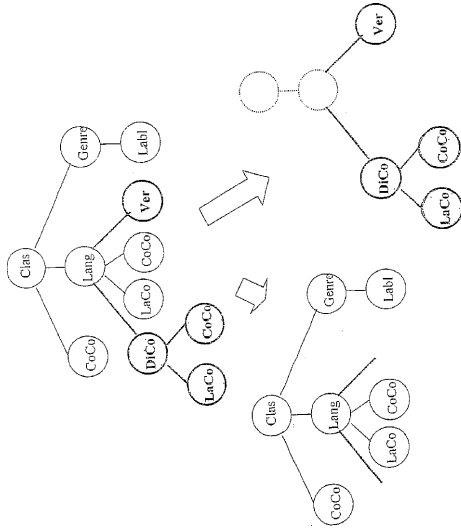


Figure 9

---

フロントページの続き

(72)発明者 ミヒヤエル ヴォルボルン  
ドイツ連邦共和国 ハノーファー ザントシュタインヴェーク 10

合議体

審判長 小曳 満昭

審判官 本郷 彰

審判官 飯田 清司

(56)参考文献 Neil Day, Introduction to MPEG-7, フランス, ISO/IEC, 2000年10月, JTC1/SC29/WG11/N3751 Claude Seyrat, Text of ISO/IEC CD 15938-1 Information Technology - Multimedia content description interface - Part 1 Systems, フランス, ISO/IEC, 2000年10月, JTC1/SC29/WG11/N3701

(58)調査した分野(Int.Cl., DB名)  
G06F17/30, G06F17/21, G06F12/00