

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4197320号

(P4197320)

(45) 発行日 平成20年12月17日(2008.12.17)

(24) 登録日 平成20年10月10日(2008.10.10)

(51) Int.Cl. F I
H03M 7/30 (2006.01) H03M 7/30 Z
G06F 17/21 (2006.01) G06F 17/21 501T

請求項の数 14 (全 12 頁)

(21) 出願番号	特願2004-528352 (P2004-528352)	(73) 特許権者	390039413
(86) (22) 出願日	平成15年7月7日(2003.7.7)		シーメンス アクチエンゲゼルシャフト
(65) 公表番号	特表2005-538436 (P2005-538436A)		Siemens Aktiengesellschaft
(43) 公表日	平成17年12月15日(2005.12.15)		ドイツ連邦共和国 D-80333 ミュンヘン ヴィッテルスバッハープラッツ 2
(86) 国際出願番号	PCT/DE2003/002274		Wittelsbacherplatz 2, D-80333 Muenchen, Germany
(87) 国際公開番号	W02004/017225	(74) 代理人	100061815
(87) 国際公開日	平成16年2月26日(2004.2.26)		弁理士 矢野 敏雄
審査請求日	平成17年1月17日(2005.1.17)	(74) 代理人	100094798
(31) 優先権主張番号	10231971.5		弁理士 山崎 利臣
(32) 優先日	平成14年7月15日(2002.7.15)		
(33) 優先権主張国	ドイツ(DE)		
(31) 優先権主張番号	10248758.8		
(32) 優先日	平成14年10月18日(2002.10.18)		
(33) 優先権主張国	ドイツ(DE)		

最終頁に続く

(54) 【発明の名称】 構造化された文章、特にXML文章の符号化／復号化のための方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

XML文章の伝送と符号化のための方法において、

符号化器が設けられており、該符号化器は以下のステップ(a~e)、すなわち

a) XML文章に所属するXMLスキーマを正規化するステップと、；この場合XMLスキーマの正規化は、以下の単純化ステップの1つを含んでおり、すなわち

1つの要素のみを含んだ1つのグループの単純化ステップ、；このグループは分解され、得られた要素は、分解されたグループのレベルでコンテンツモデルに入れられ、その際要素の属性値minOccursとmaxOccursが分解されたグループの相応の属性値と再グループ化される前の要素の積によって置換され、

属性値minOccurs = 0の要素を含んだ選択グループの単純化ステップ、；前記選択グループの属性値minOccursは、先行する値に依存することなく値0にセットされ、属性値minOccurs = 0を有していた要素は、属性値minOccurs = 1に割当てられ、

ネ스팅された選択グループの単純化ステップ、；1つの選択グループは、属性値minOccurs = maxOccurs = 1を含む別の選択グループを含み、それによって当該選択グループは分解され、内容がその上位に位置する選択グループに直接取り入れられ、その際タームグループ、要素、コンテンツモデル、属性値、minOccurs, maxOccurs及び選択グループがXMLスキーマ言語の正規化された記述で定められ、

10

20

- b) 正規化されたXMLスキーマをメタスキーマを用いて符号化するステップと、
- c) 符号化されたXMLスキーマを第1ビットストリームにおいて伝送するステップと、
- d) XML文章を所属のXMLスキーマを用いて符号化するステップと、
- e) 符号化されたXML文章を第2のビットストリームにおいて伝送するステップとを実施し、

さらに前記第1及び第2のビットストリームの受信のために復号化器が設けられており、該復号化器は以下のステップ(f~g)、すなわち、

- f) 第1のビットストリームにて伝送される符号化されたXMLスキーマをメタスキーマを用いて、正規化されたXMLスキーマへ復号化するステップと、；この場合前記正規化されたスキーマとメタスキーマは符号化の際に用いられたスキーマに相応しており、
- g) 第2のビットストリームにて伝送される符号化されたXML文章を、正規化されたXMLスキーマを用いて、正規化されたXMLスキーマのさらなる正規化を実施する事無く復号するステップとを実施することを特徴とする方法。

10

【請求項2】

構造化された文章のスキーマ定義の要素宣言及び/又は属性宣言が、次のように再構築される、すなわち匿名タイプ定義(AT0)が要素宣言及び/又は属性宣言から取り出され、相応の要素のもとで参照に用いられる名前及び/又はコードが得られるように再構築される、請求項1記載の方法。

【請求項3】

タイプネーム及び/又は要素ネーム及び/又は代入グループネームの代わりに、数値並びに数値とタイプネーム、及び/又は要素ネーム、及び/又は代入グループネームの間の割当てを含んだ1つ以上のテーブルのみが符号化される、請求項1または2記載の方法。

20

【請求項4】

タイプネーム及び/又は要素ネーム及び/又は代入グループネーム並びに、タイプネーム及び/又は要素ネーム及び/又は代入グループネームのリスト内の位置を含んだ1つ又は複数のリストが、タイプネーム及び/又は要素ネーム及び/又は代入グループネームの代わりに符号化される、請求項1から3いずれか1項記載の方法。

【請求項5】

タイプ、大域要素及び/又は代入グループの継承ツリーに対する情報が符号化され、その際各タイプが、マスタタイプに関するタイプコードと、記述タイプに係わる全てのタイプコードの長さに関する情報によって記述され、及び/又は、

30

各大域要素が、SBCの長さ及びSBC及び/又は代入グループの各要素と、代入コードの長さ及び代入コードによって記述される、請求項1から4いずれか1項記載の方法。

【請求項6】

符号化方法を用いてXML文章から生成された第1及び第2のデータストリームを復号化するための方法であって、

符号化器が設けられており、該符号化器は以下のステップ(a~e)、すなわち、

- a) XML文章に所属するXMLスキーマを正規化するステップと、；この場合XMLスキーマの正規化は、以下の単純化ステップの1つを含んでおり、すなわち

1つの要素のみを含んだ1つのグループの単純化ステップ、；このグループは分解され、得られた要素は、分解されたグループのレベルでコンテンツモデルに入れられ、その際要素の属性値minOccursとmaxOccursが分解されたグループの相応の属性値と再グループ化される前の要素の積によって置換され、

40

属性値minOccurs = 0の要素を含んだ選択グループの単純化ステップ、；前記選択グループの属性値minOccursは、先行する値に依存することなく値0にセットされ、属性値minOccurs = 0を有していた要素は、属性値minOccurs = 1に割当てられ、

ネ스팅された選択グループの単純化ステップ、；1つの選択グループは、属性値minOccurs = maxOccurs = 1を含む別の選択グループを含み、それによって当該選択グループは分解され、内容がその上位に位置する選択グループに直接取り入

50

れられ、その際タームグループ、要素、コンテンツモデル、属性値、minOccurs、maxOccurs及び選択グループがXMLスキーマ言語の正規化された記述で定められ、

- b) 正規化されたXMLスキーマをメタスキーマを用いて符号化するステップと、
- c) 符号化されたXMLスキーマを第1ビットストリームにおいて伝送するステップと、
- d) XML文章を所属のXMLスキーマを用いて符号化するステップと、
- e) 符号化されたXML文章を第2のビットストリームにおいて伝送するステップとを実施している形式の方法において、

復号化器が設けられており、該復号化器が以下のステップ(f~g)、すなわち、

- f) 第1のビットストリームにて伝送される符号化されたXMLスキーマをメタスキーマを用いて、正規化されたXMLスキーマへ復号化するステップと、;この場合前記正規化されたスキーマとメタスキーマは符号化の際に用いられたスキーマに相応しており、
- g) 第2のビットストリームにて伝送される符号化されたXML文章を、正規化されたXMLスキーマを用いて、正規化されたXMLスキーマのさらなる正規化を実施する事無く復号するステップとを実施することを特徴とする方法。

【請求項7】

構造化された文章の要素宣言及び/又は属性宣言が次のように再構築される、すなわちその伝送のためにネーム及び/又はコードが割当てられた匿名タイプ(AT0)がそれぞれの要素宣言若しくは属性宣言内に挿入され、それによってそのつどの特性タイプが参照されるように再構築される、請求項6記載の方法。

【請求項8】

ビットストリームから、タイプネーム及び/又は要素ネーム及び/又は代入グループネームが、数値と1つまたは複数のテーブルを介して復号化され、前記テーブルは、数値とタイプネーム及び/又は要素ネーム及び/又は代入グループネームの間の割当てを含んでいる、請求項6または7記載の方法。

【請求項9】

ビットストリームから、タイプネーム及び/又は要素ネーム及び/又は代入グループネームが、1つまたは複数のリストと、リスト内のタイプネーム及び/又は要素ネーム及び/又は代入グループネームの位置を介して復号化され、前記リストは、タイプネーム及び/又は要素ネーム及び/又は代入グループネームを含んでいる、請求項6から8いずれか1項記載の方法。

【請求項10】

最初にビットストリームから、タイプ及び/又は大域要素及び/又は代入グループの継承ツリーに対する情報が復号化され、その際各タイプは、マスタタイプに関するタイプコードと、記述タイプに係わる全てのタイプコードの長さに関する情報によって記述され、及び/又は、

各大域要素が、SBCの長さ及びSBC及び/又は代入グループの各要素と、代入コードの長さ及び代入コードによって記述される、請求項6から9いずれか1項記載の方法。

【請求項11】

請求項1から5いずれか1項記載の符号化方法を実施するように構成されている、符号化ユニットが設けられていることを特徴とする、XML文章の符号化のための装置。

【請求項12】

請求項6から10いずれか1項記載の復号化方法を実施するように構成されている、復号化ユニットが設けられていることを特徴とする、XML文章の復号化のための装置。

【請求項13】

前記符号化ユニットは、コンフィグレーション可能なバイトコードインタプリタを有しており、該インタプリタは、バイトコード内の情報を解釈し、コンフィグレーションに依存して、構造化された文章からバイトコードに基づいてパス又はペイロードを表わすコードを生成する、請求項11記載の装置。

【請求項14】

10

20

30

40

50

前記復号化ユニットは、コンフィグレーション可能なバイトコードインタプリタを有しており、該インタプリタは、ビットストリームからの情報によってコンフィグレーション可能であり、さらにコンフィグレーションに依存して、ビットストリームからバイトコードに基づいてパス、ペイロード又はバイトコードを生成する、請求項 1 2 記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、構造化された文章からスキーマに依存してビットストリームが生成される、構造化された文章、特にXML文章の符号化のための方法および装置、並びにビットストリームからスキーマに依存して構造化された文章が生成される、構造化された文章、特にXML文章の復号化のための方法および装置に関する。

10

【背景技術】

【0002】

MPEG-7 標準規格における策定の枠内では、XMLデータのバイナリーコーディングのための方法が開発されてきた。これは以下ではバイナリーフォーマット方式(Bim)とも称され、例えばISO/IEC標準文書の“ISO/IEC FDIS 15938-1:2001(E)、Information Technology - Multimedia Content Description Interface - Part 1: Systems”から公知である。この方法は、エンコードとデコードのもとに存在するXMLスキーマ定義、例えばMPEG-7スキーマを利用している。これはXML記述の個々のデータ要素に対するコードを生成するためである。この方法は、エンコードとデコードにおいて少なくとも部分的に同じスキーマ定義が存在することを前提とする。このことは例えば正規化されたXMLスキーマがデコード内に固定的に組込まれることによって保証され得る。その他にもスキーマを本来のドキュメントとは別個に若しくは付加的にデコードに転送することも可能である。エンコードからデコードへのスキーマの伝送は、テキストフォーマットで実施可能である。その場合には、標準のテキスト圧縮、例えばZIPなどが適用されてもよい。

20

【0003】

本発明の課題は、スキーマの伝送を特に効果的に行い、伝送されるデータ量と、スキーマからのコードテーブルの生成に対して要するデコードにおける計算機コストを低減させ得る、方法ないし装置の提供にある。その他に、不完全に伝送されたスキーマの一貫性を保証することである。

30

【0004】

前記課題は請求項 1 の特徴部分に記載の本発明によるエンコーディング方法、請求項 7 の特徴部分に記載の本発明によるデコーディング方法、請求項 1 4 の特徴部分に記載のエンコーディング装置、請求項 1 5 の特徴部分に記載の本発明によるデコーディング装置によって解決される。

【0005】

本発明による方法ないし装置のさらに別の有利な実施例は従属請求項に記載される。

【0006】

本発明は実質的にスキーマからメタスキーマに依存してビット流若しくはビット流の一部を生成するエンコーディング方法からなっており、この場合 1 つまたは複数の以下の最適化が実施される。:

40

- 要素宣言と属性宣言からの匿名タイプの分離とオウンタイプ(このタイプ定義はスキーマ定義内でトップレベル要素として例示されている)としてのコーディング、
- エンコード側のシンタックスツリーの正規化
- タイプネームのキャラクタストリングスの置換え
- インヘラタンスツリーのための上方の伝送。

【0007】

デコーディングはこれらの最適化を考慮し、また逆にビットストリームからスキーマが生成される。

50

【 0 0 0 8 】

本発明は以下の明細書で図面に示されている実施例に基づいて詳細に説明する。この場合、

図 1 は、本発明による符号化/復号化の説明のための基本原理図であり、

図 2 は、本発明の有利な実施形態の詳細を説明するための図であり、

図 3 は、本発明のさらに有利な実施形態の詳細を説明するための図であり、

図 4 は、本発明によるデコーダの有利な実施形態の基本原理図である。

【 実施例 】

【 0 0 0 9 】

次に本発明を図面に基づき以下の明細書で詳細に説明する。それらのパートに対する XML スキーマは、正規化された構文定義に基づく（特にいわゆる “Schema for Schemas”（W3C 仕様）これはいわばメタスキーマを表わしている）XML 文章であるので、スキーマも前述した BiM 方式によって符号化され伝送され得る。

【 0 0 1 0 】

図 1 には装置が示されている。この装置では、第 1 のステップにおいて、BiM 符号化方法 BiM - E を用いて XML スキーマ XML S からメタスキーマ SS に依存してビットストリームの一部若しくはビットストリーム BS 1 が生成され、第 2 のステップにおいて、同じ BiM 符号化方法 BiM - E を用いて XML 文章 XML からスキーマ XML S に依存してビットストリームのさらなる部分若しくはビットストリーム BS 2 が生成され、並びに逆の方向において BiM 復号化方法 BiM - D を用いてビットストリームの 2 つの部分若しくはビットストリーム BS 1 と BS 2 から XML スキーマ及び XML 文章が再生される。

【 0 0 1 1 】

本発明の第 1 の有利な実施例では、要素宣言ないし属性宣言からのいわゆる匿名タイプ “anonymous Type” の分離が行われる。

【 0 0 1 2 】

XML 文章の伝送は、BiM 方法の場合には、“depth first” で実施される。しかしながらデコーダにおけるスキーマ編集の過程は “breadth first” を要求する。これらの用語の表現は、例えばインターネットホームページ：http://www.generation5.org/simple_search.shtml に詳細に説明されている。例えばシーケンスやチョイスなどのグループでは、これらがデコーダ側の小さなバッファメモリによって補償できる。しかしながら匿名タイプ “anonymous Types”（この場合個々の要素や属性のタイプが定義できる）の場合では、符号化器側での再構築が複雑さに見合うものとされる。匿名タイプ定義 “anonymous Type Definition”（以下の明細書では単に AT0 とも称する）は、要素 “CurriculumVitae” の要素宣言から取り出され、名前及び/又はコードが得られる。それらは相応の要素のもとでの参照に用いられる。これによって有利には、伝送されるタイプの階層の深度が低減され、それによりデコーダ側でのスキーマのコンパイルが容易となる。

例：

再構築前のスキーマ

```
<complexType name="personDescriptor">
  <element name="CurriculumVitae">
    <complexType>
      <element name="name" type="string"/>
      <element name="birthday" type="data"/>
      ...
    </complexType>
  </element>
  <element name="profession" type="profTp"/>
```

< /complexType >

再構築後のスキーマ

<complexType name="personDescriptor" >

 <element name="CurriculumVitae" type="AT0" />

 <element name="profession" type="profTp" />

< /complexType >

<complexType name="AT0" >

 <element name="name" type="string" />

 <element name="birthday" type="data" />

...

< /complexType >

本発明の第2の有利な実施例によれば、シンタクスツリの正規化がBiMにおいて特定されているように符号器側で実施される。

【0013】

BiM方法においては、ビットストリームの復号化に用いられるいわゆる“Finite State Automations”が、XMLスキーマの構造をマッピングするシンタクスツリーから生成される。符号化の効果を高めるために、このシンタクスツリーは、テキストチャーXML定義に対して1:1で対応するのではなく、正規化が実行される。この場合は以下の3つの異なるケースが生じ得る。:

1. 1つの要素しか含まないグループの単純化:

このグループは分解され、含まれている要素は分解されたグループのレベルでコンテンツモデルにソートされる。この場合要素の属性“minOccurs”と“maxOccurs”は、再区分化の前に分解されたグループと要素の相応の属性の積によって置換えられる。

2. 属性値minOccurs = 0を有する要素を含んだ選択グループの単純化:

選択グループの属性値“minOccurs”は、先行する値に依存することなくセットされる。属性値minOccurs = 0を有していた要素には、minOccurs = 1が割当てられる

3. 選択された選択グループの単純化:

ある選択グループが、属性値minOccurs = 1を含んでいる別の選択グループを含むならば、当該選択グループは分解され、内容は上位に位置する選択グループに直接組込まれる。

【0014】

これらの単純化は、スキーマの伝送の際に既にエンコーダにおいて実行されるべきである。なぜならシンタクスツリー変換は、標準コードの割当てに影響を与えるからであり、さらにコンテンツモデルが直接引き継がれる場合には、スキーマのコンパイルがデコーダ側で単純化されるからである。

【0015】

これによってデコーダも負担が軽減され、コンテンツモデルは、タイプ復号化で生じるように直接供給可能となる。

【0016】

本発明の第3の有利な実施形態は、図2に示されているように、タイプネーム“TypeName”の文字列の置換えが実行される。

【0017】

タイプ定義の属性値“name”ないし“base”と、要素宣言または属性宣言の属性値“type”においては、スキーマにおいて頻繁に同じタイプネームが生じる。これらは文字列として何度も伝送されるものになってしまう。タイプネームの符号化の際には有利にはネームの代わりに数値(ナンバー)だけが符号化され、それとは別個にそれらの数値をもとのネームに再び関係づけるテーブルが用いられる。これに適する数値としては、全てのcomplexTypeのマスタタイプの継承ツリー“inheritance

10

20

30

40

50

t r e e ”に割当てられる、タイプネームが提案される。

【 0 0 1 8 】

相応にこれらのことは大域要素宣言の属性値 “ n a m e ” とそれらの “ r e f ” 属性における参照、並びに属性値 “ s u b s t i t u t i o n G r o u p ” における代入グループのネームに対しても当て嵌まる。このケースでは例えば大域要素のスキーマ分岐コード S B C が用いられてもよい。

【 0 0 1 9 】

これに伴ってデータ容量が節約可能となる。なぜなら同じタイプネームに対して繰返される参照がよりコンパクトな形態で表せるからである。またタイプ割当てテーブルも標準圧縮ツールによってより良好に圧縮できる。なぜならタイプネームはビットストリームを介して分散して現れるのではなく、コンパクトな形態でビットストリーム内の関連領域に現れるからである。

【 0 0 2 0 】

別の有利な実施形態によれば、タイプネーム若しくは要素ネームあるいは代入グループネームを含んだリストが符号化される。ネームに対して明示的に割当てられる数値の代わりに当該実施形態では、リスト内のネームの位置が数値として利用される。このことは次のような利点となる。すなわち複数の数値がもはやリスト内で符号化される必要がなくなり、それによって効率的な伝送が保証される。

【 0 0 2 1 】

本発明の第 4 の実施例によれば、継承ツリーに対する情報の伝送が行われる。

【 0 0 2 2 】

各タイプ定義は、いわゆる属性値 “ b a s e ” を含んでおり、その情報があればどのタイプかを継承づけられる。スキーマに対してこれらの全ての情報が収集されれば、いわゆる継承ツリーとしてのツリー構造が得られる。この継承ツリーは、B i M 符号化方法の場合に、タイプ変換のケースにおいて (t y p e - c a s t) 要素の新たなタイプを伝送するために用いられる。その際にはコードが、ベースタイプによって継承される全てのタイプに対応付けられ (いわゆるタイプコード)、並びに当該コードの長さも正確な復号化のために重要となる。この長さは、ベースタイプのもとで継承ツリー内の全てのタイプの総数から生じる。スキーマが完全に伝送された場合には、コードもコード長もデコーダ側で一義的に求めることができる。しかしながらスキーマがデコーダ側で不完全である時には、既に伝送されたタイプにタイプコードを割当てるためにはさらなる付加情報が伝送されなければならない。

【 0 0 2 3 】

伝送された各コードは、ネーム領域において、マスタコードに関するタイプコードナンバを有して。それにより、派生されるタイプのタイプコードは、簡単な差分形成によって求められる。さらに伝送されたタイプによって定められるサブツリーの能力に関する情報と、当該伝送されたタイプから導出されるタイプのタイプコード長に関する情報も欠落し得る。この長さは、可変の長さコードの中で少ないビットで伝送され得る。

【 0 0 2 4 】

図 3 には、さらなるタイプの派生しているタイプ A を伴ったスキーマの継承ツリーの例が表わされている。このタイプは、マスタタイプ “ a n y T y p e ” に関して例えばタイプコード 1 3 4 を得ている。タイプ A からはタイプ A A , A B , A C が派生しており、それらのタイプコードはマスタタイプに関して与えられる。ベースタイプ A に関するタイプコードを求めるためには、所望のタイプのタイプコードから、ベースタイプのタイプコードと 1 を減算するだけでじゅうぶんである。:

$$T C_{T y p e} = T C_{T y p e \text{ b z g l . } U r t y p} - T C_{B a s i s t y p \text{ b z g l . } U r t y p} - 1$$

タイプコードの長さに関する欠落した情報は、有利には参照テーブルにおいて付加的な数として統合可能である。

【 0 0 2 5 】

タイプ割当てテーブルの中の情報を標準圧縮ツールを用いて圧縮できるようにするためには、全バイトをそろえて記憶させることが推奨される (byte-aligned)。最初の数値 “v l u i m s b f 5” は、テーブル内のラインナンバを符号化する数であり、それに続く “v l u i m s b f 5” は、タイプコードに対するビットの数を符号化する数であり、さらに続く “v l u i m s b f 5” は、マスタタイプ自体に係るタイプコードを表わす数である。さらに充填ビット若しくはスタッフィングビットがバイトの境界合わせを達成するために続く。：

【 0 0 2 6 】

【表 1】

タイプ割当てテーブルのフォーマット			
Vuimsbf5	Vuimsbf5	ビット	文字列
ラインナンバー			
タイプコード の長さ 1	タイプ コード 1	0～7 充填ビット	ネーム タイプ 1
タイプコード の長さ 2	タイプ コード 2	0～7 充填ビット	ネーム タイプ 2
...	...	...	...

【 0 0 2 7 】

タイプ割当てテーブルの伝送は、基礎をなすスキーマがないか若しくはまだ完全に伝送されていない、及び/又は復号化されていなかった場合ですら、符号化された文章内によって存在するタイプコードを正確に復号化させることを可能にする。

【 0 0 2 8 】

相応にグローバル S B C は大域要素と共に伝送されなければならない、代入コードは代入グループに属する要素のケースにおいて伝送されなければならない。その場合全ての大域要素に先立って 1 つのグローバル S B C 長さと、代入グループのヘッダ要素と共に各代入コードの長さが伝送されなければならない。

【 0 0 2 9 】

符号化の際には、個々の実施例で表わされた特徴のあらゆる組み合わせが可能であり、さらに復号化の際にも相応の形式で用いられる。

【 0 0 3 0 】

B i M 方法は、XML スキーマを 1 つのフォーマットにコンパイルする必要がある。このフォーマットは、コードワードの長さの決定とコードの値によるデータ要素の選択を許容する。これについては複数の可能性が存在する。M P E G - 7 標準 (I S O / I E C 1 5 9 3 8 - 1 : 2 0 0 1 P a r t 1 : S y s t e m s ないし I S O / I E C 1 5 9 3 8 - 6 : 2 0 0 1 P a r t 6 : 参照ソフトウェア) では、有効情報ないしペイロードの復号化に対し有限状態オートマトンと、コンテキストパスの復号化のためのスキーマから生成されるコードテーブルを使用するモデルが提案されている。

【 0 0 3 1 】

図 4 に示されている有利な本発明によるデコーダでは、復号化過程がバイトコードモデルで記述されており、この場合スキーマ構造が、バイトコードインタープリタ B C I によって処理されるインターリンクされた状態のシステムへトランスレートされる。ここでは

エンコーダによって受信されたビットストリームBSが、選択すべきサブシーケンス状態に関する情報を含んでいる。MPEG-7標準で提案されるモデルとの違いは、バイトコードモデルが、ペイロードを表わすビットストリームもコンテキストパスを表わすビットストリームも復号化できることである。そのためスキーマの中に含まれている同じ情報をデコーダにおいて異なるコード化方法毎に2度維持する必要がなくなる。インタープリタBCIは、入力ビットストリームからの情報を読み出し、それをバイナリーフォーマットBIMでXML文章またはXMLスキーマに符号化する。この情報は、バイトコード内に記憶されている目下の状態のサブシーケンス状態に基づく選択を可能にする。それらのサブシーケンス状態は、バイトコード内でポインタPとして永久的に作成される。コンフィグレーションに応じてパス、ペイロード又はバイトコードが出力される。

10

【0032】

スキーマの復号化は、前述した変更によっても効果的にバイトコードモデル内で実現され得る。この場合には、ペイロードもパスも出力されず、代わりにバイトコードが直接生成される。このバイトコードは対応するタイプの復号化のためのバイトコードインタプリタによって利用可能である。

【0033】

バイトコードは、構造要素ないしは状態からなっている。これらの状態とは、当該状態のヘッダビット領域を用いて識別される種々異なるタイプのものである。これらの状態はタイプに依存して様々な情報領域を含んでおり、それらの領域はバイトコードインタプリタによって読み出され、コンフィグレーション(ペイロード/コンテキストパス)や目下の状態に応じて評価される。

20

【0034】

スキーマ情報を表している状態の種別に対しては多数の変化例が考えられる。重要なことは、XMLスキーマの全ての構文要素がバイトコードモデルの状態によって再構築できることと、MPEG-7標準で定められる2つのアルゴリズム(コンテキストパス/ペイロード)の効果的な復号化に必要な全ての情報が当該状態の中から得られることである。

【0035】

以下では、バイトコードの可能な構造を簡単に述べる。：

状態のタイプ：概要

1. Complex typeのヘッダ状態

30

タイプのヘッダ状態は、Complex typeの復号化の際のスターティングポイントを再生する。それにはネームとタイプが含まれ(anonymous typeではない)並びにタイプの継承(inheritance)に関する情報(pointer to base state)及び多態性に関する情報が含まれる。ペイロード符号化に対する特定因子は、タイプ属性のリストに対するポインタからなる。コンテキストパス符号化に対する特定因子は、コンテキスト及びオペランドツリーブランチコードテーブルに対する子の要素の数を有するフィールドからなる。最後の情報フィールドは、サブシーケンス状態に対するポインタである。すなわち、Complex typeの内容を表している第1の状態である(例えば要素状態若しくは選択状態)。

【0036】

40

【表 2】

ヘッダ情報のグラフィカル表示：

ヘッダビット領域
ネーム付きストリングに対するポインタ
ベースタイプのヘッダ状態に対するポインタ
継承ツリーに対するポインタ
コンテキストTBCの子の数
オペランドTBCの子の数
サブシーケンス状態に対するポインタ

10

2．選択状態

選択状態は、XMLスキーマの選択グループを再生する。この選択状態は、実質的に可能なサブシーケンス状態を伴うポインタリストを含んでいる。実際に選択された状態を決定するために、ペイロードの復号化の際にはビットストリームが読出されなければならない。選択状態からは2つの変化例が存在する。すなわち、開始状態（これは様々な生じ得るサブシーケンス状態に分岐する）と終了状態（これは選択を再び集約する）である。

20

3．要素状態

要素状態はスキーマのcomplexTypeにおける要素宣言を再生する。それには要素のネームを伴った文字列に対するポインタ並びにタイプのヘッダ状態に対するポインタが含まれる。さらに場合によっては、ポジションコードの長さに関する情報（パスの復号化に対してのみ）や代入グループに対する情報が存在し得る。

4．属性状態

属性状態は、スキーマの属性宣言を再生する。これには俗世位置のネームに対するポインタ並びに属性値のsimpleTypeのヘッダ状態に対するポインタが含まれる。

5．オカレンス状態

30

オカレンス状態は、XMLスキーマ、例えば要素若しくはグループ（選択、シーケンスなど）のもとで生じ得るminOccurs属性とmaxOccurs属性を再生する。それには、さらなる要素又はグループのインスタンスの場合のサブシーケンス状態に対するポインタ、並びにグループの最後のインスタンスが符号化された場合のサブシーケンス状態に対するポインタが含まれる。XMLスキーマの場合要素自体が含まれる可能性が生じ得るので（要素自体がcomplexType定義内、若しくはより深いネ스팅内において再び生じる）、オカレンス状態も同時に2回以上アクティブとなる。それ故にオカレンス状態内でステープルに対するポインタが必要となる。これはオカレンス状態の各アクティブインスタンスの現下の状態を保証する。

6．タイプの終了状態

40

タイプの終了状態は当該タイプの全ての属性値を伴うポインタリストを含む。それはパスの復号化の際に必要となる。なぜならツリー分岐コードテーブル内では全ての属性値がテーブルの最後に取り入れられるからである。終了状態に達した場合には、バイトコードインタープリタが当該タイプを読出した要素内へ階層的に分岐される。呼び出される要素に関する相応の情報は、バイトコードインタープリタの作業メモリ内に記憶されなければならない。

7．simpleTypeのヘッダ状態

この状態は内容の復号化を制御している。すなわちそれには、該当するタイプの特定のデータをビットストリームから読出し復号化できるコーデックに対するポインタが含まれている。このコーデックのタイプは、情報フィールド内に特定化されている。

50

【 0 0 3 7 】

従来技法の M P E G 7 参照ソフトウェアと比べた場合のバイトコードモデルの利点は以下の通りである。：

1．スキーマ情報は、2つのコーディング方法（コンテキストパス/ペイロード）毎に一度だけデコーダで表される。バイトコード状態における情報の大半部分は、2つの方法に対して関連している。残りの僅かな部分は、2つの方法のうちのそれぞれ1つに対してのみ特定化される。それ故にデコーダにおけるスキーマ情報の表記は非常にコンパクトになる。

2．バイトコードモデルにより、スキーマ情報に対して例えば事前コンパイルや記憶にも適した（テキストとしてXMLスキーマの代わりに）よく定義されたデータフォーマット

10

3．標準プロセッサによるバイトコードの実施は非常に高速に行われ得る。なぜならバイトコードモデルが復号化過程を非常に効果的に準備するからである。全ての情報は、ポインタを介して状態の中で直接得ることができ、（ISO/IEC 15938-6，パート6の参照ソフトウェアのように）一部を最初にリスト内で検索する必要はない。

【 0 0 3 8 】

相応するエンコーダは、同じ形式で実現できる。この場合には、次のように逆になる。すなわち構造化された文章のテキスト表記によって状態が制御され状態遷移がバイナリ表記を生成する。

【図面の簡単な説明】

20

【 0 0 3 9 】

【図1】本発明による符号化/復号化の説明のための基本原理図

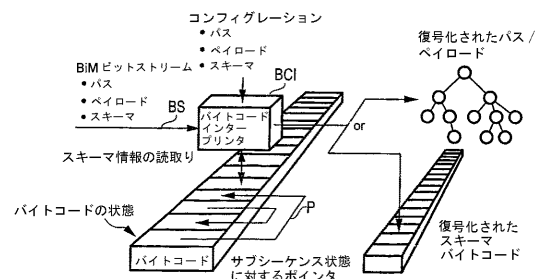
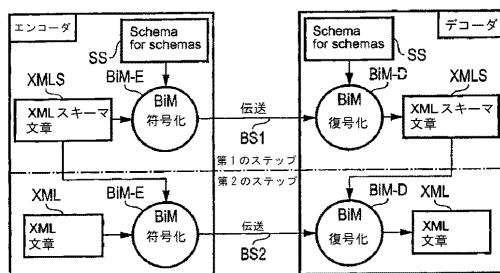
【図2】本発明の有利な実施形態の詳細を説明するための図

【図3】本発明のさらに有利な実施形態の詳細を説明するための図

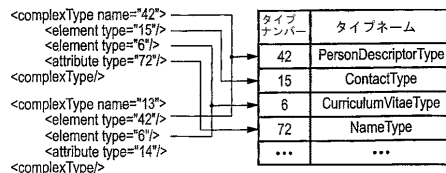
【図4】本発明によるデコーダの有利な実施形態の基本原理図

【図1】

【図4】

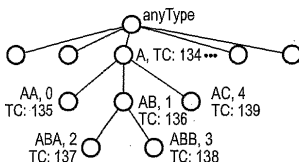


【図2】



【図3】

FIG 3



フロントページの続き

(74)代理人 100099483

弁理士 久野 琢也

(74)代理人 100114890

弁理士 アインゼル・フェリックス＝ラインハルト

(74)代理人 230100044

弁護士 ラインハルト・アインゼル

(72)発明者 イェルク ホイアー

ドイツ連邦共和国 ミュンヘン フィッシュバッハアウアーシュトラッセ 8

(72)発明者 アンドレアス フッター

ドイツ連邦共和国 ミュンヘン シュトウルミスヴェーク 42

(72)発明者 ウルリッヒ ニーダーマイアー

ドイツ連邦共和国 ランダウ フィーハウザーシュトラッセ 18

審査官 須藤 竜也

(56)参考文献 仏国特許出願公開第2813743(FR, A1)

国際公開第02/33977(WO, A1)

(58)調査した分野(Int.Cl., DB名)

H03M 7/30

G06F 17/21