

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2006-514354

(P2006-514354A)

(43) 公表日 平成18年4月27日(2006.4.27)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 12/00 (2006.01)	G06F 12/00 511C	5B009
G06F 17/21 (2006.01)	G06F 12/00 547Z	5B082
	G06F 17/21 501T	

審査請求 未請求 予備審査請求 未請求 (全 148 頁)

(21) 出願番号 特願2004-557436 (P2004-557436) (86) (22) 出願日 平成15年11月29日 (2003.11.29) (85) 翻訳文提出日 平成17年7月26日 (2005.7.26) (86) 国際出願番号 PCT/US2003/038137 (87) 国際公開番号 W02004/051423 (87) 国際公開日 平成16年6月17日 (2004.6.17) (31) 優先権主張番号 10/309,537 (32) 優先日 平成14年12月4日 (2002.12.4) (33) 優先権主張国 米国 (US)	(71) 出願人 390009531 インターナショナル・ビジネス・マシー ズ・コーポレーション INTERNATIONAL BUSIN ESS MACHINES CORPO RATION アメリカ合衆国10504 ニューヨーク 州 アーモンク ニュー オーチャード ロード (74) 代理人 100086243 弁理士 坂口 博 (74) 代理人 100091568 弁理士 市位 嘉宏 (74) 代理人 100108501 弁理士 上野 剛史
---	---

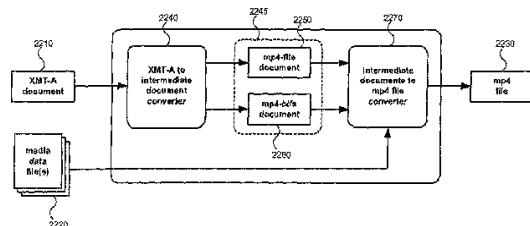
最終頁に続く

(54) 【発明の名称】 MPEG-4 Textual 表現から MPEG-4 Intermediate Format を作成する効率的な手段

(57) 【要約】

【課題】 XMT (Extensible MPEG-4 Textual) 文書をバイナリ MPEG-4 (mp4) ファイルに変換する方法、システム、およびコンピュータ・プログラム製品を提供すること。

【解決手段】 XMT 文書に、0 個以上の関連するメディア・データ・ファイルが含まれる可能性がある。本発明に、mp4 ファイルを表す中間文書を生成することと、中間文書および関連するメディア・データ・ファイルに基づいて mp4 ファイルを作成することが含まれる。第 1 コンバータが、XMT 文書を入力し、mp4 ファイルの構造を表す少なくとも 1 つの中間文書を生成するように構成される。第 2 コンバータが、中間文書および関連するメディア・データ・ファイルを入力し、mp4 ファイルを生成するように構成される。



【特許請求の範囲】

【請求項 1】

Extensible MPEG-4 Textual (XMT) 文書をバイナリ MPEG-4 (mp4) ファイルに変換する方法であって、前記 XMT 文書は、0 個以上の関連するメディア・データ・ファイルを有し、

前記 mp4 ファイルを表す中間文書を生成することと、

前記中間文書および前記関連するメディア・データ・ファイルに基づいて前記 mp4 ファイルを作成することとを含む方法。

【請求項 2】

10

前記中間文書の作成が、さらに、前記 mp4 ファイルの特定の部分を表す 1 つまたは複数の追加中間文書を生成することを含む、請求項 1 に記載の方法。

【請求項 3】

前記追加中間文書の生成が、さらに、scene description stream を表す mp4-bifs 文書を生成することを含む、請求項 2 に記載の方法。

【請求項 4】

前記 mp4-bifs 文書が、前記 mp4 ファイルを表す前記中間文書と別々の別個の文書である、請求項 3 に記載の方法。

【請求項 5】

前記追加中間文書の生成が、さらに、object descriptor stream 20 を表す文書を生成することを含む、請求項 2 に記載の方法。

【請求項 6】

前記 object descriptor stream を表す前記文書が、前記 mp4 ファイルを表す前記中間文書と別々の別個の文書である、請求項 5 に記載の方法。

【請求項 7】

前記 mp4 ファイルの作成が、各関連するメディア・データ・ファイルの各時間的要素を表すバイト数を判定することを含む、請求項 1 に記載の方法。

【請求項 8】

前記 mp4 ファイルの作成が、各関連するメディア・データ・ファイルの各時間的要素の時間的持続時間を判定することを含む、請求項 1 に記載の方法。

30

【請求項 9】

Extensible MPEG-4 Textual (XMT) 文書をバイナリ MPEG-4 (mp4) ファイルに変換するシステムであって、前記 XMT 文書は、0 個以上の関連するメディア・ファイルを有し、

前記 XMT 文書を入力し、前記 mp4 ファイルの構造を表す少なくとも 1 つの中間文書を生成するように構成された第 1 コンバータと、

前記中間文書およびすべての関連するメディア・ファイルを入力するように構成され、さらに、前記 mp4 ファイルを生成するように構成された第 2 コンバータとを含むシステム。

【請求項 10】

40

前記中間文書が、前記 mp4 ファイルの特定の部分を表す追加中間文書を含む、請求項 9 に記載のシステム。

【請求項 11】

前記追加中間文書が、scene description stream を表す mp4-bifs 文書を含む、請求項 10 に記載のシステム。

【請求項 12】

前記追加中間文書が、object descriptor stream を表す文書を含む、請求項 10 に記載のシステム。

【請求項 13】

object descriptor stream を表す前記文書が、前記 mp4 フ 50

ファイルを表す前記中間文書に含まれる、請求項12に記載のシステム。

【請求項14】

object descriptor streamを表す前記文書が、前記mp4ファイルを表す前記中間文書と別々の別個の文書である、請求項12に記載のシステム。

【請求項15】

前記mp4ファイルを表す前記中間文書が、object descriptor streamを表す文書を参照する、請求項14に記載のシステム。

【請求項16】

有形の媒体で実施されたコンピュータ・プログラムであって、

Extensible MPEG-4 Textual (XMT) 文書をバイナリMP E G-4 (mp4) ファイルに変換する、前記有形の媒体に結合されたコンピュータ・可読プログラム・コード

を含み、前記XMT文書が、0個以上の関連するメディア・データ・ファイルを有し、前記コンピュータ可読プログラム・コードが、前記プログラムに、

前記mp4ファイルを表す中間文書を生成させ、

前記中間文書および前記関連するメディア・データ・ファイルに基づいて前記mp4ファイルを作成させる

ように構成される、コンピュータ・プログラム。

【請求項17】

前記中間文書が、Extensible Markup Language (XML) 文書である、請求項16に記載のコンピュータ・プログラム。

【請求項18】

前記中間文書を生成させるように構成された前記コンピュータ可読プログラム・コードが、さらに、前記mp4ファイルの特定の部分を表す1つまたは複数の追加中間文書を生成するように構成されたコンピュータ可読プログラム・コードを含む、請求項16に記載のコンピュータ・プログラム。

【請求項19】

前記追加中間文書が、Extensible Markup Language (XML) 文書である、請求項18に記載のコンピュータ・プログラム。

【請求項20】

1つまたは複数の追加中間文書を生成するように構成された前記コンピュータ可読プログラム・コードが、さらに、scene description streamを表すmp4-bifs文書を生成するように構成された可読プログラム・コードを含む、請求項18に記載のコンピュータ・プログラム。

【請求項21】

前記mp4-bifs文書が、前記mp4ファイルを表す前記中間文書に含まれる、請求項20に記載のコンピュータ・プログラム。

【請求項22】

前記mp4-bifs文書が、前記mp4ファイルを表す前記中間文書と別々の別個の文書である、請求項20に記載のコンピュータ・プログラム。

【請求項23】

前記mp4ファイルを表す前記中間文書が、前記mp4-bifs文書を参照する、請求項22に記載のコンピュータ・プログラム。

【請求項24】

1つまたは複数の追加中間文書を生成するように構成された前記コンピュータ可読プログラム・コードが、さらに、object descriptor streamを表す文書を生成するように構成されたコンピュータ可読プログラム・コードを含む、請求項18に記載のコンピュータ・プログラム。

【請求項25】

前記object descriptor streamを表す前記文書が、前記mp

10

20

30

40

50

4 ファイルを表す前記中間文書に含まれる、請求項 24 に記載のコンピュータ・プログラム。

【請求項 26】

前記 object descriptor stream を表す前記文書が、前記 mp4 ファイルを表す前記中間文書と別々の別個の文書である、請求項 24 に記載のコンピュータ・プログラム。

【請求項 27】

前記 mp4 ファイルを表す前記中間文書が、前記 object descriptor stream を表す前記文書を参照する、請求項 26 に記載のコンピュータ・プログラム。

10

【請求項 28】

前記 mp4 ファイルを作成させるように構成された前記コンピュータ可読プログラム・コードが、各関連するメディア・データ・ファイルの各時間的要素を表すバイト数を判定するように構成されたコンピュータ可読プログラム・コードを含む、請求項 16 に記載のコンピュータ・プログラム。

【請求項 29】

前記 mp4 ファイルを作成させるように構成された前記コンピュータ可読プログラム・コードが、各関連するメディア・データ・ファイルの各時間的要素の時間的持続時間を判定するように構成されたコンピュータ可読プログラム・コードを含む、請求項 16 に記載のコンピュータ・プログラム。

20

【請求項 30】

前記関連するメディア・データ・ファイルが、1 つまたは複数のオーディオ・データ・ファイルを含む、請求項 16 に記載のコンピュータ・プログラム。

【請求項 31】

前記関連するメディア・データ・ファイルが、1 つまたは複数のイメージ・データ・ファイルを含む、請求項 16 に記載のコンピュータ・プログラム。

【請求項 32】

前記関連するメディア・データ・ファイルが、1 つまたは複数のビデオ・データ・ファイルを含む、請求項 16 に記載のコンピュータ・プログラム。

【発明の詳細な説明】

30

【技術分野】

【0001】

本発明は、全般的にはマルチメディア情報のデータ表現に関し、具体的には、「MPEG-4 Textual Representation」と称するマルチメディア情報表現の 1 つ形の、「MPEG-4 Intermedia Format」と称するマルチメディア情報表現の別の形への変換に関する。

【背景技術】

【0002】

コンピュータは、一般に、イメージ、オーディオ・サンプル（サウンド）、およびビデオ・メディアを含むさまざまなデジタル・メディアならびにテキストおよび幾何形状を表すのに使用される。これらのメディア・タイプのそれぞれを、個別に提示することができ、あるいは、複数のそのようなメディア要素を、コンポジット・マルチメディア・プレゼンテーション（composite multimedia presentation）と称するもので一緒に提示することができる。

40

【0003】

コンポジット・マルチメディア・プレゼンテーションを作成し、配布する能力は、さまざまなメディア・タイプに基づく情報の散布に非常に重要である。さらに、コンポジット・マルチメディア・プレゼンテーションを提示する標準化された手段が作成されて、多数の作成者が、パーソナル・コンピュータ、セットトップ・ボックス、および他のデバイスなどのさまざまなコンピュータ・プラットフォームで再生できるプレゼンテーションを作

50

成できるようになった。

【0004】

MPEG (Motion Pictures Experts Group) によって開発されたコンポジット・マルチメディア・プレゼンテーションの2つの周知の標準化されたフォーマットが、XMT (Extensible MPEG-4 Textual) フォーマットおよびバイナリ・コーディングされたMPEG-4 (mp4) フォーマットである。XMTフォーマットは、コンポジット・マルチメディア・プレゼンテーションのオーサリングによく適し、mp4フォーマットは、コンポジット・マルチメディア・プレゼンテーションのコンパクトな保管および伝送によく適する。

【非特許文献1】 <http://www.w3.org/TR/2000/REC-XML-20001006>

10

【非特許文献2】 http://mpeg.telecomitalialab.com/working_documents.htm

【非特許文献3】 ISO/IEC文書14496-1:2000 Amd.2、2000年10月

【非特許文献4】 ISO-IEC文書ISO/IEC 14496-1:2001、2001年8月、第9章

【非特許文献5】 <http://mpeg.telecomitalialab.com/documents.htm>

【非特許文献6】 <http://developer.apple.com/techpubs/quicktime/qtdevdocs/REF/refFileFormat96.htm>

【非特許文献7】 <http://developer.apple.com/techpubs/quicktime/qtdevdocs/PDF/QTFFileFormat.pdf>

【非特許文献8】 ISO/IEC文書14496-2 (1999年、2000年修正) 「情報テクノロジーオーディオ・ビジュアル・オブジェクトのコーディングー第2部：ビジュアル (Information Technology-Coding of Audio-Visual Objects - Part 2: Visual)」

20

【非特許文献9】 ISO/IEC文書13818-7 (1997年) 「情報テクノロジー動画および関連するオーディオ情報の包括的コーディングー第7部：高度なオーディオ・コーディング (Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 7: Advanced Audio Coding)」

【発明の開示】

【発明が解決しようとする課題】

【0005】

したがって、XMTフォーマットされたプレゼンテーションをmp4フォーマットされたプレゼンテーションに効率的に変換することが望ましい。

30

【課題を解決するための手段】

【0006】

下で詳細に示すように、本発明は、XMT (Extensible MPEG-4 Textual) フォーマットをバイナリ・コーディングされたMPEG-4 (mp4) フォーマットに変換する方法、システム、および装置である。本発明は、比較的少量のソフトウェアからなる、XMTフォーマットからmp4フォーマットへのコンポジット・マルチメディア・プレゼンテーション変換を達成するのに適度なリソースだけを必要とする効率的な機能を使用する。

【0007】

40

したがって、本発明の態様に、XMT (Extensible MPEG-4 Textual) 文書をバイナリMPEG-4 (mp4) ファイルに変換する方法が含まれる。XMT文書は、0個以上の関連するメディア・データ・ファイルからなるものとすることができる。この方法には、mp4ファイルを表現する中間文書を生成することと、中間文書および関連するメディア・データ・ファイルに基づいてmp4ファイルを作成することが含まれる。

【0008】

本発明のもう1つの態様は、0個以上の関連するメディア・ファイルを有するXMT (Extensible MPEG-4 Textual) 文書をバイナリMPEG-4 (mp4) ファイルに変換するシステムである。このシステムには、XMT文書を入力し、

50

mp4ファイルの構造を表す少なくとも1つの中間文書を生成するように構成された第1コンバータが含まれる。第2コンバータが、中間文書およびすべての関連するメディア・ファイルを入力し、mp4ファイルを生成するように構成される。

【0009】

本発明のもう1つの態様は、0個以上の関連するメディア・ファイルを有するXMT (Extensible MPEG-4 Textual) 文書をバイナリMPEG-4 (mp4) ファイルに変換する、有形の媒体で実施されたコンピュータ・プログラム製品である。このコンピュータ・プログラムは、mp4ファイルを表す中間文書を生成する動作と、中間文書および関連するメディア・データ・ファイルに基づいてmp4ファイルを作成する動作を実行する。

10

【0010】

本発明の前述および他の特徴、有用性、および利益は、添付図面に示された本発明のさまざまな実施形態の次の詳細な説明から明白になる。

【発明を実施するための最良の形態】

【0011】

見出しの一覧

MPEG-4 Textual Representation 1. 0

MPEG-4 Intermedia Formatファイル 2. 0

Scene Description Stream (sds) 3. 0

Object Descriptor Stream (ods) 4. 0

20

mp4-file文書 5. 0

mp4-bifs文書 6. 0

xmta-mp4コンバータ 7. 0

XMT-A文書に基づく中間文書の作成 7. 1

XMT-A文書、mp4-file文書、およびmp4-bifs文書の作成 7. 1. 1

mp4-bifs文書の新しい「bifsConfig」要素の作成 7. 1. 2

mp4-file文書の新しい「moov」要素の作成 7. 1. 3

XMT-A「Header」要素の処理 7. 1. 4

XMT-A Body要素の処理 (パス1) 7. 1. 5

XMT-A par要素の処理 (パス1) 7. 1. 5. 1

30

XMT-Aコマンド要素の処理 (Pass1) 7. 1. 5. 2

「ODUpdate command-1の処理」手順 7. 1. 5. 3

「ODRemove cmndの処理」手順 7. 1. 5. 4

odsのEdit List作成 7. 1. 6

XMT-A Body要素の処理 (パス2) 7. 1. 7

XMT-A par要素の処理 (パス2) 7. 1. 7. 1

ODUpdate command-2の処理 7. 1. 7. 2

Insertコマンドの処理 7. 1. 7. 3

「InsertRouteコマンドの作成」手順 7. 1. 7. 4

「InsertNodeコマンドの作成」手順 7. 1. 7. 5

40

Deleteコマンドの処理 7. 1. 7. 6

Replaceコマンドの処理 7. 1. 7. 7

「ReplaceRouteコマンド作成」手順 7. 1. 7. 8

「ReplaceSceneコマンド作成」手順 7. 1. 7. 9

「XMTA BIFSノードの処理」手順 7. 1. 7. 10

データ・フォーマット変換 7. 1. 7. 11

mp4-bifs文書へのコマンド・フレームの挿入 7. 1. 8

odsのmdat要素へのODコマンドの挿入 7. 1. 9

mp4-bifs文書およびmp4-file文書のbifsConfigの更新 7

. 1. 10

50

ES_Descriptorの処理	7. 1. 10. 1	
track要素の作成	7. 1. 10. 2	
予備的サンプル・テーブル要素の作成	7. 1. 10. 3	
BIFS構成の処理	7. 1. 10. 4	
中間XML文書に基づくmp4バイナリ・ファイルの作成	7. 2	
入力文書および出力宛先の確立	7. 2. 1	
mp4アトム作成の処理	7. 2. 1. 1	
mp4オブジェクト構造の作成の処理	7. 2. 1. 2	
作業配列の作成	7. 2. 2	
「mdat」要素の処理	7. 2. 3	10
メディア・ファイル・データの挿入	7. 2. 3. 1	
メディア・データ・チャンクの挿入	7. 2. 3. 2	
odsmデータの挿入	7. 2. 3. 3	
ObjectDescrUpdate要素	7. 2. 3. 4	
ObjectDescrRemove要素	7. 2. 3. 5	
sdsmdデータの挿入	7. 2. 3. 6	
ノード挿入BIFSコマンド	7. 2. 3. 7	
Indexed Value Insertion BIFSコマンド	7. 2. 3. 8	
Route Insertion BIFSコマンド	7. 2. 3. 9	20
Node Deletionコマンド	7. 2. 3. 10	
Indexed Value Deletion BIFSコマンド	7. 2. 3. 11	
Route Deletion BIFSコマンド	7. 2. 3. 12	
Node Replacement BIFSコマンド	7. 2. 3. 13	
Field Replacement BIFSコマンド	7. 2. 3. 14	
Indexed Value Replacement BIFSコマンド	7. 2. 3. 15	
Route Replacement BIFSコマンド	7. 2. 3. 16	
Scene Replacement BIFSコマンド	7. 2. 3. 17	30
Route構造	7. 2. 3. 18	
SFNode構造	7. 2. 3. 19	
SFField構造	7. 2. 3. 20	
「moov」要素の処理	7. 2. 4	
mp4fiods要素の処理	7. 2. 4. 1	
各track要素の処理	7. 2. 4. 2	
mdia要素の処理	7. 2. 4. 3	
hdlr要素の処理	7. 2. 4. 4	
minf要素の処理	7. 2. 4. 5	
stbl要素の処理	7. 2. 4. 6	40
stsc要素の処理	7. 2. 4. 7	
stts要素の処理	7. 2. 4. 8	
stco要素の処理	7. 2. 4. 9	
stsz要素の処理	7. 2. 4. 10	
Indexed Value Replacement BIFSコマンド	7. 2. 4. 11	
Route Deletion BIFSコマンド	7. 2. 4. 12	
ES_Descr要素の処理	7. 2. 4. 13	
Field Replacement BIFSコマンド	7. 2. 4. 14	
Indexed Value Replacement BIFSコマンド	7. 2. 4. 15	50

4. 15

Visual Config要素の処理 7. 2. 4. 16

Audio Config要素の処理 7. 2. 4. 17

メディア・ヘッダ要素の処理 7. 2. 4. 18

treef要素の処理 7. 2. 4. 19

edts要素の処理 7. 2. 4. 20

任意選択のユーザ・データ要素の処理 7. 2. 5

odsmバッファ・サイズの更新 7. 2. 6

【0012】

本発明は、XMT (Extensible MPEG-4 Textual) フォーマット (本明細書では、XMT-A文書およびMPEG-4 Textual Representationとも称する) をバイナリ・コーディングされたMPEG-4 (mp4) フォーマット (MPEG-4 intermedia binaryフォーマットとも称する) に変換する方法、システム、およびコンピュータ・プログラムである。本発明は、比較的少量のソフトウェアおよび適度なリソースだけを必要とする、XMT-Aからmp4への変換を達成する新規の手法を使用する。本発明を、本明細書で、図1から107を参照して説明する。

【0013】

1. 0 MPEG-4 Textual Representation

MPEG-4 Textual Representationは、マルチメディア・プレゼンテーションの構造を表す「テキスト・ファイル」からなる。マルチメディア・プレゼンテーションは、サウンド、静止画像、ビデオ・クリップ、および他の要素の同期化された組合せまたはシーケンスからなる。テキスト・ファイルは、文字、数字、および句読点のバイナリ・コードのシーケンスからなる電子データ構造である。一般に、テキスト・ファイルは、一般に「テキスト・エディタ」と称するソフトウェアを使用して解釈することができる。Windows (R) オペレーティング・システムに基づくコンピュータ用の「Notepad.exe」と称するソフトウェアおよび集合的にUNIXと称するさまざまなオペレーティング・システムを使用するコンピュータ用の「vi」を含む、テキスト・エディタの多数の例がある。Windowsは、米国ワシントン州レッドモンドのMicrosoft Corporation社の登録商標である。MPEG-4 Textual Representationを含む特定のタイプのテキスト・ファイルを、
「XMT-A」ファイルと称する。

【0014】

テキスト・ファイルの範囲内で、XMT-Aファイルは、XML (Extensible Markup Language) ファイルの例である。XMLファイルは、World Wide Web Consortiumによって指定される原理に基づく構造化文書である (<http://www.w3.org/TR/2000/REC-XML-20001006>を参照されたい)。XMT-Aファイルは、国際標準化機構および国際電気標準会議によって指定される、XMLファイルの特定の実施形態を表す (http://mpeg.telecomitalia.com/working_documents.htmおよび、スイス国ジュネーブ20、CH-1211、1、rue de Varembé、Case postale 56の国際標準化機構 (ISO) から入手可能なISO/IEC文書14496-1:2000 Amd. 2、2000年10月を参照されたい)。XMT-A仕様のすべての部分の完全な記述は、膨大な量になる。したがって、XMT-Aファイルの次の説明は、その仕様のうちで、本発明を説明するのに必要な部分に限られる。XMT-Aファイル構造の完全な説明については、上で示したXMT仕様書を調査されたい。

【0015】

XMLファイルと同様に、XMT-Aファイルは、「要素」の階層的な組からなる。各要素に、子要素と称する従属要素を含めることができる。さらに、各要素は、「属性」と称するデータ値の組を所有することができる。各属性は、名前および値を有する。特定の

要素によって所有される、特定の属性名および可能な子要素は、その要素の型に依存する。各属性値の解釈は、対応する属性名およびその属性を所有する要素に依存する。

【0016】

図1からわかるように、XMT-Aファイル100は、2つの部分すなわち、Header要素110およびBody要素120からなる。Header要素110に、InitialObjectDescriptor要素130と定義された単一の子要素が含まれる。Body要素120に、子要素として1つまたは複数の「par」要素140が含まれる。

【0017】

InitialObjectDescriptorは、1つの属性、ObjectDescriptorID (ODID) 130を有し、その値は、文字ストリングである。図2からわかるように、この要素は、2つの子すなわち、Profiles要素150およびDescr要素160を有する。Profiles要素150は、子要素を有しない。Profiles要素150は、「includeInclineProfileLevelFlag」、「sceneProfileLevelIndication」、「ODProfileLevelIndication」、「audioProfileLevelIndication」、「visualProfileLevelIndication」、「graphicsProfileLevelIndication」を含む複数の属性を所有する。

【0018】

Descr要素160は、複数の型の子要素を有することができる。本発明に必須の唯一の型が、単一の「esDescr」要素170である。esDescr要素170は、1つまたは複数の「ES__Descriptor」子要素180および190を所有することができる。ES__Descriptor要素は、MPEG-4ドキュメンテーションで定義される概念である「エレメンタリ・ストリーム (elementary stream)」のあるプロパティを指定する。ES__Descriptor要素の構造を、下で示す。

【0019】

InitialObjectDescriptor要素130に従属するesDescr要素170は、1つまたは2つのES__Descriptor要素180および190を所有することができる。どの場合でも、「sdsm」または「scene_descriptor_stream」として定義されたエレメンタリ・ストリームのES__Descriptor 180が存在しなければならない。さらに、「odsm」または「object_descriptor_stream」として定義されたエレメンタリ・ストリームの第2のES__Descriptor 190を設けることができる。odsmのES__Descriptor要素190は、オーディオ・データ、ビジュアル・データ、またはsdsm内で指定されない他のタイプのメディア・データに依存するXMT-Aファイルに限って必要である。

【0020】

図3からわかるように、各par要素140および200に、1つまたは複数の「par-child」要素210が含まれる。「par-child」要素は、別のpar要素、odsmコマンド、またはbifsコマンドとすることができる。各par要素に、名前「begin」を有する属性も含まれる。begin属性の値は、par要素内のodsmコマンドまたはbifsコマンドが実行される時を指定する。par要素のbegin属性によって決定される時間値は、親によって暗示される時間値に関して計算され、Body要素120は、0の開始時刻を暗示する。

【0021】

par-child要素210に、図4に示されているように、2つの型のodsmコマンド要素のインスタンスを含めることができる。これには、ObjectDescriptorUpdate要素220およびObjectDescriptorRemove

10

20

30

40

50

要素250が含まれる。ObjectDescriptorUpdate要素220には、単一のOD子要素230が含まれ、OD要素230には、単一のObjectDescriptor子要素240が含まれる。ObjectDescriptor要素240は、下で詳細に説明する。ObjectDescriptorRemove要素250は、1つの属性を有し、子要素を有しない。ObjectDescriptorRemove要素250の属性は、「ODID」という名前を有する。

【0022】

図5から7に示されているように、par-child要素210に、3つの型のbifsコマンド要素のインスタンスを含めることができる。これには、Insert要素300、Delete要素310、およびReplace要素320が含まれる。図5からわかるように、Insert要素300は、「xmtaBifsNode」子要素330または「ROUTE」子要素340のいずれかを有することができる。Delete要素310は、子を有しない。Replace要素320は、「xmtaBifsNode」350子要素、「ROUTE」子要素360、または「Scene」子要素370を有することができる。Scene要素は、「xmtaTopNode」子要素380を有する。Scene要素は、1つまたは複数のROUTE子要素390も有することができる。

【0023】

ROUTE要素340および390は、子を有しない。ROUTE要素340および390の属性に、「fromNode」、「fromField」、「toNode」、および「toField」が含まれる。

【0024】

用語「xmtaBifsNode要素」330は、約100個の定義済みBIFS Node要素のいずれか1つを表す。これらのそれぞれが、図8に示された全体的な構造400を有する。各xmtaBifsNode要素400は、MPEG-4 Systems仕様書（ISO-IEC文書ISO/IEC 14496-1:2001、2001年8月、第9章）で定義されたバイナリ・データ構造体であるBIFS Nodeを表す。この文書に関する情報は、<http://mpeg.telecomitalia.com/documents.htm>およびスイス国ジュネーブ20、CH-1211、1、rue de Varembe、Case postal 56の国際標準化機構（ISO）で入手可能である。各xmtaBifsNode要素400の要素タグは、MPEG-4 Systems仕様で定義された対応するNodeNameに基づく。いくつかの型のxmtaBifsNode要素は、対応するBIFS Nodeのあるプロパティに基づいて従属（子）要素を有することができる。これを、nodeField要素410と呼ぶ。各nodeField要素は、さらなるxmtaBifsNode要素420からなる1つまたは複数の従属要素を有することができる。この配置を、BIFSノードの階層木を記述するために再帰的に繰り返すことができる。この階層の深さに制限はない。

【0025】

各BIFS Nodeは、「フィールド」と称する複数のプロパティを有する。これらのフィールドのそれぞれが、定義されたフィールド名（ストリング）およびフィールド・データ型（boolean、integer、floatなど）を有する。フィールド・データ型の1つが、「Node」である。Node以外のフィールド・データ型のすべてが、xmtaBifsNode要素400の同一の名前の属性によって表される。型「Node」を有する各フィールドは、xmtaBifsNode要素400の同一の名前の子要素410によって表される。xmtaBifsNode要素400の各子要素410は、子要素として1つまたは複数のxmtaBifsNode要素420（xmtaBifsNode親要素400の孫）を有することができる。

【0026】

XMT-A BIFS NodeのXML表現を、図9から10に示す。各XMT-A BIFS Node要素は、100個を超える可能な型のXMT-A BIFSノードの1つを一意に識別するNodeNameタグ500および570によって識別される。

各ノード要素は、オリジナルのノード要素 500 または再利用されるノード要素 570 とすることができる。オリジナルのノード要素 500 の場合に、任意選択の属性「DEF」510 を使用して、特定のノードの一意の英数字記述を提供することができる。この属性が設けられる場合に、そのノードは、「再利用可能」として分類される。

【0027】

オリジナルの XMT-A BIFS ノード要素は、フィールド属性 520 の組も所有し、各プロパティ・フィールドの 1 つのフィールド属性が、型 `NodeName` であり、「`node`」または「`buffer`」以外のノード・データ型を有するノードについて定義される。これらの属性は、図 9 では「`field0`」、「`field2`」、「`field3`」、および「`field5`」として識別される。これらの属性のそれぞれの実際の名前は、型「`NodeName`」のノードに関する MPEG-4 Systems 仕様で定義された対応するプロパティ・フィールド名によって決定される。これらの属性のそれぞれに割り当てられる値は、MPEG-4 Systems 仕様で定義されたノード・データ型（`boolean`、`integer`、`float` など）を有するデータ値を表さなければならない。

10

【0028】

さらに、オリジナルの XMT-A BIFS ノード要素 500 は、「`node`」または「`buffer`」のデータ型を有するプロパティ・フィールドのフィールド名に対応する要素タグを有する 1 つまたは複数のフィールドー値子要素 530 および 540 を有することができる。各そのようなフィールドー値要素は、開始タグ 530 および終了タグ 540 を有する。そのようなフィールドー値要素 530 および 540 の例が、図 9 の要素タグ `<field1>...</field1>` および `<field4>...</field4>` によって表されている。

20

【0029】

データ型「`node`」を有するプロパティ・フィールドの場合に、フィールドー値要素に、BIFS-Node 要素 550 に対応する 1 つまたは複数の子要素を含めることができる。そのような BIFS-Node 子の例が、要素タグ `<NodeName1 ... />`、`<NodeName2 ... />`、および `<NodeName3 ... />` によって表される。

30

【0030】

データ型「`buffer`」を有するプロパティ・フィールドの場合に、フィールドー値要素に、BIFS コマンド要素 300、310、および 320 に対応する 1 つまたは複数の子要素を含めることができる。

【0031】

XMT-A BIFS Node 要素に、フィールドー値子要素が含まれる場合に、その Node 要素は、標準的な XML の原理に従って、`</NodeName>` 終了タグ 560 によって終端される。

【0032】

XMT-A BIFS ノード要素の前述の定義は、従属 BIFS ノード要素（`<NodeName1>` など）のそれぞれに再帰的に適用され、ノードの階層木を作成できるようになる。XMT-A BIFS ノード要素のこの木の深さに制限はない。

40

【0033】

再利用されるノード 570 の場合に、ノード要素は、1 つの属性だけを有し、子を有しない。唯一の属性は、「USE」属性 580 であり、その値 590 は、ノード ID スtring である。USE 属性の値として提供されるノード ID スtring は、同一の `NodeName` を有するオリジナルのノード要素 500 の DEF 属性 510 として指定されたノード ID スtring と一致しなければならない。

【0034】

用語「`xmtaTopNode`」は、Scene 要素に対する子要素としてサービスすることを許容される `xmtaBifsNode` 要素の定義されたサブセットの 1 つを表す

50

。

【0035】

図11からわかるように、ObjectDescriptor要素240および600 (ObjectDescriptorUpdate要素220の孫)は、上で説明したInitialObjectDescriptor要素130に類似する。InitialObjectDescriptor要素130と異なって、ObjectDescriptor要素240および600には、Profiles子要素150が欠けている。InitialObjectDescriptor要素130と同様に、ObjectDescriptor要素240および600は、「ObjectDescriptorID」 (ODID) 属性606を有する。通常のObjectDescriptor要素240 10
 および600は、単一のDescr子要素610を有し、Descr要素610は、単一のesDescr子要素620を有し、esDescr要素620は、単一のES_Descriptor子要素630を有する。Descr要素610、esDescr要素620、およびES_Descriptor要素630は、InitialObjectDescriptor要素130の対応する子160、170、180、および190に類似する。

【0036】

ES_Descriptor要素180、190、および630を、ObjectDescriptor要素600またはInitialObjectDescriptor要素130のいずれかに含めることができる。どちらの場合でも、ES_Descriptor要素180、190、および630は、図12に示された構造640を有する。ES 20
 _Descriptor要素640の「ES_ID」属性636の値は、各ストリームに固有の英数字ストリングである。ES_Descriptor640要素は、必ず、decConfigDescr子要素646およびslConfigDescr子要素660を有する。ES_Descriptor要素630および640が、ObjectDescriptor要素600に従属する場合に、そのES_Descriptor要素630および640は、StreamSource子要素670も有する。ES_Descriptor要素180、190、および640が、InitialObjectDescriptor要素130に属する場合に、そのES_Descriptor要素180、 30
 190、および640は、StreamSource子要素670を有しない。

【0037】

decConfigDescr要素646は、DecoderConfigDescriptor子要素650を有する。DecoderConfigDescriptor要素650は、親のES_Description要素640が、オーディオ、ビジュアル、sds、ods、または他のタイプのメディアのどれを表すかを示す、「streamType」および「objectTypeIndication」を含む複数の属性を有する。DecoderConfigDescription要素650は、streamTypeおよびobjectTypeIndicationの値に応じて、decSpecificInfo子要素656も有することができる。

【0038】

sds (scene description stream) のES_Descriptor要素180の場合に、DecoderConfigDescriptor650要素は、decSpecificInfo子要素656を有する。図13からわかるように、decSpecificInfo680要素は、BIFSConfig子要素686を有する。BIFSConfig要素686は、BIFS Nodeをエンコードする形を指定する複数の属性を有する。BIFSConfig要素686は、commandStream要素690も所有し、commandStream要素690は、「size」要素696を所有する。

【0039】

slConfig要素660は、SLConfigDescriptor子要素666 50

を有する。SLConfigDescriptor要素666は、「predefined」という名前の1つの属性を有し、子要素を有しない。「predefined」属性は、必ず「2」の値を有する。

【0040】

StreamSource要素670は、1つの属性「url」を有し、子要素を有しない。url属性の値は、特定のストリームの実際のサウンド、イメージなどを定義するオーディオ・データ、ビジュアル・データ、または他のデータを含むメディア・データ・ファイルの位置を示す、ファイル名またはインターネット・アドレス（URL、Uniform Resource Locator）のいずれかを指定する。StreamSource要素670は、sdsm（scene description stream）またはodsm（object descriptor stream）の場合には存在しない。というのは、これらのストリームの両方が、XMT-Aファイルによって決定されるからである。

【0041】

2.0 MPEG-4 Intermedia Formatファイル

MPEG-4 Intermedia Formatファイルは、MPEG-4 Systems仕様書、ISO-IEC文書ISO/IEC 14496-1:2001、2001年8月の第13章で定義された構造および構成を有する電子データの1形式である。電子データ構造のこの形式は、一般に「バイナリ・ファイル」と称するものの例である。というのは、これが、文字、数字、および句読点の表現に制限されないバイナリ・データ値のシーケンスからなるからである。これによって、XMT-Aファイルなどの通常のテキスト・ファイルによって可能になるものよりコンパクトなデータ構造が可能になる。MPEG-4 Intermedia Formatによって定義された構造を有する電子データの保管された形が、「mp4バイナリ・ファイル」である。XMT-Aファイルと異なって、mp4バイナリ・ファイルは、ほとんどのテキスト編集ソフトウェアによって解釈することができない。

【0042】

MPEG-4 Intermedia Formatは、1996にアップル・コンピュータ（Apple Computers, Inc.）社によって定義された、<http://developer.apple.com/techpubs/quicktime/qtdevdocs/REF/refFileFormat96.htm>および<http://developer.apple.com/techpubs/quicktime/qtdevdocs/PDF/QTFileFormat.pdf>でオンラインで入手できるQuickTime（R）ファイル・フォーマットから派生した。QuickTimeは、アップル・コンピュータ社の登録商標である。

【0043】

QuickTime（R）の継承のゆえに、MPEG-4 Intermedia Formatは、QuickTime（R）仕様から派生した複数の特性を保持する。この特性に、データ構造の単位としての「アトム」の概念が含まれる。各アトムは、2つの部分すなわち、ヘッダおよび本体を有する。ヘッダには、ヘッダを含むアトムを構成するバイト数を指定するアトム・サイズ値が含まれる。ヘッダには、アトムの型を指定するatomIdも含まれる。アトムの本体には、アトムによって伝えられるデータが含まれる。このデータに、従属アトムを含めることができる。その最も基本的な形で、アトムは、4バイト（符号なし整数）からなるアトム・サイズ値と、やはり4バイト（文字）からなるatomIdを有する。4バイトを超えるアトム・サイズ値およびatomIdを有する、アトムの拡張された形も、MPEG-4仕様で定義されている。

【0044】

図14からわかるように、mp4バイナリ・ファイル700は、1つまたは複数の「mdat」アトム706および1つの「moov」アトム712からなる。moovアトム712は、mdatアトム706の前または後に置くことができる。図15からわかるように、各mdatアトム718は、アトム・サイズ値724と、それに続く4バイトのatomId「mdat」730と、「チャンク」736と呼ばれるデーバ・ブロックのシーケンスからなる。図16からわかるように、各チャンク742は、メディア・データ「

サンプル」748のシーケンスからなる。各サンプル748は、単一のメディア・ストリームに関する、特定の時点に関連するデータのブロックを指定する。単一のチャンク内のすべてのサンプルが、同一のメディア・データ・ストリームを表さなければならない。mdatアトム700の観察から、個々のサンプル748またはチャンク736および742を識別することは、必ずしも可能でない。各サンプル748ならびにチャンク736および742は、mp4バイナリ・ファイル内の他の場所に保管されたテーブルを使用して識別することができる。

【0045】

図17からわかるように、moovアトム758は、アトム・サイズ値760と、それに続く4バイトのatomId「moov」766ならびに「mvhd」(moov header)アトム772、「iods」(initial object descriptor)アトム778、および1つまたは複数の「trak」アトム790を含む複数の従属アトムからなる。「moov」アトム712および758には、データ・ストリームごとに1つの「trak」アトム790が含まれ、この「trak」アトム790には、存在する場合にsdsm(scene description stream)およびodsm(object descriptor stream)が含まれる。「moov」アトム712および758に、任意選択の「udta」(user data)アトム784も含めることができる。「udta」アトム784は、著作権メッセージなどの任意選択の情報をmp4バイナリ・ファイルに埋め込むのに使用することができる。

【0046】

mvhdアトム772は、アトム・サイズ値と、それに続く4バイトのatomId「mvhd」ならびに日付時刻スタンプ、時間スケール値、およびファイル持続時間値を含む複数のデータ値からなる。mvhdアトムのアトム・サイズ値は、必ず108である。日付時刻スタンプは、ファイルが作成された時を示す。時間スケール値は、そのファイルの時間値を表すのに使用される毎秒のティック数を示す。ファイル持続時間値は、時間スケール値で指定される時間単位での、ファイル内の材料を表現するのに必要な総時間を示す。

【0047】

図18からわかるように、iodsアトム800は、アトム・サイズ値804と、それに続く4バイトのatomId「iods」808、8ビットのバージョン値812、24ビットのフラグ値816、およびMp4fInitObjDescrデータ構造820からなる。図19からわかるように、Mp4fInitObjDescrデータ構造824は、1バイトのMP4_IOD_TAG値828、後続データ・ブロックのバイト数832、10ビットのObjectDescriptorID 836、2つのフラグ・ビット840および844、4つの予約済みビット848、ならびに5つのプロファイル・レベル表示値852、856、860、864、および868からなる。プロファイル・レベル表示値の後に、1つまたは2つのMPEG-4 ES_ID_Incデータ構造872が続く。1つのES_ID_Inc構造は、sdsm(scene description stream)に対応するtrakアトム790のES_ID値を示す。第2のES_ID_Inc構造は、存在する場合に、odsm(object descriptor stream)に対応するtrakアトム790のES_IDを示す。第2のES_ID_Inc構造は、odsmが存在する場合に限って存在する。図20からわかるように、各ES_ID_Incデータ構造は、1バイトのES_ID_IncTag値880、後続データ内のバイト数884(必ず4)、および32ビットのES_ID値888からなる。

【0048】

図21からわかるように、各trakアトム900は、アトム・サイズ値903と、それに続く4バイトのatomId「trak」906、「tkhd」(track header)アトム910、および「mdia」(media)アトム912からなる。odsm(object descriptor stream)を表すtrakアトムの

10

20

30

40

50

場合に、`trak`アトムに、「`tref`」(`track reference`)アトム940も含まれる。遅れた開始を有するトラックの場合に、「`edts`」(`edit list`)アトム945が、トラックを開始する時を示すために設けられる。`mdia`アトム912は、「`mdhd`」(`media header`)アトム915、「`hdlr`」(`handler`)アトム918、「`minf`」(`media information`)アトム920、「`stbl`」(`sample tables`)アトム933、およびメディア情報ヘッダ・アトム936からなる。ラベル「`*mhd`」は、「`nmhd`」(`sdsm`トラックおよび`odsm`トラック)、「`smhd`」(オーディオ・トラック)、「`vmhd`」(ビジュアル・トラック)などを含む複数のメディア情報ヘッダ・アトム型のいずれか1つを表す。

10

【0049】

`tkhd`アトム910、`mdhd`アトム915、および`hdlr`アトム918に、`trackId`番号、時刻日付スタンプ、メディア時間スケール、およびメディア持続時間値を含む複数のデータ値が含まれる。各トラックは、それ自体の時間スケールを有し、この時間スケールは、`mvhd`アトム772で指定されるグローバル時間スケールと異なるものとすることができる。

【0050】

図22からわかるように、`sample tables`アトム950は、アトム・サイズ値954と、それに続く、4バイトの`atomId`「`stbl`」957、一連のサンプル・テーブル・アトム960、963、966、970、974、および978からなる。さまざまなサンプル・テーブル・アトムは、任意の順序とすることができる。これには、「`stsc`」(`sample-to-chunk table`)アトム960、「`stts`」(`time-to-sample table`)アトム963、「`stco`」(`chunk offset table`)アトム966、「`stsz`」(`sample size table`)アトム970、可能な「`stss`」(`sync sample table`)アトム974、および「`stsd`」(`sample description table`)アトム978が含まれる。これらのサンプル・テーブル・アトムのそれぞれに、関連する`mdat`アトム706および718に保管されたバイナリ・メディア・データ736および748のプロパティを記述するデータが含まれる。

20

【0051】

`sample-to-chunk table`アトム(`stsc`アトム)960は、アトム・サイズ値、4バイトの`atomId`(「`stsc`」)、32ビット符号なし整数(`numStscEntries`)、および`sample-to-chunk`データ・レコードのシーケンスが含まれる。`numStscEntries`の値は、`sample-to-chunk`データ・レコードのシーケンスの項目数を指定する。各`sample-to-chunk`データ・レコードは、開始チャンク番号、チャンクごとのサンプル数、およびサンプル記述インデックスを指定する、3つの32ビット符号なし整数からなる。サンプル記述インデックスは、`sample description table`978内の項目へのインデックスである。チャンクごとのサンプル数は、開始チャンク番号によって指定されるチャンク736内および次の項目によって指定される開始チャンクの前のすべての後続チャンク内のサンプル748の個数を指定する。

30

40

【0052】

`time-to-sample table`アトム(`stts`アトム)963は、アトム・サイズ値、4バイトの`atomId`(「`stts`」)、32ビット符号なし整数(`numSttsEntries`)、および`time-to-sample`データ・レコードのシーケンスからなる。`numSttsEntries`の値は、`time-to-sample`データ・レコードのシーケンス内の項目数を指定する。各`time-to-sample`データ・レコードは、サンプル・カウントおよびトラック時間スケール単位でのサンプル持続時間を指定する、2つの32ビット符号なし整数からなる。サンプル・カウント値は、対応するサンプル持続時間を有する連続するサンプル748の個数を指定する。

50

【0053】

chunk offset tableアトム (stcoアトム) 966は、アトム・サイズ値、4バイトのatomId (「stco」)、32ビット符号なし整数 (numStcoEntries)、およびチャンク・オフセット値のシーケンスからなる。numStcoEntriesの値は、チャンク・オフセット値のシーケンス内の項目の個数を指定する。このシーケンスの各項目は、mp4ファイルの先頭と、mdatアトム718内の対応するチャンク736の先頭との間のバイト数を指定する1つの32ビット符号なし整数からなる。

【0054】

sample size tableアトム (stszアトム) 970は、アトム・サイズ値、4バイトのatomId (「stsz」)、このtrakアトム900に関連するすべてのメディア・データ・サンプル748のサイズを指定する32ビット符号なし整数 (iSampleSize) からなる。このtrakアトムに関連するメディア・データ・サンプルが、サイズにおいてすべて同一ではない場合に、iSampleSizeの値は、0と指定され、その次に、32ビット符号なし整数 (numStszEntries) およびサンプル・サイズ値のシーケンスが続く。numStszEntriesの値は、サンプル・サイズ値のシーケンス内の項目の個数を指定する。このシーケンスの各項目は、このtrakアトム900に関連するメディア・データ718内の対応するサンプル748内のバイト数を指定する1つの32ビット符号なし整数からなる。

【0055】

sync sample tableアトム (stssアトム) 974は、存在する場合に、アトム・サイズ値、4バイトのatomId (「stss」)、32ビット符号なし整数 (numStssEntries)、およびサンプル・インデックス値のシーケンスからなる。numStssEntriesの値は、サンプル・インデックス値のシーケンス内の項目の個数を指定する。このシーケンスの各項目は、「ランダム・アクセス・サンプル」のサンプル・インデックスを指定する1つの32ビット符号なし整数からなる。ランダム・アクセス・サンプル・インデックスは、このtrakアトム900に関連するメディア・データ内の、メディア・プレイヤーが前のサンプルに無関係にメディア・データの処理を開始できる点に対応するメディア・データ・サンプル748を識別する。このテーブルのサンプル・インデックス値は、単調に増加しなければならない。

【0056】

sample description tableアトム (stsdアトム) 978は、アトム・サイズ値、4バイトのatomId (「stsd」)、32ビット符号なし整数 (numStsdEntries)、およびサンプル記述データ・レコードのシーケンスからなる。numStsdEntriesの値は、サンプル記述データ・レコードのシーケンス内の項目の個数を指定する。各サンプル記述データ・レコードは、sample-to-chunkデータ・レコード内の対応するインデックスによって指定されるメディア・データ・サンプルをエンコードするのに使用された手段を指定する。サンプル記述データ・レコードのシーケンスは、通常、単一の項目を有し (numStsdEntries=1)、この項目は、メディアのタイプ (オーディオ、ビジュアル、sdsm、odsm)、オーディオ・サンプルおよびビデオ・サンプルに使用された圧縮アルゴリズムなどを指定する。各サンプル記述テーブル項目は、「mp4*」アトム982に含まれ、この「mp4*」は、「mp4s」 (sdsmサンプルおよびodsmサンプル)、「mp4a」 (オーディオ・サンプル) 、および「mp4v」 (ビジュアル・サンプル) の包括的な代理である。各「mp4*」アトム982に、「esds」 (elementary stream descriptor) アトム986が含まれ、各esdsアトム986に、MPEG-4エレメンタリ・ストリーム記述子 (Es_Descr) データ構造990が含まれる。

【0057】

MPEG-4エレメンタリ・ストリーム記述子1000の構造を、図23に示す。この

10

20

30

40

50

データ構造は、1バイトのタグ (E S _ D e s c r T a g) 1 0 0 4 と、それに続く、このデータ構造の残りのバイト数の表示 1 0 0 8、16ビットの E S _ I D 値 (通常は 0) 1 0 1 2、3つの1ビット・フラグ (s t r e a m D e p e n d e n c e F l a g、U R L _ F l a g、および O C R s t r e a m F l a g) 1 0 1 6、および5ビットのストリーム優先順位値 1 0 2 0 からなる。3つのフラグ 1 0 1 6 のいずれかが非 0 である場合に、追加のデータ値 (図示せず) をストリーム優先順位値 1 0 2 0 に続けることができる。これらの任意選択のデータ値は、本発明に必要でない。

【0058】

ストリーム優先順位値 1 0 2 0 に、デコーダ構成記述子データ構造 1 0 2 4 および同期レイヤ構成記述子データ構造 1 0 2 8 が続く。デコーダ構成記述子 1 0 2 4 の構造 1 0 3 2 を、図 2 4 に示す。このデータ構造は、1バイトのタグ (D e c o d e r C o n f i g D e s c r T a g) 1 0 3 6 と、それに続く、このデータ構造の残りのバイト数の表示 1 0 4 0、および一連のデータ値からなり、このデータ値には、o b j e c t T y p e 1 0 4 4、s t r e a m T y p e 1 0 4 8、u p S t r e a m ビット 1 0 5 2、予約済みビット 1 0 5 6、b u f f e r S i z e D B 1 0 6 0、m a x B i t r a t e 1 0 6 4、および a v g B i t r a t e 1 0 6 8 が含まれる。これらの値の後に、s t r e a m T y p e および o b j e c t T y p e に依存するデコーダ固有情報データ構造 1 0 7 2 を続けることができる。デコーダ固有情報データ構造 1 0 7 2 は、s d s m に必要であるが、o d s m には不要である。ほとんどのオーディオおよびビジュアルのメディア・データ・ストリームも、デコーダ構成記述子 1 0 3 2 内にデコーダ固有情報データ構造を有する。

【0059】

デコーダ固有情報 1 0 7 2 の構造 1 0 7 6 を、図 2 5 に示す。このデータ構造は、1バイトのタグ (D e c o d e r S p e c i f i c I n f o T a g) 1 0 8 0 と、それに続く、このデータ構造の残りのバイト数の表示 1 0 8 4 からなる。残りのバイトは、o b j e c t T y p e および s t r e a m T y p e に依存する。s d s m (s c e n e d e s c r i p t i o n s t r e a m または B I F S) の場合に、デコーダ固有情報 1 0 7 2 および 1 0 7 6 に、n o d e I D 値のエンコードに使用されたビット数の表示と、r o u t e I D 値のエンコードに使用されたビット数の表示が含まれる。これらの値のそれぞれが、5ビットの符号なし整数によって表される。

【0060】

同期レイヤ構成記述子 1 0 2 8 の構造 1 0 8 8 を、図 2 6 に示す。このデータ構造は、1バイトのタグ (S L C o n f i g D e s c r T a g) 1 0 9 0 と、それに続く、このデータ構造の残りのバイト数の表示 1 0 9 4 (必ず 1) および単一のデータ・バイト (値 2 すなわち「p r e d e f i n e d」) 1 0 9 8 からなり、このデータ・バイトは、同期レイヤに事前定義の構成が使用されなければならないことを示す。

【0061】

3.0 Scene Description Stream (s d s m)

特定のタイプのオーディオ・データ・ストリームおよびビジュアル・データ・ストリームのエンコードまたはデコードに使用される手段は、X M T - A 仕様および M P E G - 4 I n t e r m e d i a F i l e 仕様によって決定されず、したがってこれらのストリームをエンコードする形の詳細には、本明細書では触れない。X M T - A 文書に、s t r e a m d e s c r i p t i o n s t r e a m (s d s m) および o b j e c t d e s c r i p t i o n s t r e a m (o d s m) の内容に関する詳細な情報が含まれる。その結果、各 X M T - A 文書は、M P E G - 4 I n t e r m e d i a F i l e に含まれる s d s m ストリームおよび o d s m ストリームに密接に関連する。このセクションでは、s d s m データの構造を説明し、次のセクションで、o d s m データの構造を説明する。

【0062】

他のメディア・データ・ストリームと同様に、s d s m データは、1つまたは複数のチ

ヤンクからなり、各チャンクは、1つまたは複数のサンプルからなる。図27からわかるように、sdsmバイナリ・チャンク1100内の各サンプルは、「コマンド・フレーム」1110として定義される。各コマンド・フレーム1110は、バイト整列される。図28からわかるように、各コマンド・フレーム1110は、1つまたは複数の「BIFSコマンド」1120からなる。BIFSは、「Binary Format for Streams」の略である。各BIFSコマンド1120に、継続ビット1130および1140が続く。継続ビットの値が(1)1130である場合に、もう1つのBIFSコマンドが続く。そうでない場合1140に、継続ビットに、最後のバイトを完了するのに十分な個数のヌル・パディング・ビット1150が続く。個々のBIFSコマンド1120は、コマンド・フレームの最初のBIFSコマンドを除いて、一般にバイト整列されない。

10

【0063】

図29から32からわかるように、4タイプのBIFSコマンドすなわち「挿入」、「削除」、「置換」、「シーン置換」がある。BIFS挿入コマンド1200は、2ビットの挿入コード(値=「00」)1206と、それに続く2ビットのパラメータ型コード1210および挿入コマンド・データ1216からなる。BIFS削除コマンド1220は、2ビットの削除コード(値=「01」)1226と、それに続く2ビットのパラメータ型コード1230および削除コマンド・データ1236からなる。BIFS置換コマンド1240は、2ビットの置換コード(値=「10」)1244と、それに続く2ビットのパラメータ型コード1250および置換コマンド・データ1260からなる。BIFSシーン置換コマンド1270は、2ビットのシーン置換コード(値=「11」)1280と、それに続くBIFSシーン・データ構造1290からなる。

20

【0064】

図33から35からわかるように、3タイプのBIFS挿入コマンドすなわち、(a) Node Insertionコマンド、(b) Indexed Value Insertionコマンド、および(c) Route Insertionコマンドがある。挿入コマンドのタイプは、パラメータ型値1210によって決定される。Node Insertionコマンド1300は、2ビットの挿入コード(値=「00」)1304と、それに続く、2ビットのパラメータ型コード(値=「01」、型=Node)1308、nodeID値1312、2ビットの挿入位置コード1316、およびSFNodeデータ構造1324からなる。挿入位置コード1316の値が0である場合には、8ビットの位置値1320が、挿入位置コード1316に続く。nodeID値1312は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。これおよび他のnodeID値をエンコードするのに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。SFNodeデータ構造1324の構造は、下で説明する。

30

【0065】

Indexed Value Insertionコマンド1328は、2ビットの挿入コード(値=「00」)1332と、それに続く、2ビットのパラメータ型コード(値=「10」、型=Indexed Value)1336、nodeID値1340、inFieldID値1344、2ビットの挿入位置コード1348、およびフィールド値データ構造1356からなる。挿入位置コード1348の値が0である場合に、8ビットの位置値1352が、挿入位置コード1348に続く。nodeID値1340は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1340のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。inFieldID値1344は、nodeID1340の値によって指定されるBIFSノードのデータ・フィールドの1つを識別する。inFieldID値1344をエンコードするのに使用されるビット数は、MPEG-4 Systems仕様に含まれるテーブルに依存する。

40

【0066】

50

フィールド値データ構造の内容は、指定されたBIFSノードの指定されたデータ・フィールドのフィールド・データ型(boolean、integer、float、string、nodeなど)に依存する。これは、1ビットのように単純なものまたはSFNodeデータ構造のように複雑なものとするができる。各BIFSノードの各データ・フィールドのフィールド・データ型は、MPEG-4 Systems仕様に含まれるテーブルで指定される。

【0067】

Route Insertionコマンド1360は、2ビットの挿入コード(値=「00」)1364と、それに続く、2ビットのパラメータ型コード(値=「11」、型=Route)1368、「isUpdateable」ビット1372、departureNodeID値1380、departureFieldID値1384、arrivalNodeID値1388、およびarrivalFieldID値1392からなる。「isUpdateable」ビットの値が(1)である場合に、routeID値1376が、「isUpdateable」ビット1372に続く。departureNodeID値1380およびarrivalNodeID値1388のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。departureFieldID値1384のエンコードに使用されるビット数およびarrivalFieldID値1392のエンコードに使用されるビット数は、MPEG-4 Systems仕様に含まれるテーブルに依存する。

【0068】

図36から38からわかるように、3タイプのBIFS削除コマンドすなわち(a)Node Deletionコマンド、(b)Indexed Value Deletionコマンド、および(c)Route Deletionコマンドがある。削除コマンドのタイプは、パラメータ型値1230によって決定される。Node Deletionコマンド1400は、2ビットの削除コード(値=「00」)1406と、それに続く、2ビットのパラメータ型コード(値=「00」、型=Node)1412およびnodeID値1418からなる。nodeID値1418は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1418のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。

【0069】

Indexed Value Deletionコマンド1424は、2ビットの削除コード(値=「01」)1430と、それに続く、2ビットのパラメータ型コード(値=「10」、型=Indexed Value)1436、nodeID値1442、inFieldID値1448、および2ビットの削除位置値1454からなる。nodeID値1418は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1418のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。

【0070】

Route Deletionコマンド1466は、2ビットの削除コード(値=「10」)1472と、それに続く、2ビットのパラメータ型コード(値=「11」、型=Route)1478およびrouteID値1484からなる。routeID値1484は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。routeID値1484のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。

【0071】

図39から42からわかるように、4タイプの置換コマンドすなわち、(a)Node Replacementコマンド、(b)Field Replacementコマンド、(c)Indexed Value Replacementコマンド、および(d)Route Replacementコマンドがある。挿入コマンドのタイプは、パラメータ型値1210によって決定される。Node Replacementコマンド1

500は、2ビットの置換コード（値＝「10」）1504と、それに続く、2ビットのパラメータ型コード（値＝「01」、型＝Node）1508、nodeID値1510、およびSFNodeデータ構造1514からなる。nodeID値1510は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1510のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。SFNodeデータ構造1514の構造は、下で説明する。

【0072】

Field Replacementコマンド1520は、2ビットの置換コード（値＝「10」）1524と、それに続く、2ビットのパラメータ型コード（値＝「01」、型＝Field）1528、nodeID値1530、inFieldID値1534、およびフィールド値データ構造1538からなる。nodeID値1530は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1530のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。inFieldID値1534は、nodeID1530の値によって指定されるBIFSノードのデータ・フィールドの1つを識別する。inFieldID値1534のエンコードに使用されるビット数は、MPEG-4 Systems仕様に含まれるテーブルに依存する。

10

【0073】

Indexed Value Replacementコマンド1540は、2ビットの置換コード（値＝「10」）1544と、それに続く、2ビットのパラメータ型コード（値＝「10」、型＝IndexedValue）1548、nodeID値1550、inFieldID値1554、2ビットの置換位置コード1558、およびフィールド値データ構造1564からなる。置換位置コード1558の値が0である場合に、8ビットの位置値1560が、置換位置コード1558に続く。nodeID値1550は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。nodeID値1550のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。inFieldID値1554は、nodeID1550の値によって指定されるBIFSノードのデータ・フィールドの1つを識別する。inFieldID値1554のエンコードに使用されるビット数は、MPEG-4 Systems仕様に含まれるテーブルに依存する。

20

30

【0074】

Route Replacementコマンド1570は、2ビットの置換コード（値＝「10」）1574と、それに続く、2ビットのパラメータ型コード（値＝「11」、型＝Route）1578、routeID値1580、departureNodeID値1584、departureFieldID値1588、arrivalNodeID値1590、およびarrivalFieldID値1594からなる。routeID値1580は、BIFSコマンドの他所で定義された更新可能なノードの組の1つを指定する。routeID値1580のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。departureNodeID値1584およびarrivalNodeID値1590のエンコードに使用されるビット数は、sdsmストリームのデコーダ固有情報1072で指定される。departureFieldID値1588のエンコードに使用されるビット数およびarrivalFieldID値1594のエンコードに使用されるビット数は、MPEG-4 Systems仕様に含まれるテーブルに依存する。

40

【0075】

図32からわかるように、BIFSシーン置換コマンド1270は、2ビットのシーン置換コード（値＝「11」）1280と、それに続くBIFSシーン・データ構造1290からなる。図43からわかるように、BIFSシーン・データ構造1600は、6ビットの予約済みフィールド1610、2つの1ビット・フラグ（USENAMES 1620およびprotoList 1630）、SFTopNodeデータ構造1640、お

50

よび1ビット・フラグ (hasRoutes) 1650からなる。protoListフラグ1630が真(1)の場合に、MPEG-4 Systems仕様で定義された追加データが、protoListフラグに続く。SFNodeデータ構造1640は、図44から48に示されたSFNodeデータ構造の特殊な例である。hasRoutesフラグ1650が真(1)の場合に、Routesデータ構造1660が、hasRoutesフラグに続く。Routesデータ構造の構造を、図49から51に示す。

【0076】

図44、45、および46に示されているように、SFNodeデータ構造は、3つの形すなわち、(a)再利用、(b)マスク・ノード、および(c)リスト・ノードの1つを有することができる。3つの形のすべてが、1ビットのフラグ(isReused)から始まる。再利用されるSFNode1700の場合に、isReusedフラグの値は「1」(真)1704であり、SFNodeデータ構造の残りは、nodeIDref値1708からなる。nodeIDref1708の値は、sdsmデータの他所で定義された更新可能なSFNodeのnodeID値と一致しなければならない。

10

【0077】

isReusedFlagが偽(0)1712および1732の場合に、SFNode型は、maskAccessフラグ・ビット1722および1742の値に応じて、図45および46に示された2つの形の1つを有することができる。どちらの場合でも、SFNodeのデータに、ローカル・ノード型値(localNodeType)1714および1734、1ビットのフラグ(isUpdateable)1716および1736、ならびに第2の1ビットのフラグ(maskAccess)1722および1742が含まれる。ローカル・ノード型1714および1734のエンコードに使用されるビット数は、MPEG-4 Systems仕様で指定されるテーブルに依存する。isUpdateableフラグ1716および1736が真(1)の場合に、nodeID値1718および1738がisUpdateableフラグに続く。isUpdateableフラグ1716および1736が真(1)であり、関連するBIFSシーン・データ構造1600のUSENAMESフラグ1620も真(1)である場合に、ヌル終端ストリング(「name」)1720および1740が、nodeID値1718および1738に続く。

20

【0078】

maskAccessビットが真(1)1722の場合に、SFNodeは、「マスク・ノード」構造1710を有する。この場合に、図45に示されているように、maskAccessビット1722に、localNodeType 1714の値によって与えられるノード型を有するBIFSノードのMPEG-4仕様で定義されたnFieldsプロパティ・フィールドごとに1つの、マスク・ビット1726の順序付きシーケンスが続く。これらのマスク・ビットのうちの1つが真(1)である場合のそれぞれで、マスク・ビットに、localNodeType1734によって決定されるフィールド・データ型(integer、boolean、string、nodeなど)に従ってエンコードされたバイナリ・フィールド値1728、フィールド番号(マスク・ビットのシーケンス内の位置)、およびMPEG-4仕様で定義されたテーブルが続く。

30

40

【0079】

maskAccessビットが偽(0)である場合1742に、SFNodeは、「リスト・ノード」構造1730を有する。この場合に、図46に示されているように、maskAccessビット1742に、1つまたは複数のフィールド参照レコードが続く。各フィールド参照レコードは、1ビットの終了フラグ1744および1750から始まる。終了フラグが偽(0)である場合1744に、終了フラグ1744に、ローカル・ノード型1734について定義されたプロパティ・フィールドのフィールド参照インデックス番号(fieldRef)1746が続き、fieldRef値1746に、ローカル・ノード型1734によって決定されるフィールド・データ型(integer、boolean、string、nodeなど)およびfieldRef値1746によって示さ

50

れるプロパティ・フィールドに従ってエンコードされたバイナリ・フィールド値1748が続く。fieldRef値1746のエンコードに使用されるビット数は、MPEG-4 Systems仕様で定義されたテーブルによって決定される。終了フラグが真(1)である場合1750に、フィールド値のリストが終了する。

【0080】

SFNode構造に含まれる各プロパティ・フィールド値は、単一のデータ値(SFFieldデータ構造)または複数のデータ値(MFFieldデータ構造)からなるものとすることができる。各MFFieldデータ構造に、0個以上のSFFieldコンポーネントが含まれる。図47および図48からわかるように、MFField構造には、isListビット1766および1786の値に基づいて、リスト形式1760およびベクトル形式1780の2つの形がある。両方の形式が、1ビットの予約済みビット1762および1782と、それに続くisListビット1766および1786から始まる。

【0081】

isListビットが、値(1)を有する場合1766に、MFFieldデータ構造は、リスト形式1760を有する。この場合に、isListビット1766に、1ビットのendFlag値1770および1772のシーケンスが続く。endFlagビットの値が「0」である場合1770に、endFlagビットに、SFFieldデータ構造1774が続く。endFlagビットの値が「1」である場合1772に、MFFieldデータ構造が終わる。

【0082】

isListビットが、値(0)を有する場合1786に、MFFieldデータ構造は、ベクトル形式1780を有する。この場合に、isListビットに1786、5ビットのフィールド(nBits)1790が続き、このフィールドは、次のフィールド・カウント値(nFields)1792のビット数を指定する。これに、nFields個のSFField構造1796のシーケンスが続く。

【0083】

各SFField値の構造は、MPEG-4 Systems仕様で指定されるテーブルによって示されるように、対応するプロパティ・フィールドに関連する特定のフィールド・データ型に依存する。たとえば、booleanフィールドは、単一ビットからなる。integer、float、string、SFNodeを含む他の事例が、MPEG-4 Systems仕様で定義され、説明されている。

【0084】

BIFSシーン・データ構造1600の最後のコンポーネントが、任意選択のRoutesデータ構造1660である。図49および50からわかるように、Routesデータ構造には、リスト形式1800およびベクトル形式1830の2つの形がある。Routesデータ構造の両方の形式が、1ビットのリスト・フラグ1805および1835から始まる。リスト・フラグの値が真(1)である場合1805に、Routesデータ構造は、リスト形式1800を有する。この場合に、リスト・ビット1805に、1つまたは複数のRouteデータ構造1810が続き、各Routeデータ構造1810に、1ビットのmoreRoutesフラグ1815および1820が続く。moreRoutesフラグの値が真(1)の場合1815に、もう1つのRouteデータ構造1810が続く。moreRoutesフラグの値が偽(0)の場合1820に、Routesデータ構造1800が終了する。

【0085】

Routesデータ構造のリスト・フラグの値が偽(0)である場合1835に、Routesデータ構造は、ベクトル形式1830を有する。この場合に、リスト・ビット1835に、5ビットのnBitsフィールド1840が続く。nBitsフィールドに含まれる符号なし整数値は、それに続くnumRoutes値1845のエンコードに使用されたビット数を指定する。numRoutes値1845でエンコードされた符号なし

整数は、numRoutes 値 1845 に続く Route データ構造 1850 の個数を指定する。

【0086】

図 51 からわかるように、Route データ構造 1860 は、1 ビットのフラグ (isUpdateable) 1865、outNodeID 値 1880、outFieldRef 値 1885、inNodeID 値 1890、および inFieldRef 値 1895 からなる。isUpdateable フラグ 1865 の値が真 (1) である場合に、isUpdateable フラグ 1865 に、routeID 値 1870 が続く。isUpdateable フラグ 1865 の値が真 (1) であり、対応する BIFS シーン・データ構造 1600 の USENAMES フラグ 1620 の値も真 (1) である場合に、routeID 値 1870 に、ヌル 終端 ストリング (routeName) 1875 が続く。outNodeID 値、inNodeID 値、および routeID 値のエンコードに使用されるビット数は、sdsM ストリームのデコーダ固有情報 1072 で指定される。outFieldRef および inFieldRef のエンコードに使用されるビット数は、MPEG-4 Systems 仕様で定義されるテーブルによって決定される。

【0087】

4.0 Object Descriptor Stream (odsm)

他の MPEG-4 エレメンタリ・ストリームと同様に、odsm (object descriptor stream) は、1 つまたは複数のチャンク 736 のシーケンスに含まれる。図 52 から 53 からわかるように、各 odsm チャンク 1900 は、odsm サンプル 1920 のシーケンスからなり、各 odsm サンプル 1940 は、odsm コマンド 1960 のシーケンスからなる。各 odsm チャンク 1900 の odsm サンプル 1920 の個数は、object descriptor stream の trak アトム 790 および 900 の sample-to-chunk table アトム (stsc) 960 の内容によって決定される。各 odsm サンプル 1940 の odsm コマンド 1960 の個数は、object descriptor stream の trak アトム 790 および 900 の sample size table アトム (stsz) 970 によって決定される。

【0088】

2 つの可能な odsm コマンドすなわち、ObjectDescriptorUpdate コマンド および ObjectDescriptorRemove コマンドがある。図 54 からわかるように、ObjectDescriptorUpdate コマンド 2000 は、1 バイトの ObjectDescriptorUpdateTag 2010、コマンドの残りのバイト数の表示 (numBytes) 2020、および ObjectDescriptors 2030 のシーケンスからなる。ObjectDescriptor の構造を、図 56 から 57 に要約した。図 55 からわかるように、ObjectDescriptorRemove コマンド 2040 は、1 バイトの ObjectDescriptorRemoveTag 2050、コマンドの残りのバイト数の表示 (numBytes) 2060、objectDescriptorId 値 2070 のシーケンス、および 2 ビットから 6 ビットのパディング・ビット 2080 からなる。

【0089】

各 numBytes 値 2020 および 2060 は、odsm コマンドの残りのバイト数を指定する。numBytes の値が 128 未満である場合に、この値は、単一のバイトでエンコードされる。そうでない場合には、numBytes の値は、サイズ・バイトのシーケンスでエンコードされる。各サイズ・バイトの上位ビットは、別のサイズ・バイトが続くかどうかを示す。この上位ビットが「1」である場合には、次のサイズ・バイトが続く。各サイズ・バイトの残りの 7 ビットは、numBytes の結果の符号なし整数値の 7 ビットを指定する。

【0090】

各 objectDescriptorId 値 2070 は、10 ビットでエンコードされ

、ObjectDescriptorRemoveコマンド2040で見つかる10ビットのobjectDescriptorId値のシーケンスは、バイトのシーケンスにパックされる。objectDescriptorId値の個数が4の倍数でない場合には、このコマンドの最後のバイトを充てんするために、2個、4個、または6個のヌル・ビット2080が、最後のobjectDescriptorId値に続く。

【0091】

図56からわかるように、ObjectDescriptorUpdateコマンド2000内のObjectDescriptor 2100は、1バイトのMP4_OD_Tag 2108と、それに続く、numBytes値2116、10ビットのObjectDescriptorId値2124、1ビットのURL_Flag値2132、5ビットの予約済みフィールド(0x1f)2140、およびES_Descrデータ構造またはEsIdRefデータ構造2148のいずれかからなる。ObjectDescriptorのこの形式では、URL_Flag 2132の値が、必ず偽(0)である。numBytes値2116は、このオブジェクト記述子の残りを含むバイト数を指定し、これは、ObjectDescriptorUpdateコマンド2000またはObjectDescriptorRemoveコマンド2040に見られるnumBytes値2020および2060について指定されたものと同一の形でエンコードされる。

【0092】

ES_Descrデータ構造1000の構造を、図23に示した。図57からわかるように、EsIdRefデータ構造2160は、1バイトのES_ID_RefTag 2170、numBytes値2180、および16ビットのエレメンタリ・ストリームID(ES_ID)値2190からなる。この場合に、numBytesの値は、必ず「2」であり、この値は、8ビット整数として指定される。

【0093】

本発明の動作を、全般的に図58に示す。本発明2200は、XMT-A文書2210および関連するメディア・データ・ファイル2220の内容に基づいて、MP4-4 Intermediaファイル2230を作成する。出力のMP4-4 Intermediaファイル2230を、「mp4バイナリ・ファイル」または「mp4ファイル」とも称する場合がある。入力 of XMT-A文書2210は、ISO/IEC 14496-1:2000 Amd. 2に記載のXMT-A仕様またはそのようなファイルのデータ構造表現の組に基づくテキスト・ファイルからなるものとすることができる。関連するメディア・データ・ファイル2220は、XMT-A文書2210に含まれるStream Source参照670によって指定されるオーディオ・データ、ビデオ・データ、およびイメージ・データを表す。メディア・データ・ファイル2220の個数は、0個以上とすることができる。

【0094】

本発明2200によって実行される論理動作は、(1)コンピュータ・システムで稼動するコンピュータ実施されるステップのシーケンスとして、または(2)コンピューティング・システム内の相互接続された機械モジュールとして、あるいはその両方として実施することができる。実施形態は、本発明を適用するシステムの性能要件に依存する選択の問題である。したがって、本明細書で説明する本発明の実施形態を構成する論理動作を、その代わりに動作、ステップ、またはモジュールと呼ぶ。

【0095】

さらに、本発明によって実行される動作を、コンピュータ可読媒体として実施されたコンピュータ可読プログラムとすることができる。制限ではなく例として、コンピュータ可読媒体に、コンピュータ記憶媒体および通信媒体が含まれる。コンピュータ可読媒体に、コンピュータ可読命令、データ構造、プログラム・モジュール、または他のデータなどの情報の保管の方法または技術で実施された、揮発性および不揮発性の、取り外し可能および取り外し不能の媒体が含まれる。コンピュータ記憶媒体に、RAM、ROM、EEPROM、フラッシュ・メモリ、または他のメモリ技術、CD-ROM、デジタル多用途ディ

スク（DVD）、または他の光ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージ、または他の磁気記憶デバイス、あるいは所望の情報の保管に使用でき、コンピュータによってアクセス可能な他のすべての媒体が含まれるが、これに制限はされない。通信媒体は、通常、搬送波または他のトランスポート機構などの変調されたデータ信号内でコンピュータ可読命令、データ構造、プログラム・モジュール、または他のデータを実施し、通信媒体には、すべての情報配信媒体が含まれる。用語「変調されたデータ信号」は、信号内で情報をエンコードする形で、その特性の1つまたは複数を設定または変更された信号を意味する。制限ではなく例として、通信媒体に、有線接続または直接配線接続などの有線媒体と、音響、RF、赤外線、および他の無線媒体などの無線媒体が含まれる。上記のいずれかの組合せも、コンピュータ可読媒体の範囲に含まなければならない。 10

【0096】

図58からわかるように、mp4ファイル2230を作成する処理は、2ステップで達成される。第1ステップでは、XMT-A-中間文書コンバータ2240が、入力XMT-A文書2210を解釈し、MPEG-4 Intermediaファイル2230を表す1つまたは複数の中間文書2245を作成する。本発明の1実施形態で、中間文書の対2250および2260が、中間文書コンバータ2240によって作成される。この中間文書は、mp4-file文書2250およびmp4-bifs文書2260からなる。第2ステップで、中間文書-mp4ファイル・コンバータ2270が、mp4-file文書2250、mp4-bifs文書2260、およびXMT-A文書2210に関連するメディア・データ・ファイル2220（存在する場合に）に基づいて出力mp4ファイル2230を生成する。 20

【0097】

この処理2200をこの2つのステップに分割する理由の1つは、出力mp4ファイル2230が、入力XMT-A文書2210およびメディア・データ・ファイル2220によって表されるものと同一の情報を表すが、出力mp4ファイル2230の編成および構造が、入力のXMT-A文書の編成および構造と大きく異なるからである。たとえば、mp4ファイルの構造は、Quicktime(R)メディア・データ・ファイルの構造に密接に関係するが、XMT-Aファイルの構造は、Quicktime(R)メディア・データ・ファイルの特性を全く有しない。XMT-A文書には、scene description stream(sdsm)およびobject descriptor stream(odsm)の記述が含まれるが、これらは、任意の順序で一緒に混合することができる。mp4ファイルにも、sdsmおよびodsmの記述が含まれるが、これらのそれぞれが、時間的に順序付けられたサンプルの別々のストリームとして表現される。 30

【0098】

mp4ファイル2230の構造および編成が、XMT-A文書2210の構造および編成と非常に異なるので、mp4ファイル2230を作成する処理を、XMT-A文書2210に基づいて、少なくとも2つのステップすなわち(a)再編成および(b)バイナリ・エンコーディングに分割することが有利である。第1ステップでは、XMT-A文書2210に含まれる情報を、mp4ファイルの構造および編成を反映した形に再編成する。第2ステップでは、結果の再編成された情報を、出力mp4ファイル2230に必要な順序でトラバースしながらこの情報のバイナリ・エンコーディングを実行することによって、出力mp4ファイル2230を作成する。この形で、第1ステップは、mp4ファイルのバイナリ・エンコーディングの要件に無関係に実行することができ、第2ステップは、XMT-A文書の構造に無関係に実行することができる。 40

【0099】

XMT-A文書2210に基づいてmp4ファイル2230を作成する処理をこの2ステップに分割するという目標を達成するために、(a)mp4ファイルの構造および編成、(b)mp4ファイルで表されるstream description stream 50

m (s d s m) の構造および編成、ならびに (c) m p 4 ファイル 2 2 3 0 で表される o b j e c t d e s c r i p t o r s t r e a m (o d s m) の構造および編成を表す新しい構造化文書を定義する必要がある。これは、3つの新しいタイプの構造化文書すなわち、m p 4 ファイルを表す構造化文書、s d s mを表す構造化文書、および o d s mを表す構造化文書を定義することによって達成することができる。もちろん、m p 4 ファイルおよび s d s mを表すのに必要な構造化文書は、比較的複雑であるが、o d s mを表すのに必要な構造化文書は、非常に単純である。その結果、o d s mの記述を、m p 4 ファイルを表すのに使用される構造化文書に組み込むことが便利である。その結果、本発明の1実施形態で、m p 4 ファイルおよび o d s mの構造化文書と s d s mの構造化文書という2つの新しい構造化文書が導入される。この2つの構造化文書は、m p 4 - f i l e 文書 2 2 5 0 および m p 4 - b i f s 文書 2 2 6 0 として識別される。この構造化文書を、集合的に中間文書と呼ぶ。

【0100】

2つのタイプの構造化文書の使用が、便利さの問題として選択されたことに留意されたい。同一の目的を、3タイプの構造化文書 (m p 4 ファイルだけ、o d s mだけ、および s d s mだけ) を定義することによって達成することができ、3タイプの情報のすべてを、単一の合成構造化文書に合併することができる。

【0101】

本発明の1実施形態で、再編成された情報を表現するために作成される構造化文書の特定のタイプは、「XML」(e X t e n s i b l e M a r k u p L a n g u a g e) 技術に基づく。これが有利なのは、下記の理由による。

(a) X M T - A 文書の定義が、XML技術に基づく

(b) XML技術は、構造化文書を表す標準化された手段を提供する

(c) XML技術に基づく構造化文書を扱う標準化されたソフトウェア・ツールが存在する

【0102】

したがって、本発明の1実施形態で、X M T - A ファイルに含まれる情報を再編成する処理が、XMLからXMLへの変換になる。さらに、XMLファイルを扱う標準化されたソフトウェアの存在によって、同一の機能を実行する新しい特殊化されたソフトウェアを開発する必要なしに、入力X M T - A 文書ならびに中間文書を管理することが可能になる。

【0103】

この実施形態は、中間文書を表すのにXML技術を使用することに基づくが、他のタイプの構造化文書を使用する、本発明の代替実施形態を作成することが可能である。

【0104】

次の材料で、(1) m p 4 - f i l e 文書 2 2 5 0 の構造、(2) m p 4 - b i f s 文書 2 2 6 0 の構造、(3) X M T - A - 中間文書コンバータ 2 2 4 0 の動作、および (4) 中間文書 - m p 4 ファイル・コンバータ 2 2 7 0 の動作を説明する。

【0105】

5. 0 m p 4 - f i l e 文書

図59からわかるように、m p 4 - f i l e 文書 2 3 0 0 の構造は、m p 4 バイナリ・ファイル 7 0 0 の構造に非常に似ている。m p 4 f i l e 文書 2 3 0 0 には、1つまたは複数のメディア・データ (m d a t) 要素 2 3 1 0 の組と、単一の m o o v 要素 2 3 2 0 が含まれる。m o o v 要素 2 3 2 0 に、m p 4 f i o d s (m p 4 f i l e i n i t i a l o b j e c t d e s c r i p t o r) 要素 2 3 3 0 と、1つまたは複数の t r a k 要素 2 3 5 0 が含まれる。m o o v 要素 2 3 2 0 に、任意選択のユーザ・データ (u d t a) 要素 2 3 4 0 を含めることもできる。u d t a 要素 2 3 4 0 は、著作権表示などの情報を含めるのに使用される。m v h d アトム 7 7 2 のプロパティは、m o o v 要素 2 3 2 0 の属性によって表される。

【0106】

図60からわかるように、mp4fiods要素2360は、objectDescriptorID属性2370を所有する。mp4fiods要素2360は、図60に示されていない他の複数の属性も所有する。これらの追加の属性に、boolean属性「includeInlineProfilesFlag」、integer属性「sceneProfileLevelIndication」、「ODProfileLevelIndication」、「audioProfileLevelIndication」、「visualProfileLevelIndication」、および「graphicsProfileLevelIndication」が含まれる。mp4fiods要素2360に、1つまたは複数のEsIdInc要素2380も含まれる。各EsIdInc要素2380は、関連するtrak要素2350のtrackID属性と一致するtrackID属性2390を所有する。 10

【0107】

図61からわかるように、各mdat要素2400に、sdsm要素2410、odsm要素2420、およびmediaFile要素2430のうちの1つまたは複数を含めることができる。これらの要素のそれぞれは、関連するtrak要素2350のtrackID属性と一致する独自の「trackID」属性を所有する。各mediaFile要素2430は、関連するメディア・データ（オーディオ・データ、ビジュアル・データなど）を含む外部バイナリ・ファイルのファイル名を指定する「name」属性を有する。各sdsm要素2410は、関連するmp4-bifs文書2260を表すXMLファイルの名前を指定する「xmlFile」属性を有する。1実施形態で、mp4-file文書またはmp4-bifs文書あるいはその両方を表すXMLファイルの作成が、診断目的に有用である場合があるが、そのようなファイルは、本発明の動作に必要ではない。 20

【0108】

図62および64からわかるように、各sdsm要素2440および各mediaFile要素2480に、1つまたは複数のチャンク要素2450および2490が含まれる。各チャンク要素2450および2490は、既知の場合にバイナリ・データの関連するブロックのバイト数を示す「size」属性を所有する。各チャンク要素2450および2490は、既知の場合に、バイナリsdsmデータ・ファイルまたはメディア・データ・ファイルの先頭と、そのバイナリsdsmデータ・ファイルまたはメディア・データ・ファイル内の現在のチャンクのデータの先頭との間のバイト数を示す「offset」属性も所有する。scene description stream (sdsm) を記述する追加情報が、mp4-bifs文書に含まれる。 30

【0109】

図63からわかるように、各odsm要素2460に、1つまたは複数のodsmChunk要素2470要素が含まれる。各odsmChunk要素2470は、既知の場合にobject descriptor streamの関連する部分のバイト数を示す「size」属性を所有する。各odsmChunk要素2470は、既知の場合に、関連するobject descriptor streamのバイナリ・データの先頭と、そのストリーム内の現在のチャンクのデータの先頭との間のバイト数を示す「offset」属性も所有する。 40

【0110】

図65からわかるように、各odsmChunk要素2500に、1つまたは複数のodsmSample要素2510が含まれる。図66からわかるように、各odsmSample要素2520に、1つまたは複数のodsm-command要素2530が含まれる。図67からわかるように、各odsm-command要素は、ObjectDescriptorUpdate要素2540またはObjectDescriptorRemove要素2570とすることができる。各ObjectDescriptorUpdate要素2540に、ObjectDescriptor要素2550が含まれ、ObjectDescriptorUpdate要素2540に含まれる各ObjectDescriptor要素2550に、Es 50

I d R e f 要素 2 5 6 0 が含まれる。

【0111】

各 o d s m S a m p l e 要素 2 5 1 0 および 2 5 2 0 は、o d s m S a m p l e 要素 2 5 1 0 および 2 5 2 0 に含まれるコマンドが実行される時を秒単位で指定する「t i m e」属性を所有する。各 O b j e c t D e s c r i p t o r 要素 2 5 5 0 および各 O b j e c t D e s c r R e m o v e 要素 2 5 7 0 は、数値のオブジェクト記述子 I D を指定する「O D I D」属性を所有する。各 E s I d R e f 要素 2 5 6 0 は、数値エレメンタリ・ストリーム I D を指定する「E s I d」属性を所有する。

【0112】

図 6 8 に示された t r a k 要素 2 3 5 0 の構造 2 6 0 0 は、m p 4 ファイル 7 0 0 内の t r a k アトム 7 9 0 および 9 0 0 の構造に非常に類似する。各 t r a k 要素 2 6 0 0 に、m d i a 要素 2 6 0 4 が含まれる。t r a k 要素 2 6 0 0 に、t r e f (t r a c k r e f e r e n c e) 要素 2 6 3 6 または e d t s (e d i t l i s t) 要素 2 6 4 4 あるいはその両方も含まれる。m p 4 ファイルの t k h d アトム 9 1 0 に類似する t k h d 要素はない。その代わりに、t k h d アトム 9 1 0 に含まれるプロパティが、t r a k 要素 2 6 0 0 の属性として表現される。

【0113】

m d i a 要素 2 6 0 4 に、h d l r 要素 2 6 0 8 および m i n f 要素 2 6 1 2 が含まれる。m d h d アトム 9 1 5 のプロパティは、m d i a 要素 2 6 0 4 の属性として表される。m i n f 要素 2 6 1 2 に、d i n f 要素 2 6 1 6、s t b l 要素 2 6 2 8、およびメディア・ヘッダ要素 2 6 3 2 が含まれる。メディア・ヘッダ要素（「* m h d」）2 6 3 2 は、関連するデータ・ストリームのデータの型に依存する複数の形式の 1 つを有することができる。s d s m または o d s m に関連する t r a k 要素内のメディア・ヘッダ要素 2 6 3 2 は、「n m h d」要素によって表される。オーディオ・ストリームに関連する t r a k 要素内のメディア・ヘッダ要素 2 6 3 2 は、「s m h d」要素によって表され、ビジュアル・ストリームに関連する t r a k 要素内のメディア・ヘッダ要素 2 6 3 2 は、「v m h d」要素によって表される。

【0114】

図 6 9 からわかるように、s t b l (s a m p l e t a b l e s) 要素 2 6 2 8 および 2 6 5 2 に、s t s c (s a m p l e - t o - c h u n k t a b l e) 要素 2 6 5 6、s t t s (t i m e - t o - s a m p l e t a b l e) 要素 2 6 6 0、s t c o (c h u n k o f f s e t t a b l e) 要素 2 6 6 4、s t s z (s a m p l e s i z e t a b l e) 要素 2 6 6 8、および s t s d (s a m p l e d e s c r i p t i o n t a b l e) 要素 2 6 7 6 が含まれる。s t b l 要素 2 6 5 2 に、ストリーム・タイプまたはメディア・タイプに応じて、s t s s (s y n c s a m p l e t a b l e) 要素 2 6 7 2 も含めることができる。s t s d 要素 2 6 7 6 に、図 6 9 で「m p 4 * 要素」2 6 8 0 と表された、複数のタイプの従属要素の 1 つを含めることができる。s d s m ストリームまたは o d s m ストリームに関連する t r a k 要素 2 6 0 0 に含まれる s t s d 要素 2 6 7 6 の場合に、s t s d 要素 2 6 8 0 に、「m p 4 s」要素が含まれる。オーディオ・ストリームに関連する t r a k 要素 2 6 0 0 に含まれる s t s d 要素 2 6 8 0 の場合に、s t s d 要素 2 6 8 0 に、「m p 4 a」要素が含まれる。ビジュアル・ストリームに関連する t r a k 要素 2 6 0 0 に含まれる s t s d 要素 2 6 8 0 の場合に、s t s d 要素 2 6 8 0 に、「m p 4 v」要素が含まれる。どの場合でも、「m p 4 *」要素 2 6 8 0 には、e s d s 要素 2 6 8 4 が含まれ、e s d s 要素 2 6 8 4 に、E S _ D e s c r 要素 2 6 8 8 が含まれる。

【0115】

図 7 0 からわかるように、E S _ D e s c r 要素 2 7 0 0 に、D e c o d e r C o n f i g D e s c r i p t o r 要素 2 7 1 0 および S L C o n f i g D e s c r i p t o r 要素 2 7 6 0 が含まれる。D e c o d e r C o n f i g D e s c r i p t o r 要素 2 7 1 0 に、B I F S _ D e c o d e r C o n f i g 要素 2 7 2 0、J P E G _ D e c o d e r C

on f i g 2 7 3 0、V i s u a l C o n f i g 2 7 4 0、またはA u d i o C o n f i g 2 7 5 0を含む複数のタイプのデコーダ固有情報要素の1つを含めることができる。さまざまなタイプのデコーダ固有情報要素のそれぞれは、バイナリD e c o d e r C o n f i g D e s c r i p t o r構造1032に含まれるD e c o d e r S p e c i f i c I n f oデータ構造1072の形式を表す。バイナリE S _ D e s c r構造1000、D e c o d e r C o n f i g D e s c r i p t o r構造1032、S L C o n f i g D e s c r i p t o r構造1088、およびD e c o d e r S p e c i f i c I n f o構造1076のプロパティは、m p 4 - f i l e文書2300の対応する要素2700、2710、2760、2720、2730、2740、および2750の属性によって表される。

10

【0116】

6. 0 m p 4 - b i f s 文書

図71からわかるように、m p 4 - b i f s 文書2800に、単一のb i f s C o n f i g要素2810と、それに続く1つまたは複数のc o m m a n d F r a m e要素2820のシーケンスが含まれる。図72からわかるように、各c o m m a n d F r a m e要素2830に、1つまたは複数のm p 4 b i f s b i f s C o m m a n d要素2840が含まれる。各c o m m a n d F r a m e要素2820および2830は、c o m m a n d F r a m e要素に含まれるコマンドが実行される時を秒単位で指定する属性「t i m e」を所有する。

【0117】

各m p 4 b i f s b i f s C o m m a n d要素2840は、11個の可能なM P E G - 4 B I F Sコマンドすなわち、I n s e r t N o d e、I n s e r t I n d e x e d V a l u e、I n s e r t R o u t e、D e l e t e N o d e、D e l e t e I n d e x e d V a l u e、D e l e t e R o u t e、R e p l a c e N o d e、R e p l a c e F i e l d、R e p l a c e I n d e x e d V a l u e、R e p l a c e R o u t e、およびR e p l a c e S c e n eのうちの1つを表す。図73からわかるように、m p 4 b i f s b i f s C o m m a n d要素2910に、1つまたは複数のm p 4 b i f s N o d e要素2920を含めることができる。11タイプのb i f s C o m m a n dのうちで、I n s e r t N o d e、I n s e r t I n d e x e d V a l u e、R e p l a c e N o d e、R e p l a c e F i e l d、R e p l a c e I n d e x e d V a l u e、およびR e p l a c e S c e n eに、従属m p 4 b i f s N o d e要素2920を含めることができる。

20

30

【0118】

図74からわかるように、R e p l a c e S c e n e b i f s C o m m a n d要素2930に、単一の従属m p 4 b i f s N o d e要素だけを含めることができ、これは、「T o p N o d e」要素2940でなければならない。T o p N o d e要素2940は、M P E G - 4 B I F Sノードの特定のサブセットのメンバに対応する。このサブセットは、M P E G - 4 S y s t e m s仕様で定義されている。さらに、R e p l a c e S c e n e b i f s C o m m a n d要素2930に、従属する「R o u t e s」要素2950も含めることができ、「R o u t e s」要素2950に、1つまたは複数の従属する「R o u t e」要素2960を含めることができる。m p 4 b i f s R o u t e要素2960は、属性「r o u t e I d」、「a r r i v a l N o d e I d」、「a r r i v a l F i e l d」、「d e p a r t u r e N o d e I d」、および「d e p a r t u r e F i e l d」を有する。

40

【0119】

可能な従属m p 4 b i f s N o d e要素のほかに、各タイプのm p 4 b i f s b i f s C o m m a n d要素は、次の属性値を所有する。

1. I n s e r t N o d e: 「p a r e n t I d」、「i n s e r t i o n P o s i t i o n」、および「p o s i t i o n」

2. I n s e r t I n d e x e d V a l u e: 「n o d e I d」、「i n F i e l d N

50

ame」、「insertionPosition」、「position」、および「value」

3. InsertRoute: 「RouteId」、「departureNode」、「departureField」、「arrivalNode」、および「arrivalField」

4. DeleteNode: 「nodeId」

5. DeleteIndexedValue: 「nodeId」、「inFieldName」、「deletionPosition」、および「position」

6. DeleteRoute: 「routeId」

7. ReplaceNode: 「parentId」

8. ReplaceField: 「nodeId」、「inFieldName」、および「value」

9. ReplaceIndexedValue: 「nodeId」、「inFieldName」、「insertionPosition」、「position」、および「value」

10. ReplaceRoute: 「routeId」、「departureNode」、「departureField」、「arrivalNode」、および「arrivalField」

11. ReplaceScene: 「USENAMES」(boolean値)

【0120】

bifsCommand要素InsertIndexedValue、ReplaceField、およびReplaceIndexedValueに関して、「inFieldName」属性によって指定されるプロパティ・フィールドが、「Node」のノード・データ型を有する(MPEG-4仕様に従って)場合に、この要素に、1つまたは複数の従属するmp4bifs Node要素2920が含まれ、「value」属性に、従属するNode要素のそれぞれに関連するノード名のリストが含まれる。

【0121】

mp4bifs Node要素2920は、多数のタイプのMPEG-4 BIFSノード・データ構造の1つを表す。100個以上の異なるタイプのBIFSノードが、MPEG-4システム仕様で定義されている。MPEG-4 BIFSノードの各タイプが、特定のNodeNameおよび1組のプロパティ・フィールドを有する。

【0122】

mp4bifs Node要素には、オリジナルNode要素および再利用されるNode要素という2つの基本的なタイプがある。図75からわかるように、mp4bifsオリジナルNode要素3000は、MPEG-4 Systems仕様で定義されたBIFSノードの1つのNodeNameプロパティに対応する「NodeName」によって指定される。

【0123】

mp4bifsオリジナルNode要素3000は、任意選択のNodeId属性3010を有することができる。NodeId属性3010の値が指定される場合に、Node要素3000は、「再利用可能Node」として分類される。NodeId属性3010の値は、指定される場合に、1から現在のシーンで定義された再利用可能Nodeの個数までの範囲の整数である。NodeId属性3010の値が指定され、関連するReplaceSceneコマンドの「USENAMES」属性の値が「真」である場合に、そのNode要素は、「name」属性3016も有する。

【0124】

NodeId属性3010およびname属性3016の他に、各オリジナルNode要素は、複数のプロパティ・フィールド属性3020を有する。各プロパティ・フィールド属性3020は、特定のNode要素のNodeNameによって識別されるノード・タイプのMPEG-4 Systems仕様で定義されたプロパティ・フィールドの1つ

10

20

30

40

50

に対応する。各プロパティ・フィールドは、boolean、integer、float など、定義されたフィールド・データ型を有する。可能なフィールド・データ型の組に、「SFNode」および「MFNode」が含まれる。特定のオリジナルNode要素のNodeNameが、フィールド・データ型「SFNode」および「MFNode」を有する1つまたは複数のプロパティ・フィールドを有するMPEG-4 BIFSノードに対応する場合に、そのNode要素は、1つまたは複数の従属するNode要素3030を所有することができる。そうである場合に、対応するプロパティ・フィールド属性の値は、そのプロパティ・フィールドに関連する各従属Node要素のNodeName スtringからなる。

【0125】

たとえば、NodeName「Group」を有する特定のmp4bifs Node要素が、「children」属性に関連する、NodeName「Transform2D」、「Valuator」、および「TimeSensor」を有する従属mp4bifs Node要素を所有する場合に、「children」属性の値は、「Transform2D Valuator TimeSensor」になる。

【0126】

Conditional BIFSノードの特別な例で、プロパティ・フィールドの1つが、「buffer」というプロパティ・フィールド名を有し、「buffer」プロパティ・フィールドのフィールド・データ型が、「command buffer」であり、「buffer」プロパティ・フィールドの値が、1つまたは複数のBIFSコマンドからなる。この場合に、対応するmp4bifs Node要素3040のNodeNameは、「Conditional」になる。Conditional Node要素3040のNodeId属性3050およびname属性3056の値は、他のmp4bifsオリジナルNode要素3000と同様に指定することができる。従属するNode要素3030の代わりに、Conditional Node要素は、1つまたは複数の従属するbifsCommand要素3070を有し、「buffer」属性の値は、従属するbifsCommand要素3070のコマンド名の順序付きリストからなる。

【0127】

たとえば、特定のConditional Node要素が、従属InsertRoute bifsCommand要素とそれに続く従属DeleteNode bifsCommand要素を所有する場合に、「buffer」属性の値は、「InsertRoute DeleteNode」になる。

【0128】

オリジナルNode要素の、従属するNode要素またはbifsCommand要素を所有する能力は、BIFSコマンドおよびNode要素の階層的集合に対して再帰的に繰り返すことができる。

【0129】

図77からわかるように、再利用されるNode要素3080は、「ReusedNode」というNodeNameを有する。ReusedNode要素3080は、従属する要素を有しない。ReusedNode要素3080は、「nodeRef」3090という名前の単一の属性を有する。nodeRef属性3090の値は、再利用可能なオリジナルNode要素3000および3040の一方のNodeId属性3010および3050の値と一致しなければならない。

【0130】

7.0 xmta-mp4コンバータ

上で述べたように、本発明の1実施形態は、XMT-A文書2210および0個以上のバイナリ・メディア・データ・ファイル2220のセットに基づいて、MPEG-4 Intermediateバイナリ・ファイル（「mp4ファイル」）2230を作成する。

【0131】

この処理は、次の2つの主要なステップからなる。

10

20

30

40

50

a. 中間文書 2 2 5 0 および 2 2 6 0 の対が X M T - A 文書 2 2 1 0 に基づいて作成される第 1 ステップ 2 2 4 0

b. M P E G - 4 I n t e r m e d i a バイナリ・ファイル 2 2 3 0 が、中間文書 2 2 5 0 および 2 2 6 0 と、X M T - A 文書 2 2 1 0 で指定されたバイナリ・メディア・データ・ファイル 2 2 2 0 に基づいて作成される第 2 ステップ 2 2 7 0

【0 1 3 2】

メディア・データ・ファイル 2 2 2 0 は、第 2 ステップでのみ使用される。第 1 ステップ 2 2 4 0 は、メディア・データ・ファイルの名前を使用することができるが、メディア・データ・ファイル自体は、第 1 ステップ 2 2 4 0 では使用されない。

【0 1 3 3】

この主要なステップを、図 5 8 に示し、下で詳細に説明する。

【0 1 3 4】

7. 1 X M T - A 文書に基づく中間文書の作成

中間文書 2 2 5 0 および 2 2 6 0 を作成する処理 2 2 4 0 を、図 7 8 に要約した。処理 2 2 4 0 を、特定の応用例の必要に合わせて、ハードウェア、ソフトウェア、またはこの 2 つの組合せで実施できることが企図されている。ハードウェア実施形態は、より高速に動作する傾向があり、ソフトウェア実施形態は、しばしば、作るのがより安価である。処理 2 2 4 0 によって実行される論理動作は、(1) コンピュータ・システムで稼動するコンピュータ実施されるステップのシーケンスとして、または(2) コンピューティング・システム内の相互接続された機械モジュールとして、あるいはその両方として実施することができる。実施形態は、本発明を適用するシステムの性能要件に依存する選択の問題である。したがって、本明細書で説明する本発明の実施形態を構成する論理動作を、その代わりに動作、ステップ、またはモジュールと呼ぶ。

【0 1 3 5】

7. 1. 1 X M T - A 文書、m p 4 f i l e 文書、および m p 4 b i f s 文書の作成
処理 2 2 4 0 は、動作 3 1 0 0 で開始される。X M T - A 文書 1 0 0 は、この文書を表す X M L ファイルを読み取り、解釈することによって作成することができる。標準 X M L 手段を使用して、そのようなファイルを読み取り、X M L ファイルに含まれるすべての情報を表す X M T - A 文書を作ることができる。これは、標準 X M L 動作であり、本発明に固有ではない。その代わりに、X M T - A ベースの M P E G - 4 オーサリング・ツールなどの他の手段から前に導出された X M T - A 文書を、次のステップを実施するソフトウェアまたは他の手段への引数として与えることができる。

【0 1 3 6】

空の m p 4 - f i l e 文書 2 3 0 0 および空の m p 4 b i f s 文書 2 8 0 0 を、標準 X M L 手段を使用して作成する。これらの文書のそれぞれに、可能なデフォルト属性以外の子および属性を有しないトップ・レベル要素が含まれる。値を、ストリング量「s d s m F i l e N a m e」に割り当てることができる。この値は、中間文書が外部テキスト・ファイルに保存される場合に限って使用される。入力 X M T - A 文書がある場合に、その名前から適切な値を導出することができる。そうでない場合に、値「m p 4 b i f s . x m l」を、量「s d s m F i l e N a m e」に割り当てることができる。動作 3 1 0 0 が完了した後に、制御フローは、動作 3 1 0 6 に移る。

【0 1 3 7】

7. 1. 2 m p 4 b i f s 文書の新しい「b i f s C o n f i g」要素の作成

動作 3 1 0 6 で、m p 4 b i f s 文書の新しい「b i f s C o n f i g」要素を作成する。標準 X M L 手段を使用して、空の「b i f s C o n f i g」要素 2 8 1 0 を作成し、これを m p 4 b i f s 文書 2 8 0 0 のトップ・レベル要素に挿入する。標準 X M L 手段を使用して、この「b i f s C o n f i g」要素の次の属性に値を割り当てる。「0」の値を、「n o d e I d B i t s」属性に割り当てる。「0」の値を、「r o u t e I d B i t s」属性に割り当てる。「0」の値を、「p r o t o I d B i t s」属性に割り当てる。「真」の値を、「c o m m a n d S t r e a m」属性に割り当てる。「真」の値を、「

10

20

30

40

50

pixelMetric」属性に割り当てる。「0」の値を、「pixelHeight」属性に割り当てる。「0」の値を、「pixelWidth」属性に割り当てる。「偽」の値を、「useBifsV2Config」属性に割り当てる。「偽」の値を、「use3DMeshCoding」属性に割り当てる。「偽」の値を、「usePredictiveMFField」属性に割り当てる。これらは、後にXMT-A文書から導出された値によって置換される、一時的な値にすぎない。動作3106が完了した後に、制御フローは、動作3110に移る。

【0138】

7.1.3 mp4file文書の新しい「moov」要素の作成

動作3110で、mp4file文書の新しい「moov」要素を作成する。標準XML手段を使用して、空の「moov」要素2320を作成し、mp4file文書2300のトップ・レベル要素に挿入する。「1」の値を、量「nextTrackID」および量「nextESID」に割り当てる。標準XML手段を使用して、値にこの「moov」要素の、次の属性を割り当てる。

【0139】

a. 「creationTime」および「modificationTime」：これらの属性の値は、符号なし整数として表された、1904年1月1日からの経過秒数を指定しなければならない。これらの値は、現在のクロック時刻によって決定されることが好ましい。現在クロック時刻が入手可能でない場合には、これらに任意の値をセットすることができる。ここで指定される実際の値は、単に情報を与えるものであり、結果のMP EG-4バイナリ・ファイルの処理に影響しない。

【0140】

b. 「timeScale」：この属性は、MP EG-4バイナリ・ファイル内で時間を測定するのに使用される、毎秒のクロック・ティック数を指定する。1000の値は、時間がミリ秒単位で指定されることを暗示する。

【0141】

c. 「duration」：値0が、この属性に割り当てられる。この値は、後で更新される。

【0142】

d. 「nextTrackID」：量「nextTrackID」の値が、この属性に割り当てられる。後に、各新しい「trak」要素が文書に追加され、量「nextTrackID」の値が増やされる時に、この属性の値が更新される。

【0143】

この時点で、任意選択の「udta」(user data)要素2340を追加することができる。これは、後続の主要なステップ2270で作成されるMP EG-4バイナリ・ファイルに著作権メッセージを挿入するのに使用することができる。作成者識別などの他の情報も、この時に指定することができる。動作3110が完了した後に、制御フローは動作3116に移る。

【0144】

7.1.4 XMT-A「Header」要素の処理

動作3116では、XMT-A Headerを処理する。このステップでは、「moov」要素2320内の「mp4fiods」要素2330および2360を作成する。「mdat」要素2310および「trak」要素2350が、scene description stream(sds)に関して作成される。オブジェクトが存在する場合に、もう1つの「mdat」要素2310およびもう1つの「trak」要素2350が、object descriptor stream(odsm)について作成される。

【0145】

標準XML手段を使用して、XMT-A文書100内の「Header」要素110を入手することができる。次に、図79に示されたステップを使用して、XMT-A「He

10

20

30

40

50

ader」要素を処理する。XMT-A「Header」処理サブ動作は、InitialObjectDescriptor処理動作3150から始まる。

【0146】

動作3150では、標準XML手段を使用して、XMT-A「Header」要素110に従属するXMT-A「InitialObjectDescriptor」要素130を入手することができる。「InitialObjectDescriptor」要素130は、「IODID:nnn」という形の値を有する「objectDescriptorID」という名前の属性を有することができ、ここで、サブストリング「nnn」は、正の整数を表す。その代わりに、この要素が、値「nnn」を有する「binaryID」という名前の属性を有する場合がある。どちらの場合でも、「nnn」によって表される値を、量「ODID」に割り当てる。この属性のどちらもが存在しない場合には、値「1」を、量「ODID」に割り当てる。動作3150が完了した後に、制御フローは、動作3160に移る。 10

【0147】

動作3160で、新しい「mp4fiods」要素2330および2360を作成し、mp4file文書2300の「moov」要素2320に挿入する。XMT-A文書100の「InitialObjectDescriptor」要素130から導出された量「ODID」の値を、「mp4fiods」要素2330および2360の「objectDescriptorID」属性2370に割り当てる。動作3160が完了した後に、制御フローは、動作3170に移る。 20

【0148】

動作3170では、標準XML手段を使用して、「InitialObjectDescriptor」要素130に従属する「Profiles」要素150を入手する。次に、「mp4fiods」要素2360の「includeInlineProfiles」、「sceneProfileLevelIndication」、「ODProfileLevelIndication」、「visualProfileLevelIndication」および「graphicsProfileLevelIndication」属性の値を、「Profiles」要素150の同一の名前の属性の値に基づいてセットする。

【0149】

「mp4fiods」要素2360の「includeInlineProfiles」属性の値は、「真」または「偽」でなければならない。「Profiles」要素150の「includeInlineProfiles」属性の値が、「真」または「偽」である場合に、同一の値を、「mp4fiods」要素2360の「includeInlineProfiles」属性に割り当てる。そうでない場合には、値「偽」を、「mp4fiods」要素2360の「includeInlineProfiles」属性に割り当てる。 30

【0150】

「mp4fiods」要素2360の5つのプロファイルおよびレベル表示属性の値は、-255から+255までの数値を表さなければならない。「Profiles」要素150の対応する属性は、同等の値を有する場合があり、あるいは、英数字ストリングによって指定される値を有する場合がある。「Profiles」要素150のプロファイルおよびレベル表示属性のいずれかが、ストリング値「none」を有する場合に、「mp4fiods」要素2360の同一の名前を有する属性に、数値「-1」または「255」を割り当てる。「Profiles」要素150のプロファイルおよびレベル表示属性のいずれかが、ストリング値「unspecified」を有する場合に、「mp4fiods」要素2360の同一の名前を有する属性に、数値「-2」または「254」を割り当てる。「Profiles」要素150のこれらの属性の他の英数字値が、MPEG-4 Systems仕様で定義され、これに含まれるテーブル内の数値に関係する。Profiles要素150のプロファイルおよびレベル表示属性のいずれかの値が、M 40 50

PEG-4 Systems仕様で定義された英数字プロファイルおよびレベル・ストリングの1つと一致する場合に、これらのテーブルで指定される対応する数値を、「mp4fiods」要素2360の対応する属性に割り当てる。「Profiles」要素150のプロファイルおよびレベル表示属性のいずれかの値が、PEG-4 Systems仕様に含まれるプロファイル値およびレベル値のテーブルのどの項目にも一致しない英数字ストリングからなる場合には、「mp4fiods」要素2360の対応する属性に、値「-2」または「254」を割り当てる。動作3170が完了した後に、制御フローは、動作3176に移る。

【0151】

動作3176で、標準XML手段を使用して、「InitialObjectDescriptor」要素130に従属する「Descr」要素160を入手する。手順「Descr要素の処理」3176は、このDescr要素160に従属するesDescr要素170を識別するために実行される。手順「esDescr要素の処理」3180は、このesDescr要素に従属するES_Descriptor要素180および190それぞれを識別するために実行される。手順「ES_Descriptorの処理」3186および3190は、esDescr要素170に従属するES_Descriptor要素ごとに実行される。

【0152】

図80からわかるように、手順「Descr要素の処理」3176は、インデックス「i」に値「0」を割り当てることによって開始される3200。この手順のインデックス「i」の値を、量「numDescrChildren」の値と比較する3210。量numDescrChildrenの値は、Descr要素160によって所有される従属要素の個数を示す。インデックス「i」の値が、numDescrChildrenの値と等しい場合に、手順「Descr要素の処理」3176が完了し3260、手順「XMT-Aヘッダの処理」3116が完了し、手順「XMT-A文書の処理」は、下で説明するパス1 XMT-A Body処理動作3120に進む。

【0153】

インデックス「i」の値が、numDescrChildrenの値と比といくくない場合に、標準XML手段を使用して、Descr要素160に従属するi番目の要素を入手し、結果の従属要素を「DescrChild」として識別する3220。DescrChildの要素名が「esDescr」である場合に、手順「esDescr要素の処理」を実行する3240。その後、インデックス「i」の値を1つ増分し3250、この「i」の値のnumDescrChildrenとの比較3210を繰り返す。各Descr要素が、単一の従属esDescr要素を作ると期待される。

【0154】

図81からわかるように、手順「esDescr要素の処理」3180は、インデックス「i」に値「0」を割り当てることによって開始される3300。このインデックス「i」は、手順「Descr要素の処理」3176内で定義される類似する量と別個である。この手順のインデックス「i」の値を、量「numEsDescrChildren」と比較する3310。量numEsDescrChildrenの値は、esDescr要素170が所有する従属要素の個数を示す。インデックス「i」の値が、numEsDescrChildrenの値と等しい場合には、手順「esDescr要素の処理」3180が完了し3360、手順「Descr要素の処理」3176は、その手順のインデックス「i」の値を増分すること3250によって継続される。

【0155】

インデックス「i」の値が、numDescrChildrenの値と等しくない場合には、標準XML手段を使用して、esDescr要素170に従属するi番目の要素を入手し、結果の従属要素を「esDescrChild」として識別する3320。esDescrChildの要素名が、「ES_Descriptor」である場合には、手順「ES_Descriptorの処理」を実行する3340。手順「ES_Descr

iptorの処理」は、図82に示されており、この手順の動作は、下の「ES_Descriptorの処理」で説明する。

【0156】

その後、インデックス「i」の値を1つ増分し3350、「i」の値と量numEsDescriptorChildrenの比較3310を繰り返す。

【0157】

InitialObjectDescriptor130に従属するDescriptor要素160に従属するesDescriptor要素170のそれぞれが、1つまたは2つのES_Descriptor要素180および190(scene description stream(sdsm)に関するES_Descriptor要素180および多分object descriptor stream(odsm)に関するES_Descriptor要素190)を作ると期待される。odsmのES_Descriptor要素190は、XMT-A文書100にオーディオ・オブジェクト、ビジュアル・オブジェクト、または他のオブジェクトが含まれる場合に、必ず期待される。各ES_Descriptor要素180および190が、手順「ES_Descriptorの処理」を使用して処理される。この手順は、下で説明する。

【0158】

7.1.5 XMT-A Body要素の処理 (パス1)

動作3120で、XMT-A文書100のBody要素120で定義された、すべてのメディア・オブジェクト、再利用可能BIFSノード、および再利用可能Routeを列挙するテーブルの組を作る。これらのテーブルは、メディア・オブジェクトの個数、BIFSノードの個数、およびRouteの個数を判定するのに使用される。これらのテーブルは、メディア・オブジェクトのあるプロパティを判定し、メディア・オブジェクト、BIFSノード、およびRouteへの参照を解決するのにも使用される。

【0159】

各メディア・オブジェクトは、ObjectDescriptor要素240によって定義される。各ObjectDescriptor要素240は、ObjectDescriptorUpdateコマンド要素220に含まれ、このコマンド要素は、XMT-A文書100のBody要素120内の「par」要素140および200に含まれる。メディア・オブジェクトのプロパティに、ObjectDescriptorID(ODID)130、オブジェクト開始時刻、オブジェクト終了時刻、およびオブジェクト持続時間が含まれる。ObjectDescriptorIDは、「ObjectDescriptor」要素240の「ObjectDescriptorID」属性として指定される英数字ストリングである。

【0160】

オブジェクト開始時刻は、それを囲む「par」要素200の「begin」属性によって決定される。

【0161】

オブジェクト終了時刻は、ObjectDescriptorRemoveコマンド要素250を囲む「par」要素200の「begin」属性によって決定される。ObjectDescriptorRemoveコマンド要素250で指定されるObjectDescriptorID(ODID)属性の値は、対応するObjectDescriptor要素240で指定されるObjectDescriptorID属性の値と一致しなければならない。オブジェクト持続時間は、オブジェクト終了時刻とオブジェクト開始時刻の間の差である。

【0162】

ObjectDescriptorIDストリングならびに関連する開始時刻および終了時刻の値は、図88に示されたObjectテーブルに保管される。このテーブルは、5つの列、ObjectDescriptorID 3910、OdId 3920、startTime 3930、stopTime 3940、およびEsId 3950

を有する。各列の個々の項目は、このテーブルの各行を識別する「position」値 3900 によって示される。

【0163】

再利用可能な B I F S ノードは、「DEF」属性の値を指定する X M T - A B I F S ノード要素によって定義される。この属性の値は、英数字ストリングである。再利用可能 B I F S ノード DEF ストリングの値は、図 89 に示されたテーブルに保管される。このテーブルは、1 つの列、NodeString 3966 を有する。この列の個々の項目は、このテーブルの各行を識別する「position」値 3960 によって示される。

【0164】

再利用可能な Route は、「DEF」属性の値を指定する X M T - A Route 要素によって定義される。この属性の値は、英数字ストリングである。再利用可能 Route DEF ストリングの値は、図 90 に示されたテーブルに保管される。このテーブルは、1 つの列、RouteString 3976 を有する。この列の個々の項目は、このテーブルの各行を識別する「position」値 3970 によって示される。

【0165】

第 4 のテーブルに、ReplaceScene コマンドの時刻値が記録される。このテーブルは、1 つの列、ReplaceSceneTime 3986 を有する。この列の個々の項目は、このテーブルの各行を識別する「position」値 3980 によって示される。

【0166】

これらのテーブルは、図 93 に示された X M T - A 「Body」要素内の従属要素をトラバースすることによって構築される。この手順は、値「0」をインデックス「i」に割り当てることによって、動作 4000 で開始される。動作 4000 が完了した後に、制御は、動作 4010 に移る。

【0167】

動作 4010 では、インデックス「i」の値を、量「numBodyChildren」の値と比較する。量 numBodyChildren の値は、X M T - A Body 要素 120 によって所有される従属要素の個数を示す。

【0168】

動作 4010 が完了した後に、インデックス「i」の値が numBodyChildren の値と等しい場合に、この手順が完了し 4060、処理は、odsm の edit list 作成に関して継続される。

【0169】

インデックス「i」の値が numBodyChildren の値と等しくない場合には、制御は、動作 4020 に移り、ここで、標準 XML 手段を使用して、Body 要素 120 に従属する i 番目の要素を入手し、結果の従属要素を「bodyChild」として識別する。要素 bodyChild の要素名は、ストリング量「childName」によって識別される。

【0170】

動作 4020 が完了した後に、制御は、動作 4030 に移り、ここで、ストリング量 childName の値をストリング「par」と比較する。childName の値が「par」である場合に、処理動作 4040 で、値「0」を量「parTime」に割り当て、親要素として現在の bodyChild 要素を使用して、手順「X M T - A par 要素の処理（パス 1）」4040 を実行する。手順「X M T - A par 要素の処理（パス 1）」は、下で説明する。

【0171】

動作 4040 が完了した後に、制御は、動作 4050 に移り、ここで、インデックス「i」の値を後に 1 つ増分し、「i」の値の numBodyChildren との比較 4010 を繰り返す。

【0172】

7. 1. 5. 1 XMT-A par要素の処理 (パス1)

XMT-A「par」要素140および200は、XMT-A Body要素120または別のXMT-A「par」要素200に従属する可能性がある。各XMT-A「par」要素200は、図94に示されているように処理される。

【0173】

動作4100で、現在の「par」要素200の「begin」属性の値を、量「parTime」4040および4136の現在の値に加算し、その結果を量「time」に割り当てる。

【0174】

動作4106で、値「0」をインデックス「i」に割り当てる。このインデックスは、10
他の手順で使用される類似するインデックス値と別個である。

【0175】

動作4110で、インデックス「i」の値を、量「numParChildren」の値と比較する。量numParChildrenの値は、現在の「親」要素によって所有される従属要素の個数を示す。

【0176】

インデックス「i」の値がnumParChildrenの値と等しい場合に、動作4116で、この手順が完了し、処理は、この「親」要素がBody要素120に従属する場合には動作4050、またはこの「親」要素が別の「par」要素200に従属する場合には動作4126で継続される。20

【0177】

動作4120で、インデックス「i」の値がnumParChildrenの値と等しくない場合に、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入手し、その結果の従属要素を、「parChild」要素として識別する。parChild要素の要素名は、ストリング量「childName」によって識別される。

【0178】

動作4130で、ストリング量childNameの値をストリング「par」と比較する。

【0179】

childNameの値が「par」である場合に、動作4136で、量「time」30
の値を量「parTime」に割り当て、現在のparChild要素を「親」要素として使用して、手順「XMT-A par要素の処理 (パス1)」を再帰的に実行する。この手順が完了した後に、処理は、下で説明するステップ4126で継続される。

【0180】

動作4140で、ストリング量childNameの値を、ストリング「Delete」と比較する。childNameの値が「Delete」である場合には、何も行われず、処理は動作4126で継続される。

【0181】

動作4150では、ストリング量childNameの値を、ストリング「Insert」と比較する。40

【0182】

動作4160では、ストリング量childNameの値を、ストリング「Replace」と比較する。

【0183】

childNameの値が「Insert」または「Replace」である場合に、動作4166で、現在のparChild要素を「親」要素として使用して、手順「XMT-A コマンド要素の処理 (パス1)」を実行する。この手順は、下で説明する。この手順が完了した後に、処理は、動作4126で継続される。

【0184】

動作4170で、ストリング量childNameの値を、ストリング「Object」50

DescriptorUpdate」と比較する。

【0185】

childNameの値が「ObjectDescriptorUpdate」である場合に、動作4176で、現在のparChild要素を「親」要素として使用して、手順「ODUpdate cmd-1の処理」を実行する。この手順は、下で説明する。この手順が完了した後に、処理は、動作4126で継続される。

【0186】

動作4180で、ストリング量childNameの値を、ストリング「ObjectDescriptorRemove」と比較する。

【0187】

childNameの値が「ObjectDescriptorRemove」である場合に、動作4186で、現在のparChild要素を「親」要素として使用して、手順「ODRemove cmdの処理」を実行する。この手順は、下で説明する。

【0188】

動作4126で、インデックス「i」の値を、その後、「1」だけ増分し、動作4110での「i」の値のnumParChildrenの値との比較を繰り返す。

【0189】

7. 1. 5. 2 XMT-Aコマンド要素の処理 (Pass 1)

手順「XMT-Aコマンド要素の処理 (パス1)」は、手順「XMT-A par要素の処理 (パス1)」の一部4166または再帰的に手順「BIFSコマンド要素の処理 (パス1)」の一部4270のいずれかとして実行することができる。図95に示されているように、この手順は、値「0」をインデックス「i」(他の手順で使用される類似するインデックス値と別個)に割り当てることによって、動作4200で開始される。

【0190】

動作4206で、インデックス「i」の値を、量「numCmdChildren」の値と比較する。量numCmdChildrenの値は、現在の「親」要素200によって所有される従属要素の個数を示す。

【0191】

インデックス「i」の値がnumCmdChildrenの値と等しい場合に、動作4290で、この手順が完了し、処理は、「親」要素がBody要素120に従属する場合には動作4050、「親」要素が別の「par」要素200に従属する場合には動作4126で継続される。

【0192】

インデックス「i」の値がnumCmdChildrenの値と等しくない場合に、動作4210で、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入手し、結果の従属要素を「cmdChild」要素として識別する。cmdChild要素の要素名は、ストリング量「childName」によって識別される。

【0193】

動作4216で、ストリング量childNameの値を、ストリング「ROUTE」と比較する。

【0194】

childNameの値が「ROUTE」である場合に、動作4220で、標準XML手段を使用して、cmdChild要素の「DEF」属性の値を入手する。「DEF」属性が存在しない場合には、何も行わず、処理は動作4280で継続される。

【0195】

cmdChild要素の「DEF」属性の値が指定されている場合に、動作4226で、「DEF」属性の値をストリング量「idString」に割り当てる。次に、idStringの値を、RouteIDテーブル3976の現在の項目のそれぞれと比較する。idStringの値が、このテーブルの現在の項目のいずれかと一致する場合に、処理は動作4280で継続される。

10

20

30

40

50

【0196】

`idString`の値が`RouteID`テーブル3976の現在の項目のどれとも一致しない場合に、動作4230で、このテーブルに新しい項目を作成し、`idString`の現在の値をその新しい項目に割り当てる。その後、処理は動作4280で継続される。

【0197】

動作4240で、ストリング量`childName`の値を、ストリング「`Scene`」と比較する。

【0198】

`childName`の値が「`Scene`」である場合に、動作4246で、`ReplaceSceneTime`テーブル3986に新しい項目を作成し、量「`time`」の現在の値を、その新しい項目に割り当てる。その後、処理は動作4270で継続される。 10

【0199】

動作4250で、標準XML手段を使用して、`cmdChild`要素の「`DEF`」属性の値を入手する。「`DEF`」属性が存在しない場合には、処理は動作4270で継続される。

【0200】

`cmdChild`要素の「`DEF`」属性の値が指定されている場合に、動作4256で、「`DEF`」属性の値を、ストリング量「`idString`」に割り当てる。次に、`idString`の値を、`NodeID`テーブル3966の現在の項目のそれぞれと比較する。`idString`の値が、このテーブルの現在の項目のいずれかと一致する場合に、処理は動作4270で継続される。 20

【0201】

動作4260で、`idString`の値が、`NodeID`テーブル3966の現在の項目のどれとも一致しない場合に、このテーブルに新しい項目を作成し、`idString`の現在の値を、その新しい項目に割り当てる。

【0202】

動作4270で、現在の`cmdChild`要素を親要素として使用して、現在の手順（`BIFS`コマンド要素の処理（パス1））を再帰的に実行する。

【0203】

動作4280で、インデックス「`i`」の値を、その後、「1」だけ増分し、動作4206での「`i`」値と`numCmdChildren`の値との比較を繰り返す。 30

【0204】

7. 1. 5. 3 「ODUpdate command-1の処理」手順

標準XML手段を使用して、（親）`ObjectDescriptorUpdate`コマンド要素220に従属する「`OD`」要素230を入手する。次に、標準XML手段を使用して、その「`OD`」要素230に従属する`ObjectDescriptor`要素240を入手する。`ObjectDescriptor`要素240の「`objectDescriptorID`」属性（図4では「`ODID`」と省略した）の値を、オブジェクト・テーブル（図88）内の`ObjectDescriptorID`列3910の項目と比較する。一致が見つかる場合に、量「`time`」の現在値を、`startTime`列3930の対応する項目に割り当てる。 40

【0205】

「`objectDescriptorID`」属性の値が、オブジェクト・テーブルの`ObjectDescriptorID`列3910の現在のどの項目とも一致しない場合に、オブジェクト・テーブルに新しい項目を追加し、「`objectDescriptorID`」属性の値を、`ObjectDescriptorID`列3910の新しい項目に割り当て、量「`time`」の現在値を、`startTime`列3930の対応する項目に置き、値「-1.0」を、`stopTime`列3940の対応する項目に置く。オブジェクト・テーブルの項目数を、`ODID`列3920の対応する項目に割り当てる。手順「`XM T-A Body`要素の処理（パス2）」の一部として、`ESID`列3950の対応する 50

項目に、値が割り当てられる。

【0206】

7.1.5.4 「ODRemove cmndの処理」手順

(親)ObjectDescriptorRemoveコマンド要素250の「objectDescriptorID」属性(図4では「ODID」と省略した)の値を、オブジェクト・テーブルのObjectDescriptorID列3910の項目と比較する。一致が見つかる場合に、量「time」の現在値を、stopTime列3940の対応する項目に割り当てる。

【0207】

「objectDescriptorID」属性の値が、オブジェクト・テーブルのObjectDescriptorID列3910の現在のどの項目とも一致しない場合に、新しい項目をオブジェクト・テーブルに追加し、「objectDescriptorID」属性の値を、ObjectDescriptorID列3910の新しい項目に割り当て、量「time」の現在値を、stopTime列3940の対応する項目に置き、値「-1.0」を、startTime列3930の対応する項目に置く。オブジェクト・テーブルの項目数を、OdId列3920の対応する項目に割り当てる。手順「XM T-A Body要素の処理(パス2)」の一部として、EsId列3950の対応する項目に、値が割り当てられる。

【0208】

7.1.6 odsmのedit list作成

XMT-A Body要素に対する第1パスの後に、オブジェクト・テーブル(図88)のstartTime項目3930の値を比較して、最小のstartTime項目(startTimeMin)を見つける。stopTime項目3940の対応する値を比較して、最大のstopTime項目(stopTimeMax)を判定する。stopTimeMaxの値とstartTimeMinの値の間の差を、量「duration」に割り当てる。

【0209】

最小のstartTime値が0を超える場合に、edit list(「edts」)要素2644を、odsm(object descriptor stream)に関連するtrak要素2600に挿入する。これは、次のように達成される。

【0210】

1. 標準XML手段を使用して、mp4-file文書2300内のmoov要素2320を入手する。

【0211】

2. 標準XML手段を使用して、moov要素2320内の各trak要素2350を入手する。

【0212】

3. 各trak要素2350のtrackID属性の値を、量trackIdForOdsmの値と比較する。この値は、odsmのtrak要素2350が作成された時に割り当てられたものである。

【0213】

4. 特定のtrak要素2600のtrackID属性の値が、trackIdForOdsmの値と一致する場合に、下記のステップを実行する。

【0214】

5. 標準XML手段を使用して、新しいedts要素2644を作成し、これを選択されたtrak要素2600に挿入する。

【0215】

6. 標準XML手段を使用して、新しいelst要素(2648)を作成し、これを新しいedts要素2644に挿入する。

【0216】

10

20

30

40

50

7. 標準XML手段を使用して、新しいsegment要素を作成し、これを新しいelement要素2648に挿入する。

【0217】

値「-1」を、この「segment」要素の「startTime」属性に割り当てる。値「1.0」を、この「segment」要素の「rate」属性に割り当てる。「moov」要素2320の「timeScale」属性とstartTimeMinの値の積を、この新しい「segment」要素の「duration」属性に割り当てる。

【0218】

8. 標準XML手段を使用して、第2の新しいsegment要素を作成し、これを新しいelement要素2648に挿入する。

10

【0219】

値「0」を、第2「segment」要素の「startTime」属性に割り当てる。値「1.0」を、第2「segment」要素の「rate」属性に割り当てる。量「duration」の値と「moov」要素2320の「timeScale」属性の値の積を、第2「segment」要素の「duration」属性に割り当てる。

【0220】

7. 1. 7 XMT-A Body要素の処理 (パス2)

このステップでは、図93に示されているように、XMT-A「Body」要素の従属要素をもう一度トラバースする。このトラバースで見つかった各従属「par」要素を、図96に示された手順「XMT-A par要素の処理 (パス2)」を使用して処理する4040。この手順が完了した後に、処理は、ステップ8「mp4bifs文書へのコマンド・フレームの挿入」で継続される。

20

【0221】

7. 1. 7. 1 XMT-A par要素の処理 (パス2)

動作4300で、現在の「親」要素200の「begin」属性の値を、量「parTime」4040の現在値に加算し、その結果を量「time」に割り当てる。

【0222】

動作4306で、値「0」をインデックス「i」に割り当てる。このインデックスは、他の手順で使用される類似するインデックス値と別個である。

【0223】

動作4310で、インデックス「i」の値を、量「numParChildren」の値と比較する。量numParChildrenの値は、現在の「親」要素によって所有される従属要素の個数を示す。

30

【0224】

インデックス「i」の値がnumParChildrenの値と等しい場合に、動作4316で、この手順が完了し、処理は、「親」要素がBody要素120に従属する場合には動作4050、「親」要素が別の「par」要素200に従属する場合には動作4336で継続される。

【0225】

インデックス「i」の値がnumParChildrenの値と等しくない場合に、動作4320で、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入手し、結果の従属要素を「parChild」要素として識別する。parChild要素の要素名は、ストリング量「childName」によって識別される。

40

【0226】

動作4330で、標準XML手段を使用して、新しいcommandFrame要素2820および2830を作成する。

【0227】

動作4340で、ストリング量childNameの値を、ストリング「par」と比較する。

【0228】

50

`childName`の値が「`par`」である場合に、動作4346で、量「`time`」の値を量「`parTime`」に割り当て、現在の`parChild`要素を「親」要素として使用して、手順「`XMT-A par`要素の処理（パス2）」を再帰的に実行する。この手順が完了した後に、処理は、下で説明するステップ4336で継続される。

【0229】

動作4350で、ストリング量`childName`の値をストリング「`ObjectDescriptorUpdate`」と比較する。

【0230】

`childName`の値が「`ObjectDescriptorUpdate`」である場合に、動作4356で、現在の`parChild`要素を「親」要素として使用して、手順「`ODUpdate command-2`の処理」を実行する。この手順は、下で説明する。この手順が完了した時に、処理は4126で継続される。 10

【0231】

動作4360で、ストリング量`childName`の値を、ストリング「`Insert`」と比較する。

【0232】

`childName`の値が「`Insert`」である場合に、動作4366で、現在の`parChild`要素を「親」要素として使用して、手順「`Insert`コマンドの処理」を実行する。この手順は、下で説明する。この手順が完了した時に、処理は、下で説明するステップ4336で継続される 20

【0233】

動作4370で、ストリング量`childName`の値を、ストリング「`Delete`」と比較する。

【0234】

`childName`の値が「`Delete`」である場合に、動作4376で、現在の`parChild`要素を「親」要素として使用して、手順「`Delete`コマンドの処理」を実行する。この手順は、下で説明する。この手順が完了した時に、処理は、下で説明するステップ4336で継続される

【0235】

動作4380で、ストリング量`childName`の値を、ストリング「`Replace`」と比較する。 30

【0236】

`childName`の値が「`Replace`」である場合に、動作4386で、現在の`parChild`要素を「親」要素として使用して、手順「`Replace`コマンドの処理」を実行する。この手順は、下で説明する。

【0237】

動作4336で、4330で作成された現在の`commandFrame`要素2830を、`commandFrame`要素の一時的順序付きリストに挿入する。これは、各新しい`commandFrame`要素を、新しい`commandFrame`要素の`time`属性の値より大きい値を有する`time`属性を有する、このリストの最初のメンバの前の点でこのリストに挿入し、あるいは、このリストの現在のメンバのどれかが新しい`commandFrame`要素の`time`属性の値より大きい値を有する`time`属性を有しない場合に、このリストの末尾に挿入することによって達成される。動作4336を、動作4326の直前（図96に示されているように）ではなく、動作4330の直後に実行することも可能である。どちらの場合でも、動作4330および4336は、コマンドを含まない空の`commandFrame`要素および他の`commandFrame`と同一の`time`属性を有する複数の`commandFrame`要素を作成することができる。動作3136で、空の`commandFrame`要素が除去され、同一の`time`属性を有する複数のコマンド・フレームが、組み合わせられる。 40

【0238】

動作4326で、インデックス「i」の値を、その後、「1」だけ増分し、「i」の値のnumParChildrenの値との比較4310を繰り返す。

【0239】

7.1.7.2 ODUUpdate command-2の処理

標準XML手段を使用して、(親)ObjectDescriptorUpdateコマンド要素220に従属する「OD」要素230を入手する。標準XML手段を使用して、「OD」要素230に従属するObjectDescriptor要素240を入手する。標準XML手段を使用して、「ObjectDescriptor」要素600に従属する「Descr」要素610を入手する。「Descr」要素610を、図80に示されているように処理して、従属「esDescr」要素620を入手する。この「esDescr」要素620を、図81に示されているように処理して、従属「ES_Descriptor」要素630を入手する。下で説明する手順「ES_Descriptorの処理」を、この形で入手された「ES_Descriptor」要素630ごとに実行する。 10

【0240】

7.1.7.3 Insertコマンドの処理

「Insert」コマンド要素を処理するのに使用されるステップを、図97に示す。この手順の親要素は、XMT-A Insertコマンド要素300である。このコマンド要素は、XMT-A「par」要素200またはConditional ノード要素3040の「buffer」属性要素410に従属する場合がある。Insertコマンド要素が、XMT-A「par」要素200に従属する場合に、mp4bifs「ターゲット」要素は、4330で作成されたcommandFrame要素2820および2830である。Insertコマンド要素が、Conditional ノード要素3040の「buffer」属性要素に従属する場合に、mp4-bifs Conditional node要素は、mp4-bifs Conditional node要素である。 20

【0241】

まず、値「偽」を、2つのboolean値、bInsertNodeおよびbInsertValueに割り当てる。

【0242】

動作4400で、親要素の「atNode」属性の値を、量「NodeId」に割り当てる。「atNode」属性の値が指定されていない場合には、この手順は、動作4446で継続される。 30

【0243】

「atNode」属性の値が指定されている場合に、動作4406で、親要素の「atField」属性の値を、量「FieldName」に割り当てる。

【0244】

「atField」属性の値が指定されている場合に、動作4416で、量「FieldName」の値をストリング「children」と比較する。

【0245】

「atField」属性の値が指定されていない場合、または量「FieldName」の値が「children」である場合に、動作4410で、値「真」をboolean量「bInsertNode」に割り当て、この手順は、動作4446で継続される。 40

【0246】

「atField」属性の値が指定されており、量「FieldName」の値が「children」でない場合に、動作4420で、標準XML手段を使用して、新しいmp4-bifs InsertIndexedValueコマンド要素(「newCommand」)を作成する。次に、標準XML手段を使用して、newCommand要素を現在のmp4-bifsターゲット要素に付加する。

【0247】

動作4426で、親要素の「value」属性の値を、量「value」に割り当てる 50

。

【0248】

XMT-A BIFSコマンド要素の「value」属性の値が指定されている場合に、動作4430で、新しいmp4bifs newCommand要素の「value」属性に値を割り当てる。ほとんどの場合に、新しいmp4bifs newCommand要素の「value」属性に割り当てられる値は、XMT-A BIFSコマンド要素の「value」属性の値と等しい。下で識別されるいくつかの場合（データ・フォーマット変換）に、新しいmp4bifs newCommand要素の「value」属性に割り当てられる値は、XMT-A BIFSコマンド要素の「value」属性の値から導出される。

10

【0249】

この場合に、この手順は完了し、処理は、動作4336またはXMT-A Conditional node要素の処理4890で継続される。

【0250】

「value」属性の値が指定されていない場合に、動作4436で、値「真」を、boolean量「bInsertValue」に割り当てる。現在の親要素に直接に従属するすべての要素の要素名のリストからなるストリング量「childNames」を作成する。

【0251】

動作4440で、ストリング量「childNames」の値を、新しいmp4-bifs要素「newCommand」の「value」属性に割り当てる。

20

【0252】

動作4446で、「0」の値を、他所で定義される他のインデックス値と別個であるインデックス値「i」に割り当てる。

【0253】

動作4450で、インデックス「i」の値を、量「numCmdChildren」の値と比較する。量numCmdChildrenの値は、現在の「親」要素によって所有される従属要素の個数を示す。

【0254】

インデックス「i」の値が、numCmdChildrenの値と等しい場合に、動作4456で、この手順が完了し、処理は、動作4336またはXMT-A Conditional node要素の処理4890で継続される。

30

【0255】

インデックス「i」の値が、numCmdChildrenの値と等しくない場合に、動作4460で、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入力し、結果の従属要素を「insertChild」要素として識別する。insertChild要素の要素名は、ストリング量「childName」によって識別される。

【0256】

動作4470で、ストリング量childNameの値をストリング「ROUTE」と比較する。

40

【0257】

ストリング量childNameの値が「ROUTE」である場合に、動作4476で、手順「InsertRouteコマンドの処理」を実行する。次に、標準XML手段を使用して、結果のnewCommand要素を現在のmp4-bifsターゲット要素に付加する。この手順は、その後、動作4466で継続される。

【0258】

ストリング量childNameの値が「ROUTE」でない場合に、動作4480で、boolean量bInsertNodeの値を値「真」と比較する。

【0259】

boolean量bInsertNodeの値が「真」である場合に、動作4486で

50

、手順「InsertNodeコマンドの作成」を実行する。その後、標準XML手段を使用して、結果のnewCommand要素を現在のmp4-bifsターゲット要素に付加する。この手順は、その後、動作4496で継続される。

【0260】

boolean量bInsertNodeの値が「真」でない場合に、動作4490で、boolean量bInsertValueの値を値「真」と比較する。

【0261】

boolean量bInsertValueの値が「真」である場合に、またはboolean量bInsertNodeの値が「真」である場合に、動作4496で、現在のinsertChild要素を親要素として使用して、手順「XMT-A BIFSノードの処理」を実行する。その後、標準XML手段を使用して、結果のmp4-bifsノード要素をnewCommand要素に付加する。処理は、動作4466で継続される。

【0262】

boolean量bInsertValueの値が「真」でない場合に、動作4498で、XMT-A文書が有効でなく、エラーを報告する。現在のinsertChild要素を親要素として使用して、手順「XMT-A BIFSノードの処理」を、実行する。その後、標準XML手段を使用して、結果のmp4-bifsノード要素を現在のmp4-bifsターゲット要素に付加する。処理は、動作4466で継続される。

【0263】

動作4466で、インデックス「i」の値を「1」だけ増分し、numCmdChildrenとの比較4450を繰り返す。

【0264】

7.1.7.4 「InsertRouteコマンドの作成」手順

標準XML手段を使用して、新しいmp4-bifs InsertRouteコマンド要素（「newCommand」）を作成する。

【0265】

XMT-A親要素の「fromNode」属性の値を、BIFS NodeIdテーブル（図89）の項目3966と比較する。一致する項目の「position」3960の値を、整数量fromNodeIdに割り当て、結果を「1」だけ増分する。その後、結果を、newCommand要素の「departureNode」属性に割り当てる。

【0266】

XMT-A親要素の「fromField」属性の値を、newCommand要素の「departureFieldName」属性に割り当てる。

【0267】

XMT-A親要素の「toNode」属性の値を、BIFS NodeIdテーブル（図89）の項目3966と比較する。一致する項目の「position」3960の値を、整数量fromNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「arrivalNode」属性に割り当てる。

【0268】

XMT-A親要素の「toField」属性の値を、newCommand要素の「arrivalFieldName」属性に割り当てる。

【0269】

XMT-A親要素の「DEF」属性の値が指定されている場合に、この属性の値を、BIFS RouteIdテーブル（図90）の項目3976と比較する。一致する項目の「position」3970の値を、整数量routeIdに割り当て、結果を「1」だけ増分する。その結果をnewCommand要素の「routeId」属性に割り当てる。boolean量bUseNamesの値が真である場合に、「DEFS」属性の値を、newCommand要素の「name」属性に割り当てる。bUseNamesの値は、「シーン置換」コマンドを処理している間に確立される。

10

20

30

40

50

【0270】

7. 1. 7. 5 「InsertNodeコマンドの作成」手順

標準XML手段を使用して、新しいmp4-bifs InsertNodeコマンド要素（「newCommand」）を作成する。

【0271】

量「NodeId」の値を、BIFS NodeIdテーブル（図89）の項目3966と比較する。一致する項目の「position」3960の値を、整数量atNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「parentId」属性に割り当てる。

【0272】

10

XMT-A親要素の「position」属性の値が「BEGIN」である場合に、値「2」をnewCommand要素の「insertionPosition」属性に割り当てる。

【0273】

XMT-A親要素の「position」属性の値が「END」である場合には、値「3」を、newCommand要素の「insertionPosition」属性に割り当てる。

【0274】

XMT-A親要素の「position」属性の値が「BEGIN」でも「END」でもない場合には、値「0」を、newCommand要素の「insertionPosition」属性に割り当て、XMT-A親要素の「position」属性の値を、newCommand要素の「position」属性に割り当てる。

20

【0275】

7. 1. 7. 6 Deleteコマンドの処理

「Delete」コマンド要素の処理に使用されるステップを、図98に示す。この手順の親要素は、XMT-A Deleteコマンド要素310である。このコマンド要素は、XMT-A「par」要素200か、Conditional ノード要素3040の「buffer」属性要素410に従属する場合がある。Deleteコマンド要素が、XMT-A「par」要素200に従属する場合に、mp4bifs「ターゲット」要素は、4330で作成されたcommandFrame要素2820および2830である。Deleteコマンド要素がConditional ノード要素3040の「buffer」属性要素410に従属する場合に、mp4-bifs「ターゲット」要素は、mp4-bifs Conditional node要素である。

30

【0276】

動作4500で、親要素の「atRoute」属性の値を、量「RouteId」に割り当てる。

【0277】

「atRoute」属性の値が指定されている場合に、動作4510で、標準XML手段を使用して、新しいmp4-bifs DeleteRouteコマンド要素（「newCommand」）を作成し、量「RouteId」の値を、newCommand要素の「routeId」属性に割り当てる。次に、標準XML手段を使用して、このnewCommand要素を現在のmp4-bifsターゲット要素に付加する。

40

【0278】

動作4520で、親要素の「atNode」属性の値を量「NodeId」に割り当てる。

【0279】

「atNode」属性の値が指定されていない場合に、動作4530で、XMT-A文書が有効でない。

【0280】

「atNode」属性の値が指定されている場合に、動作4540で、親要素の「at

50

Field」属性の値を、量「FieldName」に割り当てる。「atField」属性の値が指定されている場合に、量「FieldName」の値をストリング「children」と比較する。

【0281】

「atField」属性の値の値が指定されており、量「FieldName」の値が「children」でない場合に、動作4560で、標準XML手段を使用して、新しいmp4-bifs DeleteIndexedValueコマンド要素(newCommand)を作成する。量atFieldの値を、newCommand要素の「inFieldName」属性に割り当てる。

【0282】

10

XMT-A親要素の「position」属性の値が「BEGIN」である場合に、値「2」を、newCommand要素の「deletionPosition」属性に割り当てる。

【0283】

XMT-A親要素の「position」属性の値が「END」である場合に、値「3」を、newCommand要素の「deletionPosition」属性に割り当てる。

【0284】

XMT-A親要素の「position」属性の値が「BEGIN」でも「END」でもない場合には、値「0」をnewCommand要素の「deletionPosition」属性に割り当て、XMT-A親要素の「position」属性の値を、newCommand要素の「position」属性に割り当てる。

20

【0285】

その後、標準XML手段を使用して、newCommand要素を現在のmp4-bifsターゲット要素に付加する。

【0286】

「atField」属性の値が指定されていないか、量「FieldName」の値が「children」である場合に、動作4580で、標準XML手段を使用して、新しいmp4-bifs DeleteNodeコマンド要素(newCommand)を作成する。

30

【0287】

量「NodeId」の値を、BIFS NodeIdテーブル(図89)の項目3966と比較する。一致する項目の「position」3960の値を、整数量atNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「nodeId」属性に割り当てる。

【0288】

その後、標準XML手段を使用して、newCommand要素を現在のmp4-bifsターゲット要素に付加する。

【0289】

40

7.1.7.7 Replaceコマンドの処理

XMT-A「Replace」要素を処理するのに使用されるステップを、図99に示す。この手順の親要素は、XMT-A Replaceコマンド要素320である。このコマンド要素は、XMT-A「par」要素200またはConditional ノード要素3040の「buffer」属性要素410に従属する場合がある。Replaceコマンド要素が、XMT-A「par」要素200に従属する場合に、mp4bifs「ターゲット」要素は、4330で作成されたcommandFrame要素2820および2830である。Replaceコマンド要素がConditional node要素3040の「buffer」属性要素410に従属する場合に、mp4-bifs「ターゲット」要素は、mp4-bifs Conditional node要素である。

【0290】

50

まず、値「偽」を、2つのboolean値、bReplaceNodeおよびbReplaceValueに割り当てる。

【0291】

動作4600で、親要素の「atNode」属性の値を、量「NodeId」に割り当てる。「atNode」属性の値が指定されていない場合には、この手順は、動作4636で継続される。

【0292】

「atNode」属性の値が指定されている場合に、動作4604で、親要素の「atField」属性の値を、量「FieldName」に割り当てる。

【0293】

「atField」属性の値が指定されていない場合に、動作4612で、標準XML手段を使用して、新しいmp4-bifs ReplaceNodeコマンド要素（「newCommand」）を作成する。量「NodeId」の値を、BIFS NodeIdテーブル（図89）の項目3966と比較する。一致する項目の「position」3960の値を、整数量atNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「nodeId」属性に割り当てる。

【0294】

標準XML手段を使用して、newCommand要素をmp4-bifsターゲット要素に付加する。値「真」を、boolean量「bReplaceNode」に割り当て、この手順は、動作4636で継続される。

【0295】

「atField」属性の値が指定されている場合に、動作4616で、標準XML手段を使用して、新しいmp4-bifs ReplaceFieldコマンド要素（「newCommand」）を作成する。量「NodeId」の値を、BIFS NodeIdテーブル（図89）の項目3966と比較する。一致する項目の「position」3960の値を、整数量atNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「nodeId」属性に割り当てる。

【0296】

量「FieldName」の値を、newCommand要素の「inFieldName」属性に割り当てる。その後、標準XML手段を使用して、newCommand要素を現在のmp4-bifsターゲット要素に付加する。

【0297】

動作4620で、親要素の「value」属性の値を量「value」に割り当てる。

【0298】

XMT-A BIFSコマンド要素の「value」属性の値が指定されている場合に、動作4624で、新しいmp4bifs newCommand要素の「value」属性に値を割り当てる。ほとんどの場合に、新しいmp4bifs newCommand要素の「value」属性に割り当てられる値は、XMT-A BIFSコマンド要素の「value」属性の値と等しい。下で識別されるいくつかの場合（データ・フォーマット変換）に、新しいmp4bifs newCommand要素の「value」属性に割り当てられる値は、XMT-A BIFSコマンド要素の「value」属性の値から導出される。この場合に、この手順が完了し、処理は、動作4336またはXMT-A Conditional node要素の処理4890で継続される。

【0299】

「value」属性の値が指定されていない場合に、動作4628で、値「真」をboolean量「bReplaceField」に割り当てる。現在の親ノードに直接に従属するすべての要素の要素名のリストからなるストリング量「childNames」を作成する。

【0300】

動作4632で、ストリング量「childNames」の値を、新しいmp4-bi

10

20

30

40

50

f s 要素「newCommand」の「value」属性に割り当てる。

【0301】

動作4636で、値「0」を、他所で定義される他のインデックス値と別個であるインデックス値「i」に割り当てる。

【0302】

動作4640で、インデックス「i」の値を、量「numCmdChildren」の値と比較する。量numCmdChildrenの値は、現在の「親」要素によって所有される従属要素の個数を示す。

【0303】

インデックス「i」の値がnumCmdChildrenの値と等しい場合に、動作4644で、この手順が完了し、処理は、動作4336またはXMT-A Conditional node要素の処理4890で継続される。 10

【0304】

インデックス「i」の値がnumCmdChildrenの値と等しくない場合に、動作4648で、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入手し、結果の従属要素を「replaceChild」要素として識別する。replaceChild要素の要素名は、ストリング量「childName」によって識別される。"

【0305】

動作4652で、ストリング量childNameの値を、ストリング「Scene」と比較する。ストリング量childNameの値が「Scene」である場合に、動作4656で、手順「ReplaceSceneコマンドの作成」を実行する。その後、標準XML手段を使用して、結果のnewCommand要素を現在のmp4-bifsターゲット要素に付加する。その後、この手順は、動作4696で継続される。 20

【0306】

ストリング量childNameの値が「Scene」でない場合に、動作4660で、ストリング量childNameの値をストリング「ROUTE」と比較する。ストリング量childNameの値が「ROUTE」である場合に、動作4664で、手順「ReplaceRouteコマンドの作成」を実行する。その後、標準XML手段を使用して、結果のnewCommand要素を現在のmp4-bifsターゲット要素に付加する。その後、この手順は、動作4696で継続される。 30

【0307】

ストリング量childNameの値が「ROUTE」でない場合に、動作4668で、boolean量bReplaceNodeの値を「真」と比較する。

【0308】

boolean量bReplaceNodeの値が「真」である場合に、動作4672で、現在のreplaceChild要素を親要素として使用して、手順「XMT-A BIFSノードの処理」を実行する。その後、標準XML手段を使用して、結果のmp4-bifsノード要素を現在のmp4-bifsターゲット要素に付加する。処理は、動作4696で継続される。 40

【0309】

boolean量bReplaceNodeの値が「真」でない場合に、動作4680で、boolean量bReplaceFieldの値を値「真」と比較する。

【0310】

boolean量bReplaceFieldの値が「真」である場合に、動作4684で、現在のreplaceChildを親要素として使用して、手順「XMT-A BIFSノードの処理」を実行する。その後、標準XML手段を使用して、結果のmp4-bifsノード要素をnewCommand要素に付加する。処理は、動作4696で継続される。

【0311】

boolean量bReplaceNodeの値が「真」でない場合に、動作4690で、XMT-A文書が有効でない。現在のreplaceChildを親要素として使用して、手順「XMT-A BIFSノードの処理」を実行する。その後、標準XML手段を使用して、結果のmp4-bifsノード要素を現在のmp4-bifsターゲット要素に付加する。処理は、動作4696で継続される。

【0312】

動作4696では、インデックス「i」の値を「1」だけ増分し、動作4640でのnumCmdChildrenとの比較を繰り返す。

【0313】

7.1.7.8 「ReplaceRouteコマンド作成」手順

10

標準XML手段を使用して、新しいmp4-bifs ReplaceRoute要素（「newCommand」）を作成する。

【0314】

XMT-A親要素の「fromNode」属性の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量「fromNodeId」に割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「departureNode」属性に割り当てる。

【0315】

XMT-A親要素の「fromField」属性の値を、newCommand要素の「departureFieldName」属性に割り当てる。

20

【0316】

XMT-A親要素の「toNode」属性の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量fromNodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「arrivalNode」属性に割り当てる。

【0317】

XMT-A親要素の「toField」属性の値を、newCommand要素の「arrivalFieldName」属性に割り当てる。

【0318】

XMT-A親要素の「atRoute」属性の値を、BIFS RouteIdテーブル（図90参照）の項目3976と比較する。一致する項目の「position」3970の値を、整数量routeIdに割り当て、結果を「1」だけ増分する。その結果をnewCommand要素の「routeId」属性に割り当てる。XMT-A親要素のatRoute属性の値が指定されていない場合に、そのXMT-A文書は無効である。

30

【0319】

7.1.7.9 「ReplaceSceneコマンド作成」手順

mp4-bifs「ReplaceScene」コマンド要素を作成するのに使用されるステップを、図100に示す。この手順のXMT-A親要素は、XMT-A Sceneコマンド要素370である。このコマンド要素は、必ずXMT-A「Replace」要素320に従属する。

40

【0320】

動作4700で、標準XML手段を使用して、新しいmp4-bifs「ReplaceScene」コマンド要素2930（newCommand）を作成する。標準XML手段を使用して、この新しいコマンド要素を現在のmp4-bifsターゲット要素に付加する。mp4-bifsターゲット要素は、mp4-bifs commandFrame要素2830またはmp4-bifs Conditional node要素でなければならない。値「偽」を、boolean量「bHaveRoutes」に割り当てる。

【0321】

50

動作4710で、XMT-A「Scene」要素の「useNames」属性の値を、boolean量「USENAMES」に割り当てる。

【0322】

XMT-A「Scene」要素の「useNames」属性の値が指定されている場合に、動作4716で、boolean量「USENAMES」の値を値「真」と比較する。

【0323】

XMT-A「Scene」要素の「useNames」属性の値が指定されていない場合、またはboolean量「USENAMES」の値が「真」でない場合に、動作4720で、値「偽」をboolean量「bUseNames」に割り当てる。

10

【0324】

boolean量「USENAMES」の値が「真」である場合に、動作4726で、値「真」をboolean量「bUseNames」に割り当てる。

【0325】

動作4730で、値「0」を、他所で定義される他のインデックス値と別個であるインデックス値「i」に割り当てる。

【0326】

動作4740で、インデックス「i」の値を量「numSceneChildren」の値と比較する。量numSceneChildrenの値は、このXMT-A Scene要素によって所有される従属要素の個数を示す。

20

【0327】

インデックス「i」の値がnumSceneChildrenの値と等しい場合に、動作4746で、この手順が完了し、処理は、動作4656で継続される。

【0328】

インデックス「i」の値がnumSceneChildrenの値と等しくない場合に、動作4750で、標準XML手段を使用して、XMT-A Scene要素に従属するi番目の要素を入手し、結果の従属要素を「sceneChild」要素として識別する。sceneChild要素の要素名は、ストリング量「childName」によって識別される。

【0329】

動作4760で、ストリング量childNameの値をストリング「ROUTE」と比較する。

30

【0330】

ストリング量childNameの値が「ROUTE」である場合に、動作4766で、boolean量bHaveRoutesの値を値「真」と比較する。

【0331】

boolean量bHaveRoutesの値が「真」でない場合に、動作4770で、標準XML手段を使用して、新しいmp4-bifs「Routes」要素を作成する。標準XML手段を使用して、結果のmp4-bifs「Routes」要素をnewCommand要素に付加する。値「真」を、boolean量「bHaveRoutes」に割り当てる。

40

【0332】

動作4776で、標準XML手段を使用して、新しいmp4-bifs「Route」要素を作成する。標準XML手段を使用して、結果のmp4-bifs「Route」要素をmp4-bifs「Routes」要素に付加する。

【0333】

XMT-A親要素の「fromNode」属性の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量fromNodeIdに割り当て、結果を「1」だけ増分する。その結果を、mp4-bifs Route要素の「fromNode」属性に割り当てる。

50

【0334】

XMT-A親要素の「fromField」属性の値を、mp4-bifs Route要素の「fromFieldName」属性に割り当てる。

【0335】

XMT-A親要素の「toNode」属性の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量fromNodeIdに割り当て、結果を「1」だけ増分する。その結果を、mp4-bifs Route要素の「toNode」属性に割り当てる。

【0336】

XMT-A親要素の「toField」属性の値を、mp4-bifs Route要素の「toFieldName」属性に割り当てる。 10

【0337】

XMT-A親要素の「DEF」属性の値が指定されている場合に、この属性の値を、BIFS RouteIdテーブル（図90参照）の項目3976と比較する。一致する項目の「position」3970の値を、整数量routeIdに割り当て、結果を「1」だけ増分する。その結果をmp4-bifs Route要素の「routeId」属性に割り当てる。boolean量bUseNamesの値が真である場合に、「DEF」属性の値を、mp4-bifs Route要素の「name」属性に割り当てる。

【0338】

20
 スtring量childNameの値が「ROUTE」でない場合に、動作4780で、現在のsceneChild要素を親要素として使用して、手順「XMT-A BIFSノードの処理」を実行する。標準XML手段を使用して、結果のmp4-bifsノード要素をnewCommand要素に付加する。処理は、動作4790で継続される。

【0339】

動作4790では、インデックス「i」の値を「1」だけ増分し、動作4740でのnumSceneChildrenとの比較を繰り返す。

【0340】

7. 1. 7. 10 「XMT-A BIFSノードの処理」手順

30
 100タイプを超えるBIFSノードが、MPEG-4 Systems仕様で定義されている。各MPEG-4 BIFSノードは、特定のノード名および1組の名前付きプロパティ・フィールドを有する。各名前付きプロパティ・フィールドは、boolean、integer、float、string、「node」、または「buffer」などの特定のデータ型を有する。MPEG-4 BIFSノードのタイプごとに、対応する同一の名前のノード要素が、XMTA文書およびmp4bifs文書で定義される。mp4bifs文書について定義された各ノード要素は、対応するMPEG-4 BIFSノードのプロパティ・フィールドの名前と一致する名前を有する属性の組を所有する。

【0341】

40
 図75から77からわかるように、データ型「node」または「buffer」を有する各MPEG-4 BIFSプロパティ・フィールドは、mp4bifsノード要素の1つまたは複数の従属要素によって表すこともでき、mp4bifsノード要素の対応する属性は、このプロパティ・フィールドに関連する従属要素の要素名のリストからなる。これらの従属要素は、ノード要素またはコマンド要素とすることができる。この形で、各mp4bifsノード要素の構造は、対応するMPEG-4 BIFSノードの構造を模倣する。

【0342】

50
 XMT-A文書について定義されたノード要素は、mp4bifs文書について定義されたノード要素に類似するが、XMT-Aノード要素ごとに定義される属性に、「node」または「buffer」のデータ型を有しないプロパティだけが含まれる点異なる。「node」または「buffer」のデータ型を有するMPEG-4 BIFSノード

ドのプロパティごとに、XMT-A仕様では、属性を有しない同一の名前の従属属性要素が定義され、対応するプロパティ・フィールドは、これらの属性要素に従属するノード要素またはコマンド要素によって表される。

【0343】

図101からわかるように、XMT-A BIFSノード要素の変換処理は、XMT-Aノード要素の「USE」属性の値をストリング量「nodeRef」に割り当てることによって、動作4800で開始される。XMT-Aノード要素の「USE」属性の値が指定されている場合には、動作4806で、標準XML手段を使用して、新しいmp4bifs ReusedNode要素を作成する。標準XML手段を使用して、新しいReusedNode要素を現在のmp4bifsターゲット要素に挿入する。

10

【0344】

動作4810で、ストリング量「nodeRef」の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量nodeIdに割り当て、結果を「1」だけ増分する。その結果を、newCommand要素の「nodeRef」属性に割り当てる。このXMT-Aノード要素の処理は、動作4816で完了し、処理は、XMT-A BIFSコマンド要素またはこのXMT-Aノード要素を所有した親XMT-A BIFSノード要素で継続される。

【0345】

XMT-Aノード要素の「USE」属性の値が指定されていない場合に、動作4820で、標準XML手段を使用して、新しいmp4bifs NodeName要素を作成するが、「NodeName」は、現在のXMT-A BIFSノード要素の名前を表す。標準XML手段を使用して、新しい「NodeName」要素を現在のmp4bifsターゲット要素に挿入する。たとえば、現在のXMT-A BIFSノード要素の要素名が「Geometry」である場合に、新しいmp4bifs「Geometry」要素を作成し、現在のmp4bifsターゲット要素に挿入する。

20

【0346】

XMT-A BIFSノード要素の「DEF」属性の値が指定されている場合に、「DEF」属性の値を、BIFS NodeIdテーブル（図89参照）の項目3966と比較する。一致する項目の「position」3960の値を、整数量nodeIdに割り当て、結果を「1」だけ増分する。その結果を、mp4bifs「NodeName」要素の「nodeId」属性に割り当てる。boolean量「bUseNames」が真である場合に、XMT-A BIFSノード要素の「DEF」属性の値を、mp4bifs NodeName要素の「name」属性に割り当てる。

30

【0347】

XMT-A BIFSノード要素の他のすべての属性の値が、新しいmp4bifs NodeName要素の同一の名前の属性に値を割り当てるのに使用される。ほとんどの場合に、mp4bifs NodeName要素の各属性に割り当てられる値は、XMT-A BIFSノード要素の対応する属性の値と等しい。下で識別されるいくつかの場合（データ・フォーマット変換）に、mp4bifs NodeName要素の属性に割り当てられる値は、XMT-A BIFSノード要素の対応する属性の値から導出される。

40

【0348】

動作4826で、値「0」を、他所で定義される他のインデックス値と別個であるインデックス値「i」に割り当てる。

【0349】

動作4830で、インデックス「i」の値を、量「numNodeChildren」の値と比較する。量numNodeChildrenの値は、現在のXMT-A BIFSノード要素によって所有される従属要素の個数を示す。numNodeChildrenの非0の値は、「Node」または「Command Buffer」のフィールド・

50

データ型を有する1つまたは複数のデータ・フィールドを有するMPEG-4 BIFSノードを表すXMT-A BIFSノード要素に限って可能である。

【0350】

インデックス「i」の値がnumNodeChildrenの値と等しい場合に、動作4836で、この手順が完了し、処理は、XMT-A BIFSコマンド要素またはこのXMT-Aノード要素を所有した親XMT-A BIFSノード要素で継続される。

【0351】

インデックス「i」の値がnumNodeChildrenの値と等しくない場合に、動作4840で、標準XML手段を使用して、現在の親要素に従属するi番目の要素を入手し、結果の従属要素を「nodeChild」要素として識別する。nodeChild要素の要素名は、ストリング量「childName」によって識別される。量「childName」の値は、「Node」または「Command Buffer」のフィールド・データ型を有するMPEG-4 BIFSノード・データ・フィールドのフィールド名と一致する。 10

【0352】

動作4846で、標準XML手段を使用して、nodeChild要素に従属するすべての要素の要素名を得る。これらの要素名のそれぞれの値を、空白スペースで区切って一緒に連結して、ストリング量「NameList」に割り当てる。結果のストリング量「NameList」の値を、現在のmp4bifs NodeName要素のchildName属性に割り当てる。たとえば、childNameの値が「children」である場合に、XMT-A「children」要素に従属するXMT-A要素の要素名のリストが、現在のmp4bifs NodeName要素の「children」属性に割り当てられる。 20

【0353】

動作4850で、値「0」を、他所で定義される他のインデックス値と別個であるインデックス値「j」に割り当てる。

【0354】

動作4856で、インデックス「j」の値を、量「numNodeChildChildren」の値と比較する。量numNodeChildChildrenの値は、現在の「nodeChild」要素によって所有される従属要素の個数を示す。 30

【0355】

インデックス「j」の値が、量numNodeChildChildrenの値と等しい場合に、動作4860で、インデックス「i」の値を「1」だけ増分し、動作4830でのnumNodeChildrenとの比較を繰り返す。

【0356】

インデックス「j」の値が、量「numNodeChildChildren」の値と等しくない場合に、動作4866で、標準XML手段を使用して、現在のnodeChild要素に従属するj番目の要素を入手し、結果の従属要素を「attributeChild」要素として識別する。attributeChild要素の要素名は、ストリング量「attributeChildName」によって識別される。 40

【0357】

動作4866で、ストリング量childNameの値を、ストリング「buffer」と比較する。

【0358】

ストリング量childNameの値が「buffer」である場合に、動作4870で、手順「XMT-Aコマンドの処理」を実行する。その後、標準XML手段を使用して、結果のnewCommand要素を現在のmp4-bifs NodeName要素に付加する。手順「XMT-Aコマンドの処理」は、childNameの値としてattributeChildNameの値を使用する、図96に示された手順「XMT-A par要素の処理（パス2）」の動作4360から4386と同等である。これは、現在 50

の手順が、必ず手順「XMT-A par要素の処理（パス2）」に従属するので、再帰処理である。その後、この手順は、動作4890で継続される。

【0359】

ストリング量childNameの値が「buffer」でない場合に、動作4880で、手順「XMT-A BIFSノードの処理」を再帰的に実行する。その後、標準XML手段を使用して、結果のNodeName要素を現在のmp4-bifs NodeName要素に付加する。

【0360】

動作4890で、インデックス「j」の値を「1」だけ増分し、動作4856でのnumNodeChildChildrenとの比較を繰り返す。

10

【0361】

7. 1. 7. 11 データ・フォーマット変換

データ・フォーマット変換は、XMT-A BIFSノードの次のプロパティ・フィールド属性に適用される。これらの変換は、XMT-A Insertコマンド要素動作4430およびXMT-A Replaceコマンド要素動作4624の「value」属性の値にも適用される。XMT-A InsertコマンドおよびXMT-A Replaceコマンドの場合に、データ型は、対応するatField属性の値によって決定される。

【0362】

1. データ型「color」を有するフィールド・プロパティの各XMT-A属性値は、6桁の16進ストリング「#RRGGBB」によって表される。これは、3部分の10進表現「rrr ggg bbb」に変換され、「rrr」は、16進数値0xRRを256で割ったものの10進表現、「ggg」は、16進数値0xGGを256で割ったものの10進表現、「bbb」は、16進数値0xBBを256で割ったものの10進表現である。

20

【0363】

2. データ型「string」を有するフィールド・プロパティの各XMT-A属性値は、XMT-Aについて定義された引用符付きストリング・フォーマットから、mp4bifsによって使用される代替フォーマットに変換される。この変換には、バックslash文字（\）が前につかない場合の「引用符」文字（"）の除去、ストリング内の空白スペースおよび他の「特殊」文字のパーセント文字（%）およびそれに続く2桁の16進数コードへの置換、ならびに空白スペースによる複数ストリングの分離が含まれる。「特殊」文字に、空白スペース、引用符、パーセント（%）、アンパサンド（&）、大なり（>）、32未満の数値を有する文字、および128以上の数値を有する文字が含まれる。空白スペースは、複数のストリングからなる属性フィールド内の個々のストリングを区切るのに使用される。このストリング属性の変換は、本発明に必要ではなく、本発明の代替実施形態では省略することができる。

30

【0364】

3. データ型「url」を有するフィールド・プロパティのXMT-A属性値が、「od:／／」または「odid:／／」から始まる場合に、対応するmp4bifs属性に割り当てられる値は、「Od id:」と、それに続く、XMT-A url属性値の残り（「od:／／」または「odid:／／」に続く）と一致するObjectDescriptor ID 3910を有するオブジェクト・テーブル（図88参照）内の項目3900のインデックスによって与えられる。

40

【0365】

7. 1. 8 mp4bifs文書へのコマンド・フレームの挿入

XMT-A Body要素120の要素に対する第2パス動作3130が完了した後に、commandFrame要素2830の一時的に順序付けられたリストの内容を、mp4bifs文書2800に挿入する。空のcommandFrameが破棄され、「time」属性の同一の値を有す複数のcommandFrame要素が、単一のcomm

50

andFrame要素に合併される。

【0366】

mp4file文書の「moov」要素の「duration」属性の値が、最後のcommandFrame要素2830のtime値に基づいて更新される。この属性に割り当てられる値は、最後のcommandFrame要素から入手された秒単位の値と、「moov」要素2320のtimeScale属性の積によって決定される。sdsmデータのtrak要素2350および2600の「duration」属性およびこのtrak要素2600に従属する「mdia」要素2604の「duration」属性のそれぞれが、類似する形で更新される。

【0367】

10

7. 1. 9 odsmのmdat要素へのODコマンドの挿入

XMT-A文書にメディア・オブジェクトが含まれることがわかった場合に、XMT-A「Body」要素動作の第1パス3120で作成されたオブジェクト・テーブル（図88参照）を使用して、odsm(object descriptor stream)のXML記述を構成する。このテーブルが項目を有しない場合には、odsmが存在せず、このステップはスキップされる。オブジェクト・テーブルに少なくとも1つの項目がある場合には、このテーブルを使用して、図92に示されている、ソート済みオブジェクト・テーブルを作成する。

【0368】

ソート済みオブジェクト・テーブルの各entry（行）3990は、オブジェクト・テーブル内のObjectDescriptorID項目3920に対応するOdId値3992、time値3994、およびbooleanフラグ(start)3996からなる。

20

【0369】

ソート済みオブジェクト・テーブルに、オブジェクト・テーブル内の項目3900ごとに2つのentry 3990が含まれる。OdId列3992の各項目の値は、オブジェクト・テーブル内の対応する項目3920にある値のコピーである。time列3994の項目の値は、オブジェクト・テーブル内のstartTime列3930またはstopTime列3940のいずれかの対応する項目の値のコピーである。ソート済みオブジェクト・テーブルのtime列3994の項目が、オブジェクト・テーブルのstartTime列3930の対応する項目から導出された場合に、値「真」が、ソート済みオブジェクト・テーブルのstart列3996の対応する項目に割り当てられる。そうでない場合には、値「偽」が、ソート済みオブジェクト・テーブルのstart列3996の対応する項目に割り当てられる。

30

【0370】

ソート済みオブジェクト・テーブルの項目は、time列3994の昇順でソートされる。ソート済みオブジェクト・テーブルの作成の後に、odsmのXML表現が、図102に示されているように作成される。

【0371】

動作4900で、値「0」を整数量「numSamples」、「odsmSize」、および「sampleSize」に割り当てる。負の値を、浮動小数点量「prevTime」に割り当てる。

40

【0372】

動作4906で、標準XML手段を使用して、odsmの「mdat」要素2310および2400内の「odsmChunk」要素2470を突き止める。標準XML手段を使用して、odsmについて前に作成された「trak」要素2350および2600内の「stts」要素2660、「stsz」要素2668、および「stsc」要素2656を突き止める。標準XML手段を使用して、この「stsc」要素2656に従属する「sampleToChunk」要素を突き止める。これらの要素は、すべて、前にXMT-A「ヘッダ」要素の処理3116中に作成されたものである。

50

【0373】

動作4910で、値「0」を、他所で定義される他のインデックス値と別個であるインデックス値「i」に割り当てる。

【0374】

動作4916で、インデックス「i」の値を、量「numEntries」の値と比較する。量numEntriesの値は、ソート済みオブジェクト・テーブルの行数3990を示す。

【0375】

インデックス「i」の値がnumEntriesの値と等しくない場合に、動作4940で、ソート済みオブジェクト・テーブルのtime列3994のi番目の項目の値を、量prevTimeの現在値と比較する。 10

【0376】

ソート済みオブジェクト・テーブルのtime列3994のi番目の項目の値が量prevTimeの現在値より大きい場合に、動作4946で、標準XML手段を使用して、新しいmp4file_odsmSample要素を作成する。そうでない場合には、処理は動作4970で継続される。

【0377】

次に、標準XML手段を使用して、新しいodsmSample要素を、動作4906で得られたodsmChunk要素に挿入する。量odsmSizeの現在値を、新しいodsmSample要素の「offset」属性に割り当て、ソート済みオブジェクト・テーブルの現在の項目（「i」）の「time」列3994の値を、新しいodsmSample要素の「time」属性に割り当てる。 20

【0378】

動作4950で、インデックス「i」の値を「0」と比較する。

【0379】

インデックス「i」の値が0を超える場合に、動作4956で、標準XML手段を使用して、新しいmp4file_timeToSample要素を作成する。そうでない場合には、処理は動作4966で継続される。

【0380】

次に、標準XML手段を使用して、新しいtimeToSample要素を、動作4906で得られたstts要素に挿入する。ソート済みオブジェクト・テーブルの現在の項目のtime値3994と量「prevTime」の値との間の差を、新しいtimeToSample要素の「duration」属性に割り当てる。値「1」を、新しい「timeToSample」要素の「numSamples」属性に割り当てる。 30

【0381】

標準XML手段を使用して、新しいmp4file_sampleSize要素を作成する。次に、標準XML手段を使用して、新しいsampleSize要素を、動作4906で得られたstsz要素に挿入する。量sampleSizeの値を、新しい「sampleSize」要素の「size」属性に割り当てる。

【0382】

動作4960で、量odsmSizeの値を量sampleSizeの値だけ増分し、値「0」を量sampleSizeの値に割り当て、量numSamplesの値を「1」だけ増分する。 40

【0383】

動作4966で、ソート済みオブジェクト・テーブルのtime列3994のi番目の項目の値を、量「prevTime」に割り当てる。

【0384】

動作4970で、ソート済みオブジェクト・テーブルのstart列3996のi番目の項目の値を、値「真」と比較する。

【0385】

ソート済みオブジェクト・テーブルの `start` 列 3996 の `i` 番目の項目の値が値「真」と等しい場合に、動作 4980 で、標準 XML 手段を使用して、新しい `mp4file ObjectDescriptorUpdate` 要素 2540 を作成する。次に、標準 XML 手段を使用して、新しい `ObjectDescriptorUpdate` 要素 2540 を、動作 4946 で作成された `odsmSample` 要素 2510 に挿入する。

【0386】

標準 XML 手段を使用して、新しい `mp4file ObjectDescriptor` 要素 2550 を作成する。次に、標準 XML 手段を使用して、新しい `ObjectDescriptor` 要素 2550 を新しい `ObjectDescriptorUpdate` 要素 2540 に挿入する。ソート済みオブジェクト・テーブルの現在の項目に関連する量「`OdId`」3992 の値を、新しい `ObjectDescriptor` 要素 2550 の「`OdId`」属性に割り当てる。

【0387】

標準 XML 手段を使用して、新しい `mp4file EsIdRef` 要素 2560 を作成する。次に、標準 XML 手段を使用して、新しい `EsIdRef` 要素 2560 を新しい `ObjectDescriptor` 要素 2550 に挿入する。ソート済みオブジェクト・テーブルの現在の項目の `OdId` 値 3992 と一致する「`OdId`」値 3920 に関連するオブジェクト・テーブル（図 88 参照）内の「`EsId`」項目 3950 の値を、「`EsIdRef`」要素 2560 の「`EsId`」属性に割り当てる。

【0388】

動作 4986 で、量 `sampleSize` の値を「10」だけ増分する。

【0389】

ソート済みオブジェクト・テーブルの `start` 列 3996 の `i` 番目の項目の値が値「真」を有しない場合に、動作 4990 で、標準 XML 手段を使用して、新しい `mp4file ObjectDescriptorRemove` 要素 2570 を作成する。次に、標準 XML 手段を使用して、新しい `ObjectDescriptorRemove` 要素 2570 を、動作 4946 で作成された `odsmSample` 要素 2510 に挿入する。ソート済みオブジェクト・テーブルの現在の項目に関連する量「`OdId`」3992 の値を、新しい `ObjectDescriptorRemove` 要素 2570 の「`OdId`」属性に割り当てる。

【0390】

動作 4996 で、量 `sampleSize` の値を「4」だけ増分する。

【0391】

動作 4936 で、インデックス「`i`」の値を「1」だけ増分し、インデックス「`i`」の値 `numEntries` との比較を繰り返す。

【0392】

インデックス「`i`」の値が `numEntries` の値と等しい場合に、動作 4920 で、量「`odsmSize`」の値を `sampleSize` の値だけ増分し、量 `numSamples` の値を「1」だけ増分する。

【0393】

動作 4926 で、標準 XML 手段を使用して、新しい `mp4file timeToSample` 要素を作成する。次に、標準 XML 手段を使用して、新しい `timeToSample` 要素を、動作 4906 で得られた `stts` 要素に挿入する。ソート済みオブジェクト・テーブルの現在の項目の `time` 値 3994 と量「`prevTime`」の値との間の差を、新しい `timeToSample` 要素の「`duration`」属性に割り当てる。値「1」を、新しい「`timeToSample`」要素の「`numSamples`」属性に割り当てる。

【0394】

標準 XML 手段を使用して、新しい `mp4file sampleSize` 要素を作成する。次に、標準 XML 手段を使用して、新しい `sampleSize` 要素を、動作 49

10

20

30

40

50

06で得られたs t s z要素に挿入する。量s a m p l e S i z eの値を、新しい「s a m p l e S i z e」要素の「s i z e」属性に割り当てる。

【0395】

動作4930で、量「numSamples」の値を「sampleToChunk」要素に割り当てる。量odsmSizeの値を「odsmChunk」要素の「size」属性に割り当てる。

【0396】

7. 1. 10 mp4-bifs文書およびmp4-file文書のbifsConfigの更新

BIFS NodeIDテーブル（図89参照）の項目数を表すのに必要な最小のビット数を判定し、量「numNodeIDBits」に割り当てる。これは、2の「n」乗がこのテーブルの項目数より大きくなる、最小の数「n」である。量numNodeIDBitsの値を、ステップ2で作成された「bifsConfig」要素（2810）の「nodeIDBits」属性に割り当てる。この値は、ステップ4で作成されたsdsm(scene description stream)の「trak」要素2350および2600に含まれる「BIFS__DecoderConfig」要素2720の「nodeIDBits」属性にも割り当てられる。

【0397】

類似する形で、BIFS RouteeIDテーブル（図90参照）の項目数を表すのに必要な最小のビット数を判定し、量「numRouteIDBits」に割り当てる。量numRouteIDBitsの値を、ステップ2で作成された「bifsConfig」要素2810のrouteIDBits属性に割り当てる。この値は、ステップ4で作成されたsdsm(scene description stream)の「trak」要素2350および2600に含まれる「BIFS__DecoderConfig」要素2720のrouteIDBits属性にも割り当てられる。

【0398】

このステップによって、mp4-file文書およびmp4-bifs文書の作成が完了する。mp4バイナリ・ファイルを作成する処理は、「3. b. Creation of an mp4 binary file based on the intermediate XML documents」で継続される。

【0399】

7. 1. 10. 1 ES__Descriptorの処理

各「ES__Descriptor」要素は、図82に示されているように処理される。この手順は、XMT-A文書100のBody要素120に含まれるES__Descriptor要素630ならびにXMT-A文書100のHeader要素110に含まれるES__Descriptor要素180および190の処理に使用される。

【0400】

各「ES__Descriptor」要素は、「ES__ID」という名前の属性を所有し、この属性の値が、ストリング量「ES__DescriptorID」に割り当てられる。

【0401】

手順「ES__Descriptorの処理」は、動作3400の手順「decConfigDescr要素の処理」から始まる。この手順は、次の4ステップからなる。

【0402】

1. 標準XML手段を使用して、ES__Descriptor要素640に従属するdecConfigDescr要素646を入手する。

【0403】

2. 標準XML手段を使用して、decConfigDescr要素646に従属するDecoderConfigDescriptor要素650を入手する。

【0404】

10

20

30

40

50

3. DecoderConfigDescriptor要素650の「streamType」属性の値を使用して、このES_Descriptor要素によって記述されるデータ・ストリームのstreamTypeプロパティの数値を確立する。「streamType」属性の値は、数値または、MPEG-4 systems仕様のテーブルで定義された英数字ストリングの組の1つからなるものとすることができる。これらの定義済みストリングに、「ObjectDescriptor」、「SceneDescription」、「Visual」、「Audio」などが含まれる。「streamType」属性の値が、これらのストリングの1つと一致する場合に、MPEG-4テーブルの関連する項目に基づいて、数値がstreamTypeに割り当てられる。たとえば、「streamType」属性の値が「ObjectDescriptor」である場合に、値1がiStreamTypeに割り当てられる。それ以外の場合に、「streamType」属性の値は、数値を表さなければならず、この数値が、このストリームのstreamTypeプロパティに割り当てられる。

10

【0405】

4. DecoderConfigDescriptor要素の「objectTypeIndication」属性の値を使用して、このES_Descriptor要素によって記述されるデータ・ストリームの「objectTypeIndication」属性の値は、数値またはMPEG-4 systems仕様のテーブルで定義された英数字ストリングの組の1つからなるものとすることができる。この定義済みストリングに、「MPEG4Systems1」、「MPEG4Visual」、「MPEG4Audio」、「Unspecified」などが含まれる。「objectTypeIndication」属性の値が、これらのストリングの1つと一致する場合に、MPEG-4テーブルの関連する項目に基づいて、数値がiObjectTypeに割り当てられる。たとえば、「objectTypeIndication」属性の値が「Unspecified」である場合に、値255がiObjectTypeに割り当てられる。そうでない場合に、「objectTypeIndication」属性の値は、数値を表さなければならず、この数値が、このストリームのobjectTypeプロパティに割り当てられる。

20

【0406】

手順「decConfigDescr要素の処理」(3400)の次に、手順「ES_Descriptorの処理」は、動作3410の手順「slConfigDescr要素の処理」で継続する。この手順は、次の3ステップからなる。

30

【0407】

1. 標準XML手段を使用して、「ES_Descriptor」要素640に従属する「slConfigDescr」要素660を入手する。

【0408】

2. 次に、標準XML手段を使用して、「slConfigDescr」要素660に従属する「SLConfigDescriptor」要素666を入手する。

【0409】

3. 「SLConfigDescriptor」666の「timeStampResolution」属性の値を使用して、このストリームのtimeScaleプロパティに数値を割り当てる。「timeStampResolution」属性の値が指定されていない場合には、デフォルト値をtimeScaleに割り当てる。このデフォルト値は、MPEG-4ビジュアル(iStreamType=4かつiObjectType=32)を除くすべてのストリームについて1000であり、MPEG-4ビジュアルの場合には、デフォルトtimeScale値は30である。

40

【0410】

動作3410の手順「slConfigDescr要素の処理」に続いて、手順「ES_Descriptorの処理」は、動作3420の手順「StreamSource要素の処理」で継続される。

50

【0411】

XMT-A Body要素120に含まれるES_Descriptor630の場合に、手順「StreamSource要素の処理」は、次の2ステップからなる。

【0412】

1. 標準XML手段を使用して、「ES_Descriptor」要素に従属する「StreamSource」要素を入手する。

【0413】

2. この「StreamSource」要素の「url」属性の値を、「mediaFileName」という名前の量に割り当てる。

【0414】

XMT-A Header要素110に含まれるES_Descriptor180および190の場合に、StreamSource要素は、存在せず、量「sdsmFileName」の値が、量「mediaFileName」に割り当てられる。

【0415】

動作3420の手順「StreamSource要素の処理」に続いて、手順「ES_Descriptorの処理」は、動作3430の手順「指定されたストリームのmdat要素の作成」で継続される。図83からわかるように、手順「指定されたストリームのmdat要素の作成」3430は、次のステップからなる。

【0416】

1. 動作3500：標準XMLを使用して、新しい「mdat」要素2310を作成し、mp4file文書2300内で前に作成された「moov」要素2320の前に挿入する。

【0417】

2. 動作3506：量「nextTrackId」の現在値を、新しいmdat要素2310の「mdatId」属性に割り当てる。この要素の「size」属性に、0の値を割り当てる。

【0418】

3a. 動作3510：手順「decConfigDescr要素の処理」動作3400によって確立されたstreamTypeプロパティを値「1」と比較する。

【0419】

4a. streamTypeプロパティの値が「1」である場合に、動作3516で、新しい「odsm」要素2420および2460を作成し、新しい「mdat」要素2310および2400に挿入し、動作3520で、量「nextTrackId」の現在値を、この新しい「odsm」要素2420の「trackID」属性に割り当て、動作3526で、新しい「odsmChunk」要素2470を作成し、新しい「odsm」要素2460に挿入し、動作3530で、値0を新しい「odsmChunk」要素2470の「offset」属性に割り当てる。

【0420】

3b. 動作3540：streamTypeプロパティの値が「1」でない場合に、streamTypeプロパティの値を値「3」と比較する。

【0421】

4b. streamTypeプロパティの値が「3」である場合に、動作3546で、新しい「sdsm」要素2410および2440を作成し、「mdat」要素2310および2400に挿入する。動作3550で、量「nextTrackId」の現在値を、この新しいsdsm要素2410の「trackID」属性に割り当て、量「mediaFileName」の値を、新しいsdsm要素2410の「xmlFile」属性に割り当てる。動作3556で、新しい「チャンク」要素2450を作成し、新しい「sdsm」要素2440に挿入する。動作3560で、値0を、新しい「チャンク」要素2450の「offset」属性に割り当てる。

【0422】

10

20

30

40

50

4 c. 動作 3 5 6 6 : `streamType` プロパティの値が「1」でも「3」でもない場合に、新しい「`mediaFile`」要素 2 4 3 0 および 2 4 8 0 を作成し、新しい「`mdat`」要素 2 3 1 0 および 2 4 0 0 に挿入する。動作 3 5 7 0 で、量「`nextTrackId`」の現在値を新しい「`mediaFile`」要素 2 4 3 0 の「`trackID`」属性に割り当てる。動作 3 5 7 6 で、新しい「チャンク」要素 2 4 9 0 を作成し、新しい「`mediaFile`」要素 2 4 8 0 に挿入する。動作 3 5 8 0 で、値 0 を、新しい「チャンク」要素 2 4 9 0 の「`offset`」属性に割り当てる。

【0 4 2 3】

値 0 を「`offset`」属性に割り当てる動作 3 5 3 0、3 5 6 0、および 3 5 8 0 の処理によって、手順「指定されたストリームの `mdat` 要素の作成」3 4 3 0 が完了する。この手順 3 4 3 0 に続いて、手順「`ES_Descriptor` の処理」は、手順「指定されたストリームの `trak` 要素の作成」3 4 4 0 で継続される。この手順は、下の「`trak` 要素の作成」で説明する。この手順 3 4 4 0 に続いて、手順「`ES_Descriptor` の処理」3 3 4 0 は、テスト「指定されたストリームは `sdsm` または `odsm` か」3 4 5 0 で継続される。

【0 4 2 4】

現在のストリームが `odsm` (`streamType` の値が 1) または `sdsm` (`streamType` の値が 3) である場合に、動作 3 4 6 0 で、新しい「`EsIdInc`」要素 2 3 8 0 を作成し、`mp4file` 文書 2 3 0 0 の「`mp4fi ods`」要素 2 3 6 0 に付加する。次に、量「`nextTrackId`」の値を、新しい「`EsIdInc`」要素 2 3 8 0 の「`trackID`」属性 2 3 9 0 に割り当てる。

【0 4 2 5】

そうでない場合 (量「`streamType`」の値が「1」でも「3」でもない) に、動作 3 4 7 0 で、量「`nextTrackId`」の値を、`odsm` の「`trak`」要素 2 6 0 0 の「`treref`」要素 2 6 3 6 内の「`mpod`」要素 2 6 4 0 の「`trackID`」要素の値に割り当てる。量「`nextTrackId`」の値は、`XMT-A` 「`Body`」要素に対する第 1 パスで作成されたオブジェクト・テーブルの「`OdId`」列 3 9 2 0 の項目にも割り当てられる。この項目は、「`ObjectDescriptorID`」項目 3 9 1 0 の値が、この「`ES_Descriptor`」要素 6 3 0 を含む「`ObjectDescriptor`」要素 6 0 0 の「`ObjectDescriptorID`」属性 6 0 6 と一致する行に対応する。量「`nextEsId`」の値を、このテーブルの同一の行の「`EsId`」項目 3 9 5 0 に割り当てる。その後、量「`nextEsId`」の値を 1 つ増分する。

【0 4 2 6】

どちらの場合でも、動作 3 4 8 0 で、量 `nextTrackId` の値を 1 つ増分し、量 `nextTrackId` の新しい値を、「`moov`」要素 2 3 2 0 の「`nextTrackID`」属性に割り当てる。

【0 4 2 7】

これで、図 8 2 に示された「`ES_Descriptor`」要素の処理が完了する。この手順は、`XMT-A` 文書 1 0 0 の「`Header`」要素 1 1 0 内の「`InitialObjectDescriptor`」要素 1 3 0 内の「`Descr`」1 6 0 要素内にある「`ES_Descriptor`」要素 1 8 0 および 1 9 0 ごとに実行される。この処理は、`XMT-A` 文書 1 0 0 の「`Body`」要素 1 2 0 内にある「`ObjectDescriptor`」要素 6 0 0 内の「`Descr`」6 1 0 要素内の「`ES_Descriptor`」要素 6 3 0 ごとにも実行される。

【0 4 2 8】

この手順 (`ES_Descriptor` の処理) の完了に続いて、`mp4file` 文書 2 2 5 0 作成の処理は、図 8 1 に示された手順「`esDescr` 要素の処理」の動作 3 3 5 0 のインデックス「`i`」の値の増分のステップで継続される。

【0 4 2 9】

7. 1. 10. 2 t r a k要素の作成

図84からわかるように、手順「指定されたストリームのt r a k要素の作成」3440は、次の11個のステップからなる。

【0430】

1. 動作3600で、標準XML手段を使用して、新しい「t r a k」要素2350および2600を作成し、m p 4 f i l e文書2300の「m o o v」要素2320に挿入する。

【0431】

新しいt r a k要素2600の次の属性に、値が割り当てられる。値「1」が、「f l a g s」属性に割り当てられる。1904年1月1日からの秒数と等しい値が、「c r e a t i o n T i m e」属性および「m o d i f y T i m e」属性に割り当てられる。量n e x t T r a c k I dの値が、t r a c k I D属性に割り当てられる。値「240」が、「t r a c k H e i g h t」属性に割り当てられる。値「320」が、「t r a c k W i d t h」属性に割り当てられる。 10

【0432】

s t r e a m T y p eプロパティの値が「1」または「3」である場合に、値「0」が、d u r a t i o n属性に割り当てられる。これらは、後で決定される訂正された値によって置換される、予備的な値にすぎない。それ以外では、囲むO b j e c t D e s c r i p t o r要素600のo b j e c t D e s c r i p t o r I Dが、X M T - A文書100のB o d y要素120に対する第1パス動作3120中に構成されるテーブルから対応するメディア持続時間値を入手するのに使用される。メディア持続時間値（秒単位）は、「S L C o n f i g D e s c r i p t o r」要素666から導出されるt i m e S c a l e値をかけられ、整数値に丸められる。 20

【0433】

s t r e a m T y p eプロパティの値が「1」である（o b j e c t d e s c r i p t o r s t r e a m）場合に、t r a c k I D属性の値が、量t r a c k I d F o r O d s mに割り当てられる。s t r e a m T y p eプロパティの値が「3」である（s c e n e d e s c r i p t i o n s t r e a m）場合に、t r a c k I D属性の値が、量t r a c k I d F o r S d s mに割り当てられる。

【0434】

2. 動作3606で、標準XML手段を使用して、新しい「m d i a」要素2604を作成し、ステップ1で作成した新しい「t r a k」要素2600に挿入する。 30

【0435】

この新しい「m d i a」要素2604の次の属性に、値を割り当てる。1904年1月1日からの秒数と等しい値を、「c r e a t i o n T i m e」属性および「m o d i f y T i m e」属性に割り当てる。これは、親t r a k要素2600の対応する属性に使用されるものと同じの値である。「S L C o n f i g D e s c r i p t o r」要素666から導出されるt i m e S c a l e値が、「t i m e S c a l e」属性に割り当てられる。親t r a k要素2600に割り当てられた持続時間値が、d u r a t i o n属性に割り当てられる。 40

【0436】

3. 動作3610で、標準XML手段を使用して、新しい「h d l r」要素2608を作成し、ステップ2で作成された新しい「m d i a」要素2604に挿入する。

【0437】

「h d l r」要素の次の属性に、値が割り当てられる：「h a n d l e r T y p e」および「n a m e」。「h a n d l e r T y p e」属性に割り当てられる値は、s t r e a m T y p eに依存する。s t r e a m T y p eが、1（o s d m）、3（s d s m）、4（ビジュアル・ストリーム）、または5（オーディオ・ストリーム）と等しい場合に、値「o d s m」、「s d s m」、「s o u n」、または「v i d e」が、「h a n d l e r T y p e」属性に割り当てられる。そうでない場合には、値「n o n e」が「h a n d l 50

erType」属性に割り当てられる。「name」属性に割り当てられる値は、囲むXMT-A ES_Descriptor要素180、190、または630のES_ID属性によって決定されるストリング「ES_DescriptorId」のコピーである。この「name」属性の選択は、必要ではないが、この選択によって、mp4文書および後続ファイルでES_ID属性ストリングの値を保持し、伝搬させることが可能になる。

【0438】

4. 動作3616で、標準XML手段を使用して、新しい「minf」要素2612を作成し、ステップ2で作成された新しい「mdia」要素2604に挿入する。

【0439】

5. 動作3620で、標準XML手段を使用して、新しい「dinf」要素2616を作成し、ステップ4で作成された新しい「minf」要素2612に挿入する。

【0440】

6. 動作3626で、標準XML手段を使用して、新しい「dref」要素2620を作成し、ステップ5で作成された新しい「dinf」要素2616に挿入する。

【0441】

7. 動作3630で、標準XML手段を使用して、新しい「urlData」要素2624を作成し、ステップ6で作成された新しい「dref」要素2620に挿入する。「1」の値を、「urlData」要素2624の「flags」属性に割り当てる。

【0442】

8. 動作3636で、標準XML手段を使用して、新しい「stbl」要素2628および2652を作成し、ステップ4で作成された新しい「minf」要素2612に挿入する。要素であるサンプル・テーブル要素の予備的な形が、下の「予備的サンプル・テーブル要素の作成」で説明するように作成される。

【0443】

9. 動作3640で、標準XML手段を使用して、新しいメディア・ヘッダ要素2632を作成し、ステップ4で作成した「minf」要素2612に挿入する。メディア・ヘッダ要素の要素名は、このストリームのstreamTypeプロパティに依存する。

【0444】

streamTypeプロパティが1(odsm)または3(sdsm)の場合に、メディア・ヘッダ要素は、属性なしの「nmhd」要素である。

【0445】

streamTypeが4(ビジュアル・ストリーム)である場合に、メディア・ヘッダ要素は、「0」の値を有する属性「transferMode」を有する「vmhd」要素である。

【0446】

streamTypeプロパティが5(オーディオ・ストリーム)である場合に、メディア・ヘッダ要素は、「0」の値を有する属性「balance」を有する「smhd」要素である。

【0447】

そうでない場合には、メディア・ヘッダ要素は、値「0」の属性「transferMode」および値「0」の属性「balance」を有する「gmhd」要素である。

【0448】

10. 動作3646で、streamTypeプロパティの値を値4および5と比較し、量startTimeの値を0と比較する。

【0449】

オーディオ・ストリームまたはビデオ・ストリームの場合に、この動作は、手順「XMT-A Body要素の処理(パス2)」3130中に実行される。この場合に、量「startTime」の値が、手順「XMT-A Body要素の処理(パス1)」3120で作成されたオブジェクト・テーブル(図88参照)から入手される。この値は、Ob

10

20

30

40

50

jectDescriptorID列の項目が、この「ES_Descriptor」要素を含む「ObjectDescriptor」要素の「objectDescriptorId」属性と一致する行のstartTime列の項目によって決定される。

【0450】

odsmおよびsdsmの場合に、この動作は、手順「XMT-A Header要素の処理」3116中に実行される。量startTimeの値を確立するのに使用されるオブジェクト・テーブルは、まだこのストリームについて作成されていない。その結果、odsmストリームのstartTime値に関する対応するテストが、別々のステップ「odsmのeditListの作成」3126の一部として実行される。sdsmは、必ず時刻0から始まる。

10

【0451】

streamTypeプロパティの値が4（ビジュアル・ストリーム）または5（オーディオ・ストリーム）であり、ストリームの量「startTime」の値が0でない場合に、動作3650で、標準XML手段を使用して、新しい「edts」（editList）要素2644を作成し、現在のtrak要素2600に挿入する。標準XML要素を使用して、新しい「elst」要素2648を作成し、新しい「edts」要素2644に挿入する。その後、2つの新しい「segment」要素を作成し、「elst」要素2648に挿入する。

【0452】

各「segment」要素は、「startTime」、「duration」、および「rate」という名前の属性を割り当てられる。値「-1」が、第1のsegment要素の「startTime」属性に割り当てられる。値「0」が、第2のsegment要素の「startTime」属性に割り当てられる。値「1.0」が、両方のsegment要素の「rate」属性に割り当てられる。第1のセグメントの「duration」属性の値は、このストリームのstartTime値と、これを囲むmoov要素の「timeScale」属性の値の積によって決定される値を割り当てられる。第2のセグメントの「duration」属性の値は、このストリームのduration値とこれを囲むmoov要素の「timeScale」属性の値の積によって決定される値を割り当てられる。このストリームのduration値は、XMT-A「Body」要素に対する第1パスで作成されたオブジェクト・テーブル（図88参照）から得られる「stopTime」値と「startTime」値の間の差によって決定される。これらの値は、ObjectDescriptorID列の項目が、この「ES_Descriptor」要素を含む「ObjectDescriptor」要素の「objectDescriptorId」属性と一致する行の対応する列の項目によって決定される。

20

30

【0453】

11. 動作3656で、streamTypeプロパティの値を「1」と比較する。streamTypeが1（odsm）である場合に、動作3660で、標準XML手段を使用して、新しい「tref」要素2636を作成し、ステップ1で作成された「trak」要素2600に挿入する。新しい「mpod」要素2640を作成し、新しい「tref」要素2636に挿入する。値「-1」を、「mpod」要素2640の「trackID」属性に割り当てる。これは、後に得られるデータによって置換される一時的な値である。

40

【0454】

このステップ3656で、手順「指定されたストリームのtrak要素の作成」3440が完了する。この手順に続いて、手順「ES_Descriptorの処理」3340は、テスト「指定されたストリームはsdsmまたはodsmか」3450で継続される。

【0455】

7. 1. 10. 3 予備的サンプル・テーブル要素の作成

最終的なmp4バイナリ・ファイル2230のサンプル・テーブルのそれぞれに、バイ

50

ナリ形式の `s d s m`、`o d s m`、およびメディア・データ・ファイルに依存する情報が含まれる。これらのテーブルの値を決定するのに必要な情報は、この時点では入手可能でない。その結果、図 85 に示されているように、これらのテーブルの予備的な表現を生成して、実際の `mp4` バイナリ・ファイル 2230 が作成される時に最終的な値が置かれる場所を示す。

【0456】

動作 3666 で、標準 XML 手段を使用して、新しい「`stsc`」要素 2656 を作成し、現在の「`trak`」要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。標準 XML 手段を使用して、新しい「`sampleToChunk`」要素を作成し、新しい「`stsc`」要素 2656 に挿入する。「1」の値を、新しい「`sampleToChunk`」要素の「`sampleDesc`」属性に割り当てる。「1」の値を、新しい「`sampleToChunk`」要素の「`firstChunk`」属性に割り当てる。このストリームの `streamType` プロパティが 1 または 3 (`odsm` または `s d s m`) である場合に、「0」の値を、新しい「`sampleToChunk`」要素の「`numSamples`」属性に割り当てる。このストリームの `objectType` プロパティが 108 (JPEG イメージ) である場合に、「1」の値を、新しい「`sampleToChunk`」要素の「`numSamples`」属性に割り当てる。そうでない場合には、「-1」の値を、新しい「`sampleToChunk`」要素の「`numSamples`」属性に割り当てる。

10

【0457】

動作 3670 で、標準 XML 手段を使用して、新しい「`stts`」要素 2660 を作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。現在の `streamType` プロパティが 1 (`odsm`) でも 3 (`s d s m`) でもない場合に、標準 XML 手段を使用して、新しい「`timeToSample`」要素を作成し、新しい「`stts`」要素 2660 に挿入する。「`trak`」要素で指定される持続時間値を、この「`timeToSample`」要素の「`duration`」属性に割り当てる。このストリームの `objectType` プロパティが 108 (JPEG イメージ) である場合に、「1」の値を、この「`timeToSample`」要素の「`numSamples`」属性に割り当てる。そうでない場合には、「-1」の値を、新しい「`timeToSample`」要素の「`numSamples`」属性に割り当てる。

20

30

【0458】

動作 3676 で、標準 XML 手段を使用して、新しい「`stco`」要素 2664 を作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。標準 XML 手段を使用して、新しい「`chunkOffset`」要素を作成し、新しい「`stco`」要素 2664 に挿入する。`nextTrackId` の現在値を、この「`chunkOffset`」要素の「`mdatId`」属性に割り当てる。「0」の値を、この「`chunkOffset`」要素の「`mdatOffset`」属性に割り当てる。「8」の値を、この「`chunkOffset`」要素の「`offset`」属性に割り当てる。

【0459】

動作 3680 で、標準 XML 手段を使用して、新しい「`stsz`」要素 2668 を作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。このストリームの `streamType` プロパティが 1 (`odsm`) でも 3 (`s d s m`) でもない場合に、新しい「`stsz`」要素 2668 の「`numSamples`」属性に値を割り当てる。このストリームの `objectType` プロパティが 108 (JPEG イメージ) である場合に、「1」の値を、新しい「`stsz`」要素 2668 の「`numSamples`」属性に割り当てる。そうでない場合に、「-1」の値を、新しい「`stsz`」要素 2668 の「`numSamples`」属性に割り当てる。

40

【0460】

動作 3686 で、`streamType` プロパティが 1 (`odsm`) または 3 (`s d s m`) である場合に、標準 XML 手段を使用して、新しい「`stss`」要素 (2672) を

50

作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。`streamType` プロパティが 1 である場合に、値「1」を、新しい「`stss`」要素 2672 の「`numEntries`」属性に割り当て、新しい「`syncSample`」要素を作成し、新しい「`stss`」要素 2672 に挿入する。値「0」を、この「`syncSample`」要素の「`sampleNumber`」属性に割り当てる。`streamType` プロパティが 3 の場合には、値「0」を、新しい「`stss`」要素 2672 の「`numEntries`」属性に割り当てる。

【0461】

`streamType` プロパティが 4 であり、`objectType` プロパティが 32 である (MPEG-4 ビデオ) 場合に、標準 XML 手段を使用して、新しい「`stss`」要素 2672 を作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入し、値「-1」を、新しい「`stss`」要素 2672 の「`numEntries`」属性に割り当てる。 10

【0462】

動作 3690 で、標準 XML 手段を使用して、新しい「`tsd`」要素 2676 を作成し、現在の `trak` 要素 2600 の「`stbl`」要素 2628 および 2652 に挿入する。新しい「`tsd`」要素 2676 内の従属要素を、図 86 に示されているように作成する。

【0463】

動作 3700 で、標準 XML 手段を使用して、新しい「`esds`」要素 2684 を作成する。 20

【0464】

動作 3706 で、`streamType` プロパティの値を「1」および「3」と比較する。

【0465】

`streamType` プロパティの値が「1」または「3」である場合に、動作 3710 で、標準 XML 手段を使用して、新しい「`mp4s`」要素 2680 を作成し、現在の「`tsd`」要素 2676 に挿入する。新しい「`esds`」要素 2684 を、新しい「`mp4s`」要素 2680 に挿入し、値「1」を、新しい「`mp4s`」要素 2680 の「`dataRefIndex`」属性に割り当てる。 30

【0466】

`streamType` プロパティの値が「1」でも「3」でもない場合に、動作 3716 で、`streamType` プロパティを「4」と比較する。

【0467】

`streamType` プロパティの値が「4」である場合に、動作 3720 で、標準 XML 手段を使用して、新しい「`mp4v`」要素 2680 を作成し、現在の「`tsd`」要素 2676 に挿入する。新しい「`esds`」要素 2684 を新しい「`mp4v`」要素 2680 に挿入し、値「-1」、「1」、「1」、「72.0」、「72.0」、「24」、「240」、および「320」を、それぞれ、新しい「`mp4v`」要素 2680 の属性「`colorTable`」、「`dataRefIndex`」、「`frameCount`」、「`horizontalRes`」、「`verticalRes`」、「`pixelDepth`」、「`height`」、および「`width`」に割り当てる。 40

【0468】

`streamType` プロパティの値が「4」でない場合に、動作 3726 で、`streamType` プロパティを「5」と比較する。`streamType` プロパティの値が「5」でない場合には、この手順 (動作 3690) が完了する。

【0469】

`streamType` プロパティの値が「5」である場合に、動作 3730 で、標準 XML 手段を使用して、新しい「`mp4a`」要素 2680 を作成し、現在の「`tsd`」要素 2676 に挿入する。新しい「`esds`」要素 2684 を新しい「`mp4a`」要素 26 50

80に挿入し、値「1」、「-1」、および「-1」を、それぞれ、新しい「mp4a」要素2680の属性「dataRefIndex」、「numChannels」、および「sampleSize」に割り当てる。

【0470】

追加のstreamTypeの事例を容易に処理することができるが、これらの事例（1、3、4、5）だけが、現在の実施形態で必要である。streamTypeプロパティの値が1、3、4、および5のどれでもない場合に、stsd要素2676についてそれ以上の処理は実行されない。

【0471】

動作3736で、streamTypeプロパティの値が、「1」、「3」、「4」、または「5」である場合に、標準XML手段を使用して、新しい「ES_Descr」要素2688を作成し、現在の「esds」要素2684に挿入する。値「0」を、新しい「ES_Descr」要素2688の「ES_ID」属性に割り当てる。値「0」を、新しい「ES_Descr」要素2688の「priority」属性に割り当てる。 10

【0472】

動作3740で、標準XML手段を使用して、新しい「DecoderConfigDescriptor」（D-C-D）要素2710を作成し、現在の「ES_Descr」要素2688に挿入する。「bufferSizeDB」属性、「avgBitrate」属性、および「maxBitrate」属性の値を、このストリームのXMT-A「DecoderConfigDescriptor」650から入手し、これらの属性の値を、新しい「DecoderConfigDescriptor」要素2710の「bufferSize」属性、「avgBitrate」属性、および「maxBitRate」属性に割り当てる。現在のストリームのstreamTypeプロパティおよびobjectTypeプロパティの値を、新しい「DecoderConfigDescriptor」要素2710の「streamType」属性および「objectType」属性に割り当てる。 20

【0473】

動作3746で、標準XML手段を使用して、新しい「SLConfigDescriptor」（SLC-D）要素2760を作成し、現在の「ES_Descr」要素2688に挿入する。値「2」を、新しい「SLConfigDescriptor」要素2760の「predefined」属性に割り当てる。 30

【0474】

streamTypeプロパティおよびobjectTypeプロパティの値に応じて、デコーダ固有情報要素を「DecoderConfigDescriptor」要素2710に挿入することができる。streamTypeプロパティの値が1（odsm）の場合に、デコーダ固有情報は不要である。

【0475】

動作3750で、streamTypeプロパティの値を「3」と比較する。

【0476】

streamTypeプロパティの値が3（sdsm）である場合に、動作3756で、手順「BIFS構成の処理」を実行する。この手順は、下で説明する。 40

【0477】

動作3760で、objectTypeプロパティの値を「32」と比較する。

【0478】

objectTypeプロパティの値が32（MPEG-4ビデオ）の場合に、動作3766で、標準XML手段を使用して、新しい「VisualConfig」要素2740を作成し、現在の「DecoderConfigDescriptor」要素2710に挿入する。

【0479】

動作3770で、objectTypeプロパティの値を「64」と比較する。 50

【0480】

objectTypeプロパティの値が64 (MP EG-4オーディオ)である場合に、動作3776で、標準XML手段を使用して、新しい「AudioConfig」要素を作成し、現在の「DecoderConfigDescriptor」要素2710に挿入する。

【0481】

動作3780で、objectTypeプロパティの値を「108」と比較する。

【0482】

objectTypeプロパティの値が108 (J P E G イメージ)である場合に、動作3786で、標準XML手段を使用して、「J P E G _ D e c o d e r C o n f i g」要素2730を作成し、現在の「DecoderConfigDescriptor」要素2710に挿入する。 10

【0483】

追加のstreamTypeおよびobjectTypeの事例を簡単に処理することができるが、上記の事例だけが、現在の実施形態に必要である。

【0484】

「B I F S _ D e c o d e r C o n f i g」要素2720を除いて、上で指定したデコーダ固有情報要素2730、2740、および2750は、単なるスタブまたはプレースホルダである。streamTypeプロパティの値が3 (s d s m)である場合に、下で説明する手順「B I F S 構成の処理」が実行される。そうでない場合に、E S _ D e s c r i p t o r 要素630の処理は、処理「t r a k 要素の作成」3440のステップ9 (動作3640)で継続される。 20

【0485】

7. 1. 10. 4 B I F S 構成の処理

手順「B I F S 構成の処理」を図87に示す。

【0486】

動作3800で、標準XML手段を使用して、新しい「B I F S _ D e c o d e r C o n f i g」要素2720を作成し、「DecoderConfigDescriptor」要素2710に挿入する。

【0487】

動作3810で、標準XML手段を使用して、mp4bifs文書2800内へ「b i f s C o n f i g」要素2810を入手する。 30

【0488】

動作3820で、標準XML手段を使用して、s d s mの「DecoderConfigDescriptor」要素650に従属する「decSpecificInfo」要素656および680を入手する。次に、標準XML手段を使用して、この「decSpecificInfo」要素680に従属する「B I F S C o n f i g」要素686を入手する。

【0489】

動作3830で、値「0」を「B I F S _ D e c o d e r C o n f i g」要素2720の「nodeIdBits」属性および「b i f s C o n f i g」要素2810の対応する属性に割り当てる。値「0」を、「B I F S _ D e c o d e r C o n f i g」要素2720の「routeIdBits」属性および「b i f s C o n f i g」要素2810の対応する属性に割り当てる。 40

【0490】

動作3840で、objectTypeプロパティの現在値を「2」と比較する。

【0491】

objectTypeプロパティの現在値が「2」である場合に、動作3846で、値「0」をB I F S _ D e c o d e r C o n f i g 要素2720の「protoIdBits」属性およびb i f s C o n f i g 要素2810の対応する属性に割り当て、B I F S 50

Config要素686の「use3DMeshCoding」および「usePredictiveMFField」属性によって決定される値を、BIFS_DecoderConfig要素2720およびbifsConfig要素2810の同一の名前の属性に割り当てる。

【0492】

動作3850で、標準XML手段を使用して、BIFSConfig要素(686)に従属するcommandStream要素690を入手する。

【0493】

BIFSConfig要素686に、従属するcommandStream要素690が含まれない場合に、動作3856で、標準XML手段を使用して、BIFSConfig要素686に従属するanimMask要素を入手する。BIFSConfig要素が、従属するanimMask要素を所有しない場合に、XMT-A文書は無効であり、動作3860でエラーを報告する。

【0494】

BIFSConfig要素686が従属するanimMask要素を所有する場合に、動作3866で、値「偽」を、BIFS_DecoderConfig要素2720のcommandStream属性に割り当てる。

【0495】

BIFSConfig要素686が、従属するcommandStream要素690を所有する場合に、動作3870で、値「真」を、BIFS_DecoderConfig要素2720のcommandStream属性に割り当てる。commandStream要素690のpixelMetric属性の値を、BIFS_DecoderConfig要素2720のpixelMetric属性に割り当てる。commandStream要素690のpixelMetric属性の値が指定されていない場合には、「偽」のデフォルト値を、BIFS_DecoderConfig要素2720のpixelMetric属性に割り当てる。

【0496】

動作3880で、標準XML手段を使用して、commandStream要素690に従属する「size」要素696を入手する。

【0497】

commandStream要素690が従属するsize要素696を所有しない場合に、動作3886で、値「0」を、BIFS_DecoderConfig要素2720の「pixelHeight」および「pixelWidth」属性に割り当てる。

【0498】

commandStream要素690が、従属するsize要素696を所有する場合に、動作3890で、「size」要素696の「pixelHeight」および「pixelWidth」属性の値を、BIFS_DecoderConfig要素2720の「pixelHeight」および「pixelWidth」属性に割り当てる。

【0499】

これらのステップが完了した後に、ES_Descriptor要素640の処理は、処理「trak要素の作成」3440のステップ9(動作3640)で継続される。

【0500】

7. 2 中間XML文書に基づくmp4バイナリ・ファイルの作成

中間XML文書2250および2260の作成の後に、中間XML文書2250および2260ならびに関連するメディア・データ・ファイル2220を使用して、オリジナルのXMT-A文書2210で指定された情報を表す新しいmp4バイナリ・ファイル2230を作成する。この新しいmp4ファイルを、「出力mp4ファイル」または「mp4ファイル」と呼ぶ。この新しいmp4ファイルを作成するのに使用される手段は、次の6つのステップからなる。

1. 入力文書および出力宛先の確立

10

20

30

40

50

2. 作業配列の作成
3. 「m d a t」要素 2 3 1 0 の処理
4. 「m o o v」要素 2 3 2 0 の処理
5. 任意選択のユーザ・データ要素 2 3 4 0 の処理
6. o d s mバッファ・サイズの更新

【0501】

これらのステップのそれぞれを、下で説明する。

【0502】

7. 2. 1 入力文書および出力宛先の確立

これらのステップの最初のステップは、上で作成されたm p 4 f i l e文書 2 2 5 0 およびm p 4 b i f s文書 2 2 6 0を表すXMLデータ構造への参照を得ることからなる。このステップには、出力m p 4 バイナリ・ファイル 2 2 3 0のファイル名を指定するデータ構造を受け取ることも含まれる。指定されたファイル名が既存のファイルに対応する場合に、そのファイルは削除される。その後、新しい空の出力ファイルが、指定されたファイル名を使用して作成される。

【0503】

空の出力ファイルを作成した後に、標準XML手段を使用して、m p 4 f i l e文書のトップ・レベル要素を入手する。m p 4 b i f s文書のトップ・レベル要素もこの時点で入手することができるが、これは、後になるまで必要でない。

【0504】

新しい出力ファイル（「m p 4ファイル」）は、「m p 4アトム」および「m p 4オブジェクト構造」と呼ばれるデータ構造の階層セットからなる。現在の実施形態では、各m p 4アトムが、32ビット「s i z e」値、32ビット「アトムID」、および1組のプロパティ値からなる。m p 4アトムに、1つまたは複数の従属するm p 4アトムまたはm p 4オブジェクト構造を含めることもできる。サイズ値は、サイズおよびアトムIDを含む完全なm p 4アトムのバイト数を指定する。m p 4オブジェクト構造は、1バイトのオブジェクト構造タグ、可変サイズのサイズ値、1組のプロパティ値、および1組の0個以上の従属するm p 4オブジェクト構造からなる。この場合に、サイズ値は、オブジェクト構造タグおよびサイズ値を除くオブジェクト構造のバイト数を指定する。

【0505】

各アトムを生成する全般的な手順を、図103に示す。オブジェクト構造を作成する、対応する手順を、図104に示す。これらの手順は、出力m p 4ファイルの「ファイル位置」を制御する能力を必要とする。「ファイル位置」は、ファイルの先頭から、次のバイトが書き込まれる点までのバイト数と定義される。ファイル位置を制御する必要のゆえに、この新しいファイルは、「ランダム・アクセス」または「読み取り／書き込み」のタイプのファイルとしてオープンされなければならない。

【0506】

7. 2. 1. 1 m p 4アトム作成の処理

m p 4アトムを作成する処理は、次のステップからなる。

【0507】

1. 動作5000で、出力ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。量「s i z e P o s」の値は、各m p 4アトムまたはオブジェクト構造に一意である。

【0508】

2. 動作5010で、値0の32ビット整数を出力ファイルに書き込む。

【0509】

3. 動作5020で、32ビット・アトムID値を出力ファイルに書き込む。たとえば、「m d a t」アトムの場合に、文字「m」、「d」、「a」、および「t」のa s c i i値を表す4バイトを、出力ファイルに書き込む。

【0510】

10

20

30

40

50

4. 動作5030で、現在のmp4アトムによって表されるmp4file要素の属性を解釈する。各mp4アトムによって所有される属性の特定の組は、MP4ファイル・フォーマットに関するMPEG-4仕様に示されているように、アトムIDによって決定される。mp4file文書で指定されない属性については、デフォルト値を提供する。

【0511】

5. 動作5040で、現在のmp4アトムの属性の値を出力ファイルに書き込む。各属性値を表すのに使用されるビット数は、MP4ファイル・フォーマットに関するMPEG-4仕様に示されている。

【0512】

6. 動作5050で、現在のmp4file要素が従属要素を所有する場合に、各そのような従属要素を処理する。従属要素が、mp4fileアトム要素に対応する場合に、現在の手順を再帰的に繰り返す。従属要素が、mp4fileオブジェクト要素に対応する場合に、図104に示された手順を実行する。 10

【0513】

7. 動作5060で、現在のファイル位置を量「endPos」に割り当てる。

【0514】

8. 動作5070で、量「endPos」の値と量「sizePos」の値の間の差を、量「size」に割り当てる。

【0515】

9. 動作5080で、出力ファイルのファイル位置を、量「sizePos」の値によって示される位置に変更する。 20

【0516】

10. 動作5090で、量「size」の値を表す32ビット整数を出力ファイルに書き込む。

【0517】

11. 動作5095で、出力ファイルのファイル位置を、量「endPos」の値によって指定される位置に変更する。

【0518】

7. 2. 1. 2 mp4オブジェクト構造の作成の処理

mp4オブジェクト構造を作成する処理は、次のステップからなる。 30

【0519】

1. 動作5100で、1バイトのオブジェクト構造タグを出力ファイルに書き込む。オブジェクト構造タグの値は、mp4オブジェクト構造によって表されるmp4file要素の要素名と、MPEG-4仕様で提供されるテーブルによって決定される。

【0520】

2. 動作5110で、出力ファイルの現在ファイル位置を、量「sizePos」に割り当てる。量「sizePos」の値は、各mp4アトムまたはオブジェクト構造に一意である。

【0521】

3. 動作5120で、mp4オブジェクト構造を表すのに必要なバイト数の推定値または上限に基づいて、量「numSizeBytes」に値を割り当てる。mp4オブジェクト構造を表すのに必要なバイト数が128未満の場合には、値「1」を量「numSizeBytes」に割り当てる。ほとんどの場合に、これで十分である。 40

【0522】

4. 動作5130で、1バイト値のシーケンスを出力ファイルに書き込む。この1バイト値の個数は、量「numSizeBytes」の値によって指定される。この1バイト量の値は、後で上書きされるので、未決定である。このバイトのそれぞれに値0を使用することができる。

【0523】

5. 動作5135で、現在のmp4オブジェクト要素によって表されるmp4file 50

要素の属性を解釈する。各mp4オブジェクト要素によって所有される属性の特定の組は、MPEG-4 systems仕様に示されているように、オブジェクト・タグによって決定される。mp4file文書で指定されていない属性については、デフォルト値を提供する。

【0524】

6. 動作5140で、現在のmp4オブジェクト構造の属性の値を出力ファイルに書き込む。各属性値を表すのに使用されるビット数は、MPEG-4 systems仕様に示されている。

【0525】

7. 動作5150で、現在のmp4file要素が従属要素を所有する場合に、各そのような従属要素を、図104に示された手順に従って処理する（再帰的に）。 10

【0526】

8. 動作5160で、現在のファイル位置を量「endPos」に割り当てる。

【0527】

9. 動作5165で、量「endPos」の値と量「sizePos」の値の間の差を、量「size」に割り当てる。

【0528】

10. 動作5170で、量「numSizeBytes」の値を、量「size」の値から引く。

【0529】

11. 動作5180で、出力ファイルのファイル位置を、量「sizePos」の値によって指定される位置に変更する。 20

【0530】

12. 動作5190で、量「size」の値を表す1バイト値のシーケンスを出力ファイルに書き込む。この1バイト値の個数は、量「numSizeBytes」の値によって指定される。この1バイト値のそれぞれの下位7ビットは、量「size」の値の対応する7ビット部分によって決定される。この1バイト値のそれぞれの上位ビットの値は、最後の1バイト値を除いて「1」である。このシーケンスの最後の1バイトの上位ビットの値は、「0」である。

【0531】

13. 動作5195で、出力ファイルのファイル位置を、量「endPos」の値によって指定される位置に変更する。 30

【0532】

7. 2. 2 作業配列の作成

上で識別された第2ステップは、mp4file文書2250および2300で表される、「trak」要素2350の個数、チャンク要素2450および2490の個数、ならびにodsmSample要素2510の個数に基づく、複数の作業配列の作成からなる。

【0533】

次の手段が、量「MaxNumTracks」の値を決定するのに使用される。 40

【0534】

標準XML手段を使用して、mp4file文書2300のトップ・レベル要素に従属するすべての要素2310および2320を識別する。これらの従属要素の1つが、「moov」要素2320になる。この「moov」要素2320の「nextTrackID」属性の値が、「moov」要素2320に従属する「trak」要素2350の個数の上限を与える。mp4file文書が、上で示したように作成される場合に、「nextTrackID」属性の値は、「moov」要素2320に従属する「trak」要素2350の個数を指定する。「nextTrackID」属性の値を、量「MaxNumTracks」に割り当てる。

【0535】

次の9つのリストが、これらのリストのそれぞれの項目数を指定するために、量「MaxNumTracks」の値を使用して作成される。

1. MediaSamples
2. MediaDataFile (「File」オブジェクトの配列)
3. MediaHeaderSize
4. MediaHeader (位置値の配列)
5. EsDescrSize
6. TrackIdForTrack
7. StreamTypeForTrack
8. ObjectTypeForTrack
9. TrackIdforOdId

10

【0536】

MediaDataFileリストおよびMediaHeaderリストを除く、これらのリストのそれぞれが、整数の配列である。

【0537】

9つのリストのこの組を作成した後に、値0を、量「TrackNum」、「MaxNumChunks」、および「MaxNumOdsmSamples」に割り当てる。

【0538】

次の手段を使用して、リストTrackIdForTrack、StreamTypeForTrack、およびObjectTypeForTrackの項目を決定する。

20

【0539】

標準XML手段を使用して、「moov」要素2320に従属するすべての要素を識別する。型「trak」2350のこのような従属要素ごとに、「trackID」属性の値をTrackIdForTrackリストの項目TrackNumに割り当てる。標準XML手段を使用して、この「trak」要素に従属する(9レベルまでの)「DecoderConfigDescriptor」要素2710を識別する。この要素2710の「streamType」属性の値を、リストStreamTypeForTrackの項目TrackNumに割り当てる。この要素2710の「objectType」属性の値を、リストObjectTypeForTrackの項目TrackNumに割り当てる。その後、量TrackNumの値を1つ増分する。

30

【0540】

次の手段を使用して、量「MaxNumChunks」および「MaxNumOdsmSamples」の値を決定する。

【0541】

標準XML手段を使用して、mp4file文書2300のトップ・レベル要素に従属するすべての「mdat」要素2310を識別する。標準XML手段を使用して、各「mdat」要素2310および2400に従属するすべての要素を識別する。結果の従属要素に、「mediaFile」要素2430、「sdsm」要素2410、および「odsm」要素2420が含まれる可能性がある。標準XML手段を使用して、各「mdat」要素2400に従属する要素2410、2420、および2430のそれぞれに従属する各「チャンク」要素2450および2490ならびに「odsmChunk」要素2470を識別する。

40

【0542】

量MaxNumChunksの値を、各「mdat」要素2400に従属する要素2410、2420、および2430のそれぞれに従属する「チャンク」要素2450および2490ごとにならびに「odsmChunk」要素2470ごとに1つ増分する。

【0543】

標準XML手段を使用して、各「odsmChunk」要素2470および2500に従属する各「odsmSample」要素2510を識別する。量MaxNumOdsmSamplesの値を、各「odsmChunk」要素2500に従属する「odsmS

50

amp l e」要素 2 5 1 0 ごとに 1 つ増分する。

【0 5 4 4】

次の 4 つのリストを、これらのリストのそれぞれの項目数を指定するために、量「M a x N u m C h u n k s」の値を使用して作成する。

1. M d a t I d F o r C h u n k
2. T r a c k I d F o r C h u n k
3. O f f s e t F o r C h u n k
4. M e d i a D a t a S i z e

【0 5 4 5】

これらのリストのそれぞれが、整数の配列である。これらのリストを作成した後に、値 10
0 を量 N u m C h u n k s に割り当てる。

【0 5 4 6】

量「M a x N u m O d s m S a m p l e s」の値が、0 を超える場合に、次の 2 つのリストを、これらのリストのそれぞれの項目数を指定するために、量「M a x N u m O d s m S a m p l e s」の値を使用して作成する。

1. O d s m S a m p l e S i z e
2. O d s m S a m p l e T i m e

【0 5 4 7】

これらのリストのそれぞれが、整数の配列である。これらのリストを作成した後に、値 20
0 を量 N u m O d s m S a m p l e s に割り当てる。

【0 5 4 8】

7. 2. 3 「m d a t」要素の処理

出力 m p 4 ファイル 2 2 3 0 の作成の第 3 ステップは、m p 4 f i l e 文書 2 3 0 0 に含まれる m d a t 要素 2 3 1 0 のそれぞれを処理することが含まれる。

【0 5 4 9】

標準 XML 手段を使用して、図 5 9 に示された、m p 4 f i l e 文書 2 3 0 0 のトップ・レベル要素に従属する各 m d a t 要素 2 3 1 0 を識別する。この「m d a t」要素 2 3 1 0 のそれぞれを、図 1 0 5 に示された手段を使用して処理する。図 1 0 5 に示された手順は、図 1 0 3 に示された処理の例である。

【0 5 5 0】

動作 5 2 0 0 で、出力 m p 4 ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作 5 2 1 2 で、値 0 の 3 2 ビット整数を出力 m p 4 ファイル 7 2 4 に書き込む。動作 5 2 2 4 で、文字「m」、「d」、「a」、および「t」に対応する A S C I I 値を表す 4 バイトを、出力 m p 4 ファイル 7 3 0 に書き込む。この m d a t 要素の「m d a t I d」属性の値を、量「m d a t I d」に割り当てる。プロパティ値は、出力 m p 4 ファイルに書き込まれない。

【0 5 5 1】

動作 5 2 3 6 で、値 0 をインデックス「i」に割り当てる。動作 5 2 4 2 で、インデックス「i」の値を量「n u m M d a t C h i l d r e n」の値と比較するが、量「n u m M d a t C h i l d r e n」は、現在の m d a t 要素によって所有される従属要素の個数 40
を示す。インデックス「i」の値が量「n u m M d a t C h i l d r e n」の値と等しい場合に、動作 5 2 4 8 で、m d a t アトム サイズ 7 2 4 を、図 1 0 3 の最後の 5 つの部分（動作 5 0 6 0 から 5 0 9 5）に示されているように更新する。

【0 5 5 2】

インデックス「i」の値が量「n u m M d a t C h i l d r e n」の値と等しくない場合に、動作 5 2 5 4 で、標準 XML 手段を使用して、現在の m d a t 要素に従属する各 XML 要素を入手する。現在の m d a t 要素に従属する i 番目の XML 要素を、「m d a t C h i l d」によって表し、要素 m d a t C h i l d の要素名を、「c h i l d N a m e」によって表す。

【0 5 5 3】

動作5260で、XML要素mdatChildの名前を、ストリング「mediaFile」と比較する。XML要素mdatChildの名前がストリング「mediaFile」と一致する場合に、動作5266で、手順「メディア・ファイル・データの挿入」を実行する。手順「メディア・ファイル・データの挿入」を実行した後に、インデックス「i」の値を1つ増分し5296、量「numMdatChildren」の値とのインデックス「i」の値の比較動作5242を繰り返す。

【0554】

XML要素mdatChildの名前がストリング「mediaFile」と一致しない場合に、動作5272で、XML要素mdatChildの名前をストリング「odsm」と比較する。XML要素mdatChildの名前がストリング「odsm」と一致する場合に、動作5278で、手順「Odsmデータの挿入」を実行する。手順「Odsmデータの挿入」を実行した後に、インデックス「i」の値を1つ増分し動作5296で、量「numMdatChildren」の値とのインデックス「i」の値の比較動作5242を繰り返す。

【0555】

XML要素mdatChildの名前がストリング「odsm」と一致しない場合に、動作5284で、XML要素mdatChildの名前をストリング「sdsm」と比較する。XML要素mdatChildの名前がストリング「sdsm」と一致する場合に、動作5290で、手順「Sdsmデータの挿入」を実行する。手順「Sdsmデータの挿入」を実行した後に、インデックス「i」の値を1つ増分し動作5296で、量「numMdatChildren」の値とのインデックス「i」の値の比較動作5242を繰り返す。

【0556】

XML要素mdatChildの名前がストリング「sdsm」と一致しない場合に、動作5296でインデックス「i」の値を1つ増分し、量「numMdatChildren」の値とのインデックス「i」の値の比較動作5242を繰り返す。

【0557】

7. 2. 3. 1 メディア・ファイル・データの挿入

図106の手順「メディア・ファイル・データの挿入」5226は、「mdat」要素2400に従属する「mediaFile」要素2430を処理するのに使用される。動作5300で、「mediaFile」要素2430の「trackId」属性の値を、量「trackId」に割り当てる。動作5306で、「mediaFile」要素2430の「name」属性の値を、量「mediaFileName」に割り当てる。

【0558】

動作5312で、量「trackId」の値と一致するTrackIdForTrackリストの項目のインデックスによって、量「trackNum」の値を決定する。動作5318で、リストStreamTypeForTrackおよびリストObjectTypeForTrackの対応する項目（インデックスtrackNumを有する）の値を、量「streamType」および「objectType」に割り当てる。

【0559】

動作5324で、mediaFileName量の値によって識別されるメディア・データ・ファイルの新しい「File」オブジェクトを作成する。動作5330で、このオブジェクトを、量trackNumの値によって決定されるインデックスを有する、MediaDataFileリストの項目として保管する。動作5336で、このメディア・データ・ファイルを含むバイト数と定義されるこのメディア・データ・ファイルのサイズを、この新しいFileオブジェクトのlengthプロパティとして入手する。このサイズ値を、量「mediaFileSize」に割り当てる。動作5342で、量「MediaHeaderSize」の値を、0に初期化する。

【0560】

動作5348で、値0をインデックス「i」に割り当てる。動作5354で、インデッ

クス「i」の値を、量「numMediaFileChildren」の値と比較するが、値を、量「numMediaFileChildren」の値は、現在のmediaFile要素2430に従属するXML要素の個数によって決定される。

【0561】

インデックス「i」の値が量「numMediaFileChildren」の値と等しい場合に、動作5360で、メディア・データ・ファイル内のサンプルの個数を数える。メディア・データ・ファイル内のサンプルを数えるのに使用される手段は、「streamType」および「objectType」の値と、メディア・データ・ファイルの各特定のタイプの詳細なファイル構造詳細に依存する。これらの手段は、本発明に固有ではなく、本明細書では提示しない。動作5366で、メディア・データ・ファイル内のサンプルの個数を数えた後に、結果のサンプル・カウントを、量trackNumの値によって決定されるインデックスを有する、MediaSamplesリストの項目として保管する。

10

【0562】

量「i」の値が量「numMediaFileChildren」の値と等しくない場合に、動作5372で、標準XML手段を使用して、現在のmediaFile要素2480に従属する各XML要素を入手する。現在のmediaFile要素に従属するi番目のXML要素を、「mediaFileChild」によって表し、要素mediaFileChildの要素名を、「childName」によって表す。

【0563】

動作5384で、XML要素mediaFileChildの名前をストリング「chunk」と比較する。

20

【0564】

XML要素mediaFileChildの名前がストリング「chunk」と一致する場合に、動作5390で、手順「メディア・データ・チャンクの挿入」を実行する。手順「メディア・データ・チャンクの挿入」を実行した後に、動作5396でインデックス「i」の値を1つ増分し、量「numMediaFileChildren」の値とのインデックス「i」の値の比較の動作5354を繰り返す。

【0565】

XML要素mediaFileChildの名前がストリング「chunk」と一致しない場合に、動作5396でインデックス「i」の値を1つ増分し、量「numMediaFileChildren」の値とのインデックス「i」の値の比較の動作5354を繰り返す。

30

【0566】

7. 2. 3. 2 メディア・データ・チャンクの挿入

手順「メディア・データ・チャンクの挿入」5390は、主に、メディア・データ・ファイル2220の内容を出力mp4ファイル2230に付加することからなる。量「streamType」および「objectType」の値によって決定されるあるタイプのメディア・データは、最初の「ヘッダ」データ・セクションから始まるものとすることができる。これには、「MPEG-4ビデオ」(streamType=4かつobjectType=32)が含まれる。特定のタイプのメディア・データのヘッダ・データ・セクションを識別するのに必要な正確な手段は、メディア・データ・ファイルの各タイプの詳細な仕様に依存する。これらのファイル仕様は、本発明の範囲の外であり、本明細書には含まれない。MPEG-4ビデオ・ストリームの説明については、ISO/IEC文書14496-2(1999年、2000年修正)「情報テクノロジーオーディオ・ビジュアル・オブジェクトのコーディング第2部：ビジュアル(Information Technology-Coding of Audio-Visual Objects - Part 2: Visual)」を参照されたい。

40

【0567】

メディア・データをメディア・データ・ファイルからmp4バイナリ・ファイルにコピーする前に、次の動作が実行される。

50

【0568】

1. 動作5230で決定される量「mdatId」の値を、リスト「MdatIdForChunk」の項目「NumChunks」に割り当てる。

【0569】

2. 動作5300で決定される量「trackId」の値を、リスト「TrackIdForChunk」の項目「NumChunks」に割り当てる。

【0570】

3. 出力mp4ファイルの現在のファイル位置の値を、リスト「OffsetForChunk」の項目「NumChunks」に割り当てる。

【0571】

4. 動作5336で決定される量「mediaFileSize」の値を、リスト「MediaDataSize」の項目「NumChunks」に割り当てる。

【0572】

5. 値0を、リスト「MediaHeaderSize」の項目「trackNum」に割り当てる。

【0573】

6. 量「streamType」および「objectType」の値によって指定されるメディア・ファイル・タイプに、最初のヘッダ・データ・セクションが含まれる場合に、このヘッダ・セクションを含むバイト数を、リスト「MediaHeaderSize」の項目「trackNum」に割り当てる。このサイズのバイト配列を作成し、メディア・ヘッダ・セクション内のデータを、メディア・データ・ファイルからこの配列にコピーする。このバイト配列の位置の値を、リスト「MediaHeader」の項目「trackNum」に割り当てる。

【0574】

7. メディア・データの残りを、（入力）メディア・データ・ファイル2220から出力mp4バイナリ・ファイル2230および730にコピーする。

【0575】

必要な場合に、この段階でデータにデータ・フォーマット変換を適用することができる。たとえば、MPEG-2オーディオ・データ（streamType=5かつobjectType=64）を、MPEG-4オーディオ・ストリームの要件を満たすように変更することができる。これらの変更は、MPEG-2 Advanced Audio Coding（AAC）データの詳細な仕様に依存する〔ISO/IEC文書13818-7（1997年）「情報テクノロジー動画および関連するオーディオ情報の包括的コーディング第7部：高度なオーディオ・コーディング（Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 7: Advanced Audio Coding）」〕。これらの仕様および関連するデータ変換は、本発明の範囲の外であり、本明細書には含まれない。

【0576】

8. 量「numChunks」の値を1つ増分する。

【0577】

7. 2. 3. 3 odsMデータの挿入

手順「odsMデータの挿入」5278は、「mdat」要素2400に従属する「odsM」要素2420を処理するのに使用される。この手順は、現在のodsM要素2460に従属する「odsMChunk」要素2470ごとに、出力mp4ファイルに新しいチャンク736を作る。

【0578】

「odsM」要素2420の「trackId」属性の値を、量「trackId」に割り当てる。標準XML手段を使用して、「odsM」要素2420および2460に従属する各「odsMChunk」要素2470を入手する。

【0579】

10

20

30

40

50

現在の「odsm」要素2460に従属する「odsmChunk」要素2470ごとに、次の動作が実行される。

【0580】

1. 動作5230で決定された量「mdatId」の値を、リスト「MdatIdForChunk」の項目「NumChunks」に割り当てる。

【0581】

2. 量「trackId」の値を、リスト「TrackIdForChunk」の項目「NumChunks」に割り当てる。

【0582】

3. 出力mp4ファイルの現在のファイル位置の値を、リスト「OffsetForChunk」の項目「NumChunks」に割り当てる。 10

【0583】

4. 値「-1」を、リスト「MediaDataSize」の項目「NumChunks」に割り当てる。

【0584】

5. 量「numChunks」の値を1つ増分する。

【0585】

6. 標準XML手段を使用して、「odsmChunk」要素2500に従属する各「odsmSample」要素2510を入手する。

【0586】

ステップ6で識別された「odsmSample」要素2510ごとに、現在のmp4ファイル位置を量「sampleStart」に割り当て、「size」属性の値を量「sampleSize」に割り当て、「time」属性の値を量「sampleTime」に割り当てる。量「sampleTime」の値を、リスト「OdsmSampleTime」の項目numOdsmSamplesに割り当てる。「sampleSize」の値は、結果のバイナリodsmサンプルの推定値として扱われる。これは、最終的なファイル位置と「sampleStart」の値との間の差によって決定される正確な値に置換される。 20

【0587】

標準XML手段を使用して、「odsmSample」要素2520に従属する各XML要素2530を入手する。これらの従属要素は、「ObjectDescrUpdate」2540または「ObjectDescrRemove」2570の要素名を有すると期待される。これらの事例のそれぞれが、下で示すように処理される。 30

【0588】

「odsmSample」要素2520に従属するすべてのXML要素2530の処理が完了した後に、出力mp4ファイルの結果のファイル位置と量「sampleStart」の間の差が、量「sampleSize」に割り当てられる（対応する属性値から導出された推定値と置換される）。この値を、リスト「OdsmSampleSize」の項目「numOdsmSamples」に割り当てる。その後、量「numOdsmSamples」の値を1つ増分する。 40

【0589】

7. 2. 3. 4 ObjectDescrUpdate要素

「odsmSample」要素2520に従属する「ObjectDescrUpdate」要素2540ごとに、図104に示された手順を使用して、出力mp4ファイル内にObjectDescrUpdateオブジェクト構造2000を作成する。構造タグ「ObjectDescrUpdateTag」（値=1）2010を、動作5100で、8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を量「sizePos」に割り当て、「sizePos」の値を、量「filePos1」に割り当てる。動作5120で、値「1」を、量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的なサイズ値2020 50

として出力mp4ファイルに書き込む。

【0590】

「ObjectDescrUpdate」要素2540は、属性を有しないので、動作5135および5140では何も行わない。

【0591】

標準XML手段を使用して、「ObjectDescrUpdate」要素2540に従属する各XML要素2550を入手する。これらの従属要素は、「ObjectDescriptor」2550の要素名を有すると期待される。下の動作5150に記載されているように各従属「ObjectDescriptor」要素2550を処理した後に、ObjectDescrUpdate構造のサイズ2020を、図104（動作5160から5195）に示されているように更新する。 10

【0592】

「ObjectDescrUpdate」要素2540に従属する「ObjectDescriptor」要素2550ごとに、図104に示された手順を使用して、出力mp4ファイル内でObjectDescriptorsオブジェクト構造2030および2100を作成する。動作5100で、構造タグ「MP4_OD_Tag」（値=17）2108を、8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当て、「sizePos」の値を、量「filePos2」に割り当てる。動作5120で、値「1」を量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的サイズ値2116として出力mp4ファイルに書き込む。 20

【0593】

「ObjectDescriptor」要素2550の「OdId」属性の値を、量「OdId」に割り当てる。量「OdId」の数値に、64をかけ（左に6ビットだけシフトする）、値「31」を結果に足して、量「OdId」の変更された値を決定する。値「31」は、ObjectDescriptorオブジェクト構造2100内の「予約済みフィールド」2140を表す。

【0594】

「ObjectDescriptor」要素2550の「url」属性が指定されている場合に、値「32」を、変更された「OdId」値に加算する。結果の値を、16ビット整数としてmp4ファイルに書き込む。「url」属性の値の文字数を示す1バイトを、mp4ファイルに書き込む。次に、「url」属性の値を、文字のシーケンスとしてmp4ファイルに書き込む。 30

【0595】

「ObjectDescriptor」要素2550の「url」属性が指定されていない場合には、変更された「OdId」値を、16ビット整数2124、2132、および2140としてmp4ファイルに書き込む。

【0596】

次に、標準XML手段を使用して、「ObjectDescriptor」要素2550に従属する各XML要素2560を入手する。これらの従属要素は、「EsIdRef」という名前の要素2560を有すると期待される。 40

【0597】

現在の「ObjectDescriptor」要素2550に従属する「EsIdRef」要素2560ごとに、図104に示された手順を使用して、出力mp4ファイル内でEsIdRefオブジェクト構造2148および2160を作成する。動作5100で、構造タグ「EsIdRefTag」（値=15）2170を、8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5120で、値「1」を、量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的サイズ値2180として出力mp4ファイルに書き込む。 50

【0598】

動作5135で、「E s I d R e f」要素2560の「E s I d」属性の値を、量「E s I d」に割り当てる。次に、動作5140で、量「E s I d」の数値を、16ビット整数2190として出力mp4ファイルに書き込む。E s I d R e f要素2560は、従属要素を有しない5150。

【0599】

「E s I d」値2190の処理の後に、E s I d R e fオブジェクト構造のサイズ2180を、図104（動作5160から5195）に示されているように更新する。

【0600】

「O b j e c t D e s c r i p t o r」要素2550の処理の後に、f i l e P o s 2 10
の値を量「s i z e P o s」に割り当て、M P 4 _ O Dオブジェクト構造2100のサイズ2116を、図104（動作5160から5195）に示されているように更新する。

【0601】

「O b j e c t D e s c r U p d a t e」要素2540の処理の後に、f i l e P o s 1
の値を、量「s i z e P o s」に割り当て、O b j e c t D e s c r U p d a t eオブジェクト構造2000のサイズ2020を、図104（動作5160から5196）に示されているように更新する。

【0602】

7. 2. 3. 5 O b j e c t D e s c r R e m o v e 要素

現在の「o d s m S a m p l e」要素2520に従属する「O b j e c t D e s c r R 20
e m o v e」要素2570ごとに、図104に示された手順を使用して、出力mp4ファイル内にO b j e c t D e s c r R e m o v eオブジェクト構造2040を作成する。動作5100で、構造タグ「O b j e c t D e s c r R e m o v e T a g」（値=2）2050を、8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を、量「s i z e P o s」に割り当て、「s i z e P o s」の値を、量「f i l e P o s 1」に割り当てる。動作5120で、値「1」を量「n u m S i z e B y t e s」に割り当てる。動作5130で、値0を、予備的サイズ値2060として出力mp4ファイルに書き込む。

【0603】

「O b j e c t D e s c r R e m o v e」要素2570の「O d I d」属性の値を、量 30
「O d I d L i s t」に割り当てる。

【0604】

量「O d I d L i s t」は、「ホワイト・スペース」（空白スペースおよび他の非数字文字）によって区切られた1つまたは複数の整数を表す文字ストリングからなる。「O d I d L i s t」内の数字文字の各シーケンスは、整数として解釈され、結果の値が、10ビットのo b j e c t D e s c r i p t o r I d値2070としてmp4ファイルに書き込まれる。mp4ファイルに書き込まれる連続するo b j e c t D e s c r i p t o r I d値2070は、バイト整列されない。o b j e c t D e s c r i p t o r I d値2070のシーケンスによって占められる総バイト数（n B i t s）が、8の倍数でない場合には、n P a d個の0ビット2080がmp4ファイルに書き込まれ、n P a dの値は、n 40
B i t s m o d u l o 8によって与えられる。

【0605】

「O d I d L i s t」量の処理の後に、O b j e c t D e s c r R e m o v eオブジェクト構造2040のサイズ2060が、図104（動作5160から5195）に示されているように更新される。

【0606】

7. 2. 3. 6 s d s mデータの挿入

手順「S d s mデータの挿入」は、「m d a t」要素2400に従属する「s d s m」要素2410を処理するのに使用される。「s d s m」要素2410および2440の「t r a c k I d」属性の値を、量「t r a c k I d」に割り当てる。任意選択の属性「x 50

mlFile」を設けることができる。この属性は、mp4bifs文書2800を表す入力XMLファイルの名前を指定するのに使用することができる。その代わりに、XMT-A文書に基づくmp4file文書およびmp4bifs文書の作成に関して上で説明した処理など、別の処理の結果としてmp4bifs文書2800を入手することができる。その後、標準XML手段を使用して、mp4bifs文書2800のトップ・レベル要素を入手する。

【0607】

図71からわかるように、mp4bifs文書2800は、単一の従属する「bifsConfig」要素2810および1つまたは複数の従属する「commandFrame」要素2820を有するmp4bifストップレベル要素からなる。各「commandFrame」要素2820は、scene description stream (sds)の「サンプル」を表す。mp4bifs文書2800の解釈の準備で、sdsサンプルの数を、mp4bifストップ要素2800に従属する「commandFrame」要素2820の個数を数えることによって決定することができる。結果の値を、量「MaxNumSdsSamples」に割り当て、それぞれがMaxNumSdsSamples個の項目を有する2つのリストを作成する。このリストの一方「SdsSampleSize」は、整数値のリストである。第2のリスト「SdsSampleTime」は、浮動小数点値、好ましくは倍精度浮動小数点値（項目あたり64ビット）のリストである。値0を、量「NumSamples」に割り当てる。

【0608】

標準XML手段を使用して、「sds」要素2440に従属する各「チャンク」要素2450を入手する。「sds」要素2440ごとに、多くとも1つの従属する「チャンク」要素2450が期待される。次の動作が、「チャンク」要素2450に対して実行される。

【0609】

1. 動作5230で決定された量「mdatId」の値を、リスト「MdatIdForChunk」の項目「NumChunks」に割り当てる。

【0610】

2. 量「trackId」の値を、リスト「TrackIdForChunk」の項目「NumChunks」に割り当てる。

【0611】

3. 現在のファイル位置の値を、リスト「OffsetForChunk」の項目「NumChunks」に割り当てる。

【0612】

4. 値「-1」を、リスト「MediaDataSize」の項目「NumChunks」に割り当てる。

【0613】

5. 量「numChunks」の値を1つ増分する。

【0614】

次に、mp4bifs文書2800を、下で説明するように解釈する。この文書が解釈される時に、データ値が、出力mp4ファイル700に書き込まれ、値が、リスト「SdsSampleSize」および「SdsSampleTime」に入力される。オブジェクト指向実施形態では、これが、新しいSdsEncoderオブジェクトを作成し、このオブジェクトのメソッド「encodeSds」を呼び出すことによって達成される。このメソッドは、完成したリスト「SdsSampleSize」および「SdsSampleTime」を返し、sdsのバイナリ・エンコーディングを表すデータを出力mp4ファイル700に付加する。

【0615】

標準XML手段を使用して、mp4bifストップ・レベル要素2800に従属する「bifsConfig」要素2810を入手する。この要素の「routeIdBits

」属性の値を、量「RouteIdBits」に割り当てる。この要素の「nodeIdBits」属性の値を、量「NodeIdBits」に割り当てる。2の「NodeIdBits」乗（または左にNodeIdBitsだけシフトされた「1」）の値を、量MaxUpdateableNodesに割り当てる。整数の2つの新しいリスト「UpdateableNodeId」および「UpdateableNodeNumber」を作成する。これらのリストのそれぞれの項目数は、「MaxUpdateableNodes」の値によって決定される。値0を、量「NumUpdateableNodes」に割り当てる。値「偽」を、boolean量「bUseNames」に割り当てる。

【0616】

次に、標準XML手段を使用して、mp4bifsトップ・レベル要素2800に従属する各「commandFrame」要素2820を入手する。次の手段を使用して、各そのような「commandFrame」要素2820を処理する。

【0617】

1. 出力mp4ファイルの現在のファイル位置の値を、量「FilePointerAtStart」に割り当てる。

【0618】

2. 「commandFrame」要素2820の「time」属性の値を、量「Time」に割り当てる。量「Time」の値を、リスト「SdsmsSampleTime」の項目「NumSamples」に割り当てる。

【0619】

3. 標準XML手段を使用して、「commandFrame」要素2830に従属するbifsCommand要素2840のそれぞれを入手する。そのような従属要素のそれぞれを、下で説明するように処理する。

【0620】

4. 出力mp4ファイルの現在のファイル位置の値を、量「FilePointerAtEnd」に割り当て、量「FilePointerAtEnd」の値と量「FilePointerAtStart」の値の間の差を、リスト「SdsmsSampleSize」の項目「NumSamples」に割り当てる。

【0621】

5. 量「NumSamples」の値を1つ増分する。

【0622】

図72からわかるように、各「commandFrame」要素2830に、1つまたは複数の従属するbifsCommand要素2840が含まれる。各bifsCommand要素2910は、sdsmsデータでエンコードできる11個の可能なBIFSコマンドの1つを表す。これには、3つの挿入コマンド（「InsertNode」、「InsertRoute」、「InsertIndexedValue」）、3つの削除コマンド（「DeleteNode」、「DeleteRoute」、「DeleteIndexedValue」）、および5つの置換コマンド（「ReplaceNode」、「ReplaceRoute」、「ReplaceIndexedValue」、「ReplaceField」、および「ReplaceScene」）が含まれる。図73からわかるように、BIFSコマンド要素2910は、BIFSノードを表す従属要素2920を有することができる。図74からわかるように、ReplaceSceneコマンドを表すbifsCommand要素2930に、1つまたは複数のRoute要素2960を含む単一の従属するRoutes要素2950も含めることができる。

【0623】

特定の「commandFrame」要素2830に従属するbifsCommand要素2840のバイナリ表現を生成する前に、各従属bifsCommand要素2910を「スキャン」して、「NodeId」属性3010の値が指定されているすべての従属するNode要素2920および3000を識別する。この「スキャン」動作は、標準XML手段を使用して、現在の「commandFrame」要素2830に従属するb

10

20

30

40

50

i f s C o m m a n d 要素 2 8 4 0 のそれぞれを入手することによって達成される。この動作は、1つまたは複数の従属する B I F S N o d e 要素 2 9 2 0 および 2 9 4 0 を含む可能性がある6つの b i f s C o m m a n d 要素 (「I n s e r t N o d e」、「I n s e r t I n d e x e d V a l u e」、「R e p l a c e N o d e」、「R e p l a c e I n d e x e d V a l u e」、「R e p l a c e F i e l d」、および「R e p l a c e S c e n e」) 2 9 1 0 だけに適用される。

【0624】

この「スキャン」動作を実行するのに使用される手順は、出力 m p 4 ファイルに対して何も行われないことと、「n o d e I d」属性およびフィールド・データ型「n o d e」または「c o m m a n d b u f f e r」を有する属性以外のすべての属性が無視されることとを除いて、下で説明する後続 B I F S コマンド解釈手順に使用されるものと同等である。「N o d e I d」属性が指定されたノードごとに、「N o d e I d」属性の値を、リスト「U p d a t e a b l e N o d e I d」の項目「n u m U p d a t e a b l e N o d e s」に割り当てる。このノードの「n o d e n u m b e r」プロパティの値を、「U p d a t e a b l e N o d e N u m b e r」リストの対応する項目に割り当てる。特定のノード要素の「n o d e n u m b e r」プロパティは、このノード要素の要素名と一致する、ノード名のテーブル内の項目のインデックスによって決定される。ノード名のテーブルは、M P E G - 4 S y s t e m s 仕様書で定義されている。その後、量「N u m U p d a t e a b l e N o d e s」の値を1つ増分する。

【0625】

従属する b i f s C o m m a n d 要素 2 8 4 0 をスキャンした後に、標準XML手段を再度使用して、現在の「c o m m a n d F r a m e」要素 2 8 3 0 に従属する各XML要素 2 8 4 0 を入手する。図 2 8 からわかるように、各 B I F S コマンド 1 1 2 0 に、「継続」ビット 1 1 3 0 および 1 1 4 0 が続く。「c o m m a n d F r a m e」要素 2 8 3 0 に従属する各 b i f s C o m m a n d 要素 2 8 4 0 を処理する前に (最初のそのようなコマンドを除いて)、値「1」を有する単一のビットを出力 m p 4 ファイルに書き込んで、「継続=1」1 1 3 0 を指定する。従属する b i f s C o m m a n d 要素 2 8 4 0 のそれぞれを、下で説明するように処理する。すべての従属要素を処理した後に、値「0」を有する単一のビットを出力 m p 4 ファイルに書き込んで、「継続=0」(コマンド・フレームの終り) 1 1 4 0 を指定する。バイナリ形式でこのコマンド・フレームを表すのに使用されるビットの総数が、8の倍数でない場合には、最後のバイトを0 1 1 5 0 でパディングして、ビットの総数を8の倍数にする。

【0626】

次に、次の手段を使用して、現在の c o m m a n d F r a m e 要素 2 8 3 0 に従属する b i f s C o m m a n d 要素 2 8 4 0 ごとに、出力 m p 4 ファイルにバイナリ B I F S コマンド・データ構造を付加する。

【0627】

b i f s C o m m a n d 要素 2 8 4 0 が、3つの挿入コマンドのうちの1つを表す場合に、2ビットの挿入コード (バイナリ値=00) 1 2 0 6 が、出力 m p 4 ファイルに書き込まれる。B I F S C o m m a n d 要素 2 8 4 0 が、3つの削除コマンドのうちの1つを表す場合に、2ビットの削除コード (バイナリ値=01) 1 2 2 0 が、出力 m p 4 ファイルに書き込まれる。B I F S C o m m a n d 要素 2 8 4 0 が、4つの置換コマンドのうちの1つ (R e p l a c e S c e n e 以外) を表す場合に、2ビットの置換コード (バイナリ値=10) 1 2 4 0 が、出力 m p 4 ファイルに書き込まれる。B I F S C o m m a n d 要素 2 8 4 0 が、R e p l a c e S c e n e コマンドを表す場合に、2ビットの s c e n e 置換コード (バイナリ値=11) 1 2 8 0 が、出力 m p 4 ファイルに書き込まれる。

【0628】

7. 2. 3. 7 ノード挿入 B I F S コマンド

「I n s e r t N o d e」b i f s C o m m a n d 要素 2 8 4 0 および 2 9 1 0 の場合に、図 3 3 に示された N o d e I n s e r t i o n B I F S コマンド 1 3 0 0 を、出

力mp 4ファイルに付加する。「ノード」の2ビットのパラメータ型コード（バイナリ値＝00）1308を、挿入コード1304に続いて出力mp 4ファイルに書き込む。InsertNode要素の「parentId」属性の値を、量「nodeID」に割り当て、この量1312の整数値を、mp 4ファイルに書き込む。量「nodeID」の値をエンコードするのに使用されるビット数は、量「nodeIdBits」の値によって指定される。InsertNode要素の「parentID」属性の値は、UpdateableNodeIDリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、従属するNode要素の親ノードの「node number」プロパティの値を指定する。

【0629】

InsertNode要素の「insertionPosition」属性の値を、量「insertionPosition」に割り当て、この量1316の整数値を表す2ビットを、mp 4ファイルに書き込む。量「insertionPosition」の値が0である場合に、InsertNode要素の「position」属性の値を、量「position」に割り当て、この量1320の整数値を表す8ビットを、mp 4ファイルに書き込む。

【0630】

各InsertNode要素に、従属するNode要素2920が含まれる。このNode要素のバイナリSFNode表現1324を、挿入位置1316および1320を表すデータに続いて、出力mp 4ファイルに付加する。SFNode構造のフォーマットは、図44から48に示されている。このSFNode構造を作成するのに使用される手順を、下で説明する。

【0631】

7. 2. 3. 8 Indexed Value Insertion BIFSコマンド

「InsertIndexedValue」bifsCommand要素2840および2910の場合に、図34に示されたIndexedValue Insertion BIFSコマンド1328を、出力mp 4ファイルに付加する。「indexed value」の2ビットのパラメータ型コード（バイナリ＝10）1336を、挿入コード1332に続いてmp 4ファイルに書き込む。InsertIndexedValue要素の「nodeId」属性の値を、量「nodeID」に割り当て、この量1340の整数値をmp 4ファイルに書き込む。量「nodeID」の値のエンコードに使用されるビット数は、量「nodeIdBits」の値によって指定される。

【0632】

InsertIndexedValue要素の「nodeId」属性の値は、UpdateableNodeIDリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、このBIFSコマンドに関連するフィールド値によって変更されるBIFSノードの「node number」プロパティの値を指定する。

【0633】

このBIFSコマンドの「field index」プロパティの値は、InsertIndexedValue要素の「inFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのinFieldIDプロパティの値は、ノード番号の値、フィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、ノード番号の値、inFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。次に、量inFieldIDの整数値を、numBits個のビットを使用してmp 4ファイルに書き込む1344。

10

20

30

40

50

【0634】

InsertIndexedValue要素の「insertionPosition」属性の値を、量「insertionPosition」に割り当て、この量の整数値を表す2ビットを、mp4ファイルに書き込む1348。量「insertionPosition」の値が0である場合に、InsertIndexedValue要素の「position」属性の値を、量「position」に割り当て、この量の整数値を表す16ビットを、mp4ファイルに書き込む1352。

【0635】

各InsertIndexedValue要素に、「value」属性が含まれる。この値属性の解釈は、InsertIndexedValue要素のinFieldName属性によって識別されるプロパティ・フィールドの「フィールド・データ型」プロパティに依存する。フィールド・データ型プロパティは、ノード番号の値、フィールド・インデックス、およびMPEG-4 Systems仕様で定義されるテーブルの組によって決定される。フィールド・データ型プロパティが「SFNode」である場合に、「value」属性の値が、従属するNode要素の名前を指定する。そうでない場合に、「value」属性の値は、「フィールド値」の値を直接に指定する。どちらの場合でも、下の「SFField構造」で説明する手段を使用して、値属性を解釈し、値属性によって指定されるフィールド値のバイナリ表現を作成し、その結果を出力mp4ファイルに付加する1356。

【0636】

7.2.3.9 Route Insertion BIFSコマンド

「InsertRoute」bifsCommand要素2840および2910の場合に、図35に示されたRoute Insertion BIFSコマンド1360を、出力mp4ファイルに付加する。「route」の2ビットのパラメータ型コード（バイナリ値=11）1368を、挿入コード1364に続けてmp4ファイルに書き込む。InsertRoute要素の「routeId」属性の値が指定されていない場合に、値「0」を有する単一ビットを、「isUpdateable」値1372として出力mp4ファイルに書き込む。そうでない場合には、値「1」を、「isUpdateable」値1372として出力mp4ファイルに書き込み、これに続けて、InsertRoute要素の「routeId」属性によって指定される値1376を書き込む。routeId属性の整数値を表すのに使用されるビット数は、量RouteIdBitsの値によって指定される。

【0637】

InsertRoute要素の「departureNode」属性の値を、量「departureNodeId」に割り当て、この値をmp4ファイルに書き込む1380。量「departureNodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0638】

InsertRoute要素の「departureNode」属性の値は、UpdateableNodeIdリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、「departure node」の「node number」プロパティの値を指定する。

【0639】

departure nodeの「field index」プロパティの値は、InsertRoute要素の「departureFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのdepartureFieldIDプロパティの値は、departure nodeのノード番号の値、departure nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの

組によって決定される。量「numBits」の値は、departure nodeのノード番号の値、departureFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量departureFieldIDの値を、numBits個のビットを使用してmp4ファイルに書き込む1384。

【0640】

InsertRoute要素の「arrivalNode」属性の値を、量「arrivalNodeID」に割り当て、この値をmp4ファイルに書き込む1388。量「arrivalNodeID」の整数値を表すのに使用されるビット数は、量「NodeIDBits」の値によって指定される。

10

【0641】

InsertRoute要素の「arrivalNode」属性の値は、UpdateableNodeIDリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、「arrival node」の「node number」プロパティの値を指定する。

【0642】

arrival nodeの「field index」プロパティの値は、InsertRoute要素の「arrivalFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのarrivalFieldIDプロパティの値は、arrival nodeのノード番号の値、arrival nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、arrival nodeのノード番号の値、arrivalFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量arrivalFieldIDの整数値を、numBits個のビットを使用してmp4ファイルに書き込む1392。

20

【0643】

7. 2. 3. 10 Node Deletionコマンド

「DeleteNode」bifsCommand要素2840および2910の場合に、図36に示されたNode Deletion BIFSコマンド1400を、出力mp4ファイルに付加する。この場合に、「node」の2ビットのパラメータ型コード（バイナリ値=00）1412を、削除コード1406に続けてmp4ファイルに書き込む。「nodeId」属性の値を、量「nodeId」に割り当て、この値を表す整数を、mp4ファイルに書き込む1418。量「nodeId」の整数値を表すのに使用されるビット数は、量「NodeIDBits」の値によって指定される。

30

【0644】

7. 2. 3. 11 Indexed Value Deletion BIFSコマンド

「DeleteIndexedValue」bifsCommand要素2840および2910の場合に、図37に示されたIndexed Value Deletion BIFSコマンド1424を、出力mp4ファイルに付加する。「indexed value」の2ビットのパラメータ型コード（バイナリ値=10）1436を、削除コード1430に続けてmp4ファイルに書き込む。「nodeId」属性の値を、量「nodeId」に割り当て、この値を、mp4ファイルに書き込む1442。量「nodeId」の整数値を表すのに使用されるビット数は、量「nodeIDBits」の値によって指定される。

40

【0645】

DeleteIndexedValue要素の「nodeId」属性の値は、UpdateableNodeIDリストの項目の1つと一致しなければならない。Update

50

ableNodeNumberリストの対応する項目は、このBIFSコマンドに関連するBIFSノードの「node number」プロパティの値を指定する。

【0646】

このBIFSコマンドの「field index」プロパティの値は、DeleteIndexedValue要素の「inFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのinFieldIDプロパティの値は、ノード番号の値、フィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、ノード番号の値、inFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量inFieldIDの整数値を、numBits個のビットを使用してmp4ファイルに書き込む1448。

【0647】

DeleteIndexedValue要素の「deletion Position」属性の値を、量「deletionPosition」に割り当て、この量の整数値を表す2ビットを、mp4ファイルに書き込む1454。量「deletionPosition」の値が0である場合に、DeleteIndexedValue要素の「position」属性の値を、量「position」に割り当て、この量の整数値を表す16ビットを、mp4ファイルに書き込む1460。

【0648】

7.2.3.12 Route Deletion BIFSコマンド

「DeleteRoute」bifsCommand要素2840および2910の場合に、図38に示されたRoute Deletion BIFSコマンド1466を、出力mp4ファイルに付加する。「route」の2ビットのパラメータ型コード（バイナリ値=11）1478を、削除コード1472に続けてmp4ファイルに書き込む。DeleteRoute要素の「routeId」属性の値を、量「routeId」に割り当て、この値の整数表現を、mp4ファイルに書き込む1484。量「routeId」の整数値を表すのに使用されるビット数は、量「RouteIdBits」の値によって指定される。

【0649】

7.2.3.13 Node Replacement BIFSコマンド

「ReplaceNode」bifsCommand要素2840および2910の場合に、図39に示されたNode Replacementコマンド1500を、出力mp4ファイルに付加する。「node」の2ビットのパラメータ型コード（バイナリ値=00）1508を、置換コード1504に続けてmp4ファイルに書き込む。ReplaceNode要素の「nodeId」属性の値を、量「nodeId」に割り当て、この値をmp4ファイルに書き込む1510。量「nodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0650】

各ReplaceNode要素に、従属するNode要素2920が含まれる。このNode要素のバイナリSFNode表現1514を、nodeID値1510を表すデータに続けて、出力mp4ファイルに付加する。SFNode構造のフォーマットは、図44から48に示されている。このSFNode構造を作成するのに使用される手順を、下で説明する。

【0651】

7.2.3.14 Field Replacement BIFSコマンド

「ReplaceField」bifsCommand要素2840および2910の場合に、図40に示されたField Replacement BIFSコマンド1520を、出力mp4ファイルに付加する。「field」の2ビットのパラメータ型コー

10

20

30

40

50

ド（バイナリ値＝01）1528を、置換コード1524に続けてmp4ファイルに書き込む。ReplaceField要素の「nodeId」属性の値を、量「nodeId」に割り当て、この値をmp4ファイルに書き込む1530。量「nodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0652】

ReplaceField要素の「nodeId」属性の値は、UpdateableNodeIdリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、このBIFSコマンドに関連するフィールド値によって変更されるBIFSノードの「node number」プロパティの値を指定する。 10

【0653】

このBIFSコマンドの「field index」プロパティの値は、ReplaceField要素の「inFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのinFieldIDプロパティの値は、ノード番号の値、フィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、ノード番号の値、inFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量inFieldIDの値を、numBits個のビットを使用してmp4ファイルに書き込む1534。 20

【0654】

各ReplaceField要素に、「value」属性が含まれる。この値属性の解釈は、ReplaceField要素のinFieldName属性によって識別されるプロパティ・フィールドの「フィールド・データ型」プロパティに依存する。フィールド・データ型プロパティは、ノード番号の値、フィールド・インデックス、およびMPEG-4 Systems仕様で定義されるテーブルの組によって決定される。フィールド・データ型プロパティが「SFNode」である場合に、「value」属性の値が、従属するNode要素の名前を指定する。そうでない場合に、「value」属性の値は、「フィールド値」の値を直接に指定する。どちらの場合でも、下の「SFField構造」で説明する手段を使用して、値属性を解釈し、値属性によって指定されるフィールド値のバイナリ表現を作成し、その結果を出力mp4ファイルに付加する1538。 30

【0655】

7.2.3.15 Indexed Value Replacement BIFS コマンド

「ReplaceIndexedValue」bifsCommand要素2840および2910の場合に、図41に示されたIndexed Value Replacement BIFSコマンド1540を、出力mp4ファイルに付加する。「indexed value」の2ビットのパラメータ型コード（バイナリ値＝10）1548を、置換コード1544に続けてmp4ファイルに書き込む。「nodeId」属性の値を、量「nodeId」に割り当て、この値をmp4ファイルに書き込む1550。量「nodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。 40

【0656】

ReplaceIndexedValue要素の「nodeId」属性の値は、UpdateableNodeIdリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、このBIFSコマンドに関連するフィールド値によって変更されるBIFSノードの「node number」プロパティの値を指定する。 50

【0657】

このBIFSコマンドの「field index」プロパティの値は、ReplaceIndexedValue要素の「inFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのinFieldIDプロパティの値は、ノード番号の値、フィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、ノード番号の値、inFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量inFieldIDの値を、numBits個のビットを使用してmp4ファイルに書き込む1554。

【0658】

InsertIndexedValue要素の「replacementPosition」属性の値を、量「replacementPosition」に割り当て、この量の値を表す2ビットを、mp4ファイルに書き込む1558。量「replacementPosition」の値が0である場合には、ReplaceIndexedValue要素の「position」属性の値を、量「position」に割り当て、この量の整数値を表す16ビットを、mp4ファイルに書き込む1560。

【0659】

各ReplaceIndexedValue要素に、「value」属性が含まれる。この値属性の解釈は、ReplaceIndexedValue要素のinFieldName属性によって識別されるプロパティ・フィールドの「フィールド・データ型」プロパティに依存する。フィールド・データ型プロパティは、ノード番号の値、フィールド・インデックス、およびMPEG-4 Systems仕様で定義されるテーブルの組によって決定される。フィールド・データ型プロパティが「SFNode」である場合に、「value」属性の値が、従属するNode要素の名前を指定する。そうでない場合に、「value」属性の値は、「フィールド値」の値を直接に指定する。どちらの場合でも、下の「SFField構造」で説明する手段を使用して、値属性を解釈し、値属性によって指定されるフィールド値のバイナリ表現を作成し、その結果を出力mp4ファイルに付加する1564。

【0660】

7.2.3.16 Route Replacement BIFSコマンド

「ReplaceRoute」bifsCommand要素2840および2910の場合に、図42に示されたRoute Replacement BIFSコマンド1570を、出力mp4ファイルに付加する。「route」の2ビットのパラメータ型コード（バイナリ値=11）1578を、置換コード1574に続けてmp4ファイルに書き込む。ReplaceRoute要素の「routeId」属性の値を、量「routeId」に割り当て、この値の整数表現を、mp4ファイルに書き込む1580。量「routeId」の整数値を表すのに使用されるビット数は、量「RouteIdBits」の値によって指定される。

【0661】

ReplaceRoute要素の「departureNode」属性の値を、量「departureNodeId」に割り当て、この値をmp4ファイルに書き込む1584。量「departureNodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0662】

ReplaceRoute要素の「departureNode」属性の値は、UpdatableNodeIdリストの項目の1つと一致しなければならない。UpdatableNodeNumberリストの対応する項目は、「departure node」の「node number」プロパティの値を指定する。

【0663】

departure nodeの「field index」プロパティの値は、ReplaceRoute要素の「departureFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのdepartureFieldIDプロパティの値は、departure nodeのノード番号の値、departure nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、departure nodeのノード番号の値、departureFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量departureFieldIDの値を、numBits個のビットを使用してmp4ファイルに書き込む1588。

【0664】

ReplaceRoute要素の「arrivalNode」属性の値を、量「arrivalNodeID」に割り当て、この値をmp4ファイルに書き込む1590。量「arrivalNodeID」の整数値を表すのに使用されるビット数は、量「NodeIDBits」の値によって指定される。

【0665】

ReplaceRoute要素の「arrivalNode」属性の値は、UpdateableNodeIDリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、「arrival node」の「node number」プロパティの値を指定する。

【0666】

arrival nodeの「field index」プロパティの値は、ReplaceRoute要素の「arrivalFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのarrivalFieldIDプロパティの値は、arrival nodeのノード番号の値、arrival nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、arrival nodeのノード番号の値、arrivalFieldIDの値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量arrivalFieldIDの整数値を、numBits個のビットを使用してmp4ファイルに書き込む1594。

【0667】

7.2.3.17 Scene Replacement BIFSコマンド

「ReplaceScene」bifsCommand要素2930の場合に、シーン置換BIFSコマンド1270を、出力mp4ファイルに付加する。図32からわかるように、シーン置換BIFSコマンド1270は、2ビットの置換コード（バイナリ値＝11）1280と、それに続くBIFSシーン・データ構造1290および1600からなる。BIFSシーン構造1600の構成要素を、図43に示す。

【0668】

2ビットのシーン置換コード1280を書き込んだ後に、6ビットの0の値（「予約済み」）1610を出力mp4ファイルに書き込む。

【0669】

ReplaceScene要素の「USENAMES」属性の値は、boolean量bUseNamesの値を決定するのに使用される。「USENAMES」属性の値が「真」である場合に、値「真」を「bUseNames」に割り当て、単一の「1」ビットをmp4ファイルに書き込む1620。そうでない場合には、「真」をbUseName

sに割り当て、単一の「0」ビットをmp4ファイルに書き込む1620。次に、単一の「0」ビットをmp4ファイルに書き込んで、このmp4ファイルに「protoList」1630がないことを示す。

【0670】

protoListビット1630に、「SFTopNode」構造が続く。これは、MPEG-4 Systems仕様で定義されたノードのサブセットのメンバだけが許容される点を除いて、下で説明する「SFNode」構造と同等である。SFTopNode構造の記述は、ReplaceSceneコマンド要素2930に従属するmp4bifsNode要素2940によって指定される。

【0671】

必要な従属Node要素のほかに、mp4bifs ReplaceScene要素2930は、単一の従属する「Routes」要素2950も有することができる。ReplaceScene要素2930が、従属する「Routes」要素2950を有しない場合に、単一の「0」ビットが、「hasRoutes」ビット1650としてmp4ファイルに書き込まれ、これによって、BIFSシーン置換コマンド1270の終りが示される。ReplaceSceneコマンド要素2930が、従属する「Routes」要素2950を有する場合には、単一の「1」ビットが、「hasRoutes」ビット1650としてmp4ファイルに書き込まれ、これに、図49から51に記載のRoutes構造1660が続く。

【0672】

Routes構造1660は、2つの形すなわち、図49に示されたリスト形式1800または図50に示されたベクトル形式1830のいずれかを有することができる。これらは、リスト形式の「1」1805またはベクトル形式の「0」1835である第1ビットの値によって区別される。本発明の1実施形態では、必ずリスト形式が選択される。その結果、「hasRoutes」1650に「1」がセットされている場合に、次のビット1805にも「1」がセットされる。リスト形式対ベクトル形式の選択は、重要でなく、本発明は、Routes構造のベクトル形式1830を同等によく使用することができる。

【0673】

mp4bifs「Routes」要素2950は、1つまたは複数の従属する「Route」要素2960を有することができる。ある「Routes」要素2950に従属する「Route」要素2960ごとに、単一の「1」ビット1805および1815が、出力mp4ファイルに書き込まれ、それにRoute要素2960のバイナリ記述1810および1860が続く。単一の「0」ビット1820が、「Routes」要素2950に従属する最後の「Route」要素2960の記述の後に出力mp4ファイルに書き込まれる。最初の従属する「Route」要素2960のバイナリ記述1810の前の「1」ビット1805は、バイナリRoutesデータ構造2950の「リスト形式」1800が使用されていることを指定する。後続の「1」ビット1815は、「moreRoutes=真」を指定する。最後の「0」ビット1820は、「moreRoutes=偽」であることを指定し、Routes構造1800および2950の終りを示す。

【0674】

各Route要素のバイナリ記述1860の構造を、図51に示す。各Route要素のバイナリ記述1860を作成するのに使用される手段を、下の「Route構造」で説明する。

【0675】

7. 2. 3. 18 Route構造

バイナリRoute構造1860を、ReplaceScene要素2930に従属するRoutes要素2950に従属するRoute要素2960ごとに出力mp4ファイルに付加する。Route要素2960の「routeId」属性の値が指定されていない場合には、値「0」を有する単一のビットを、「isUpdatable」値186

10

20

30

40

50

5として出力mp4ファイルに書き込む。そうでない場合には、値「1」を「isUpdateable」値1865として出力mp4ファイルに書き込み、これに、Route要素2960の「routeId」属性によって指定される値1870を続ける。routeId属性の整数値を表すのに使用されるビット数は、量RouteIdBitsの値によって指定される。

【0676】

Route要素2960の「routeId」属性の値が指定されており、対応するReplaceScene要素2930のUSENAMES属性の値が「真」である場合に、Route要素2960の「name」属性の値が、ヌル終端ストリング1875として出力mp4ファイルに付加される。Route要素2960の「name」属性の値は、対応するXMT-A ROUTE要素390のDEF属性のコピーである。 10

【0677】

Route要素の「toNode」属性の値を、量「outNodeId」に割り当て、この値をmp4ファイルに書き込む1880。量「outNodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0678】

Route要素の「outNodeId」属性の値は、UpdateableNodeIdリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、「departure node」の「node number」プロパティの値を指定する。 20

【0679】

departure nodeの「field index」プロパティの値は、Route要素の「toFieldName」属性の値と一致する値を有する、対応するノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのoutFieldRefプロパティの値は、departure nodeのノード番号の値、departure nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値は、departure nodeのノード番号の値、outFieldRefプロパティの値、およびMPEG-4 Systems仕様で定義されたテーブル 30
によって決定される。その後、量outFieldRefの値を、numBits個のビットを使用してmp4ファイルに書き込む1885。

【0680】

Route要素2960の「fromNode」属性の値を、量「inNodeId」に割り当て、この値をmp4ファイルに書き込む1890。量「inNodeId」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。

【0681】

Route要素2960の「inNodeId」属性の値は、UpdateableNodeIdリストの項目の1つと一致しなければならない。UpdateableNodeNumberリストの対応する項目は、「arrival node」の「node number」プロパティの値を指定する。 40

【0682】

arrival nodeの「field index」プロパティの値は、Route要素の「fromFieldName」属性の値と一致する値を有する、このノード番号のフィールド名のリストの項目のインデックスによって決定される。このフィールド名のリストは、MPEG-4 Systems仕様で定義されている。このフィールドのinFieldRefプロパティの値は、arrival nodeのノード番号の値、arrival nodeのフィールド・インデックスの値、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。量「numBits」の値 50

は、`arrival node`のノード番号の値、`inFieldRef`の値、およびMPEG-4 Systems仕様で定義されたテーブルによって決定される。その後、量`inFieldRef`の整数値を、`numBits`個のビットを使用してmp4ファイルに書き込む1895。

【0683】

7.2.3.19 SFNode構造

`mp4bifs Node`要素3000、3040、および3080は、`Insert Node`要素、`ReplaceNode`要素、`ReplaceScene`要素、または別の`mp4bifs Node`要素3000の従属要素として現れることができる。各`mp4bifs Node`要素3000、3040、および3080は、図75、図76、または図77に示された構造を有する。100個を超えるタイプの`mp4bifs Node`要素がある。これらのそれぞれが、MPEG-4 Systems仕様で定義されたBIFSノードの1つに対応する。各BIFSノードは、指定されたノード名（文字ストリング）およびプロパティ・フィールドの順序付きの組を有する。各プロパティ・フィールドは、指定された名前（文字ストリング）、指定されたデータ型（`boolean`、`integer`、`float`など）、および他の特性を有する。BIFSノードの各タイプは、同一の名前の`mp4bifs Node`要素によって表され、BIFSノードの各プロパティ・フィールドは、対応する`mp4bifs Node`要素の同一の名前の属性によって表される。

10

【0684】

`boolean`、`integer`、`float`、`color`、および`string`を含むほとんどのBIFSノード・プロパティ・フィールド・データ型について、関連するデータ値は、`mp4bifs Node`要素の対応する属性の値によって表される。2つの例外すなわち、フィールド・データ型「`node`」および「`buffer`」がある。フィールド・データ型「`node`」および「`buffer`」の場合に、関連するデータ値は、従属する`mp4bifs`要素3030および3070によって表され、対応する属性に、関連する従属要素の名前の順序付きリストが含まれる。この名前の順序付きリストは、複数の属性が`node`または`buffer`のフィールド・データ型を有する場合に`mp4bifs Node`要素の各属性に関連する特定の従属要素を決定するのに使用することができる。

20

30

【0685】

BIFSノードの各タイプに一意のプロパティ・フィールドのほかに、すべてのBIFSノードは、1組の共通プロパティを有する。これには、再利用状態、更新可能状態、およびマスク・アクセス状態が含まれる。プロパティ・フィールドおよび共通プロパティの結果の組合せを、図44から48に示す。

【0686】

次の手段を使用して、`mp4bifs Node`要素によって表されるバイナリSFNode構造を作成する。

【0687】

第1ステップは、`mp4bifs Node`要素の任意選択の「`nodeRef`」属性の存在を検査することである。この属性の値が指定されている場合に、この属性の値を、量「`nodeIDref`」に割り当てる。この場合に、そのノードは、「再利用される」ノードとして分類され、結果のBIFSノードは、図44に示された構造を有する。この場合に、値「1」を有する単一のビット1704を、mp4ファイルに書き込む。このビットは、条件「`isReused=真`」を指定する。このビットに、量「`nodeIDref`」1708の値が続く。量「`nodeIDref`」の値を表すのに使用されるビット数は、量「`NodeIDBits`」の値によって与えられる。従属要素または他の属性は、この場合には許容されない。

40

【0688】

`mp4bifs Node`要素の「`nodeRef`」属性の値が指定されていない場合

50

に、単一の「0」ビット1712および1732をmp4ファイルに書き込む。このビットは、条件「isReused=偽」を指定する。結果のBIFS SFNodeは、図45（マスク・ノード）1710または図46（リスト・ノード）1730に示された構造を有することができる。本発明のこの実施形態では、必ずマスク・ノード形式1710が選択される。リスト・ノードではなくマスク・ノードを選択することは、重要でなく、本発明は、SFNodeのリスト・ノード形式1730を使用して同等によく実施することができる。

【0689】

次に、mp4bifs Nodeノード要素3000の要素名(NodeName)を、MPEG-4 Systems仕様で定義されたBIFSノード名のリストの項目と比較して、対応するBIFSノードの「ノード番号」の値を決定する。この「ノード番号」の値、BIFSノードの「ノード・データ型」、およびMPEG-4 Systems仕様で定義されたテーブルの組を使用して、このBIFSノードの「localNodeType」の値およびこの値を表すのに使用されるビット数を決定する。この「localNodeType」の値を、指定されたビット数を使用して、mp4ファイルに書き込む1714。

【0690】

mp4bifs Node要素が、ReplaceScene bifsCommand要素に従属する場合に、対応するBIFSノードの「ノード・データ型」を「SFWorldNode」として定義する。そうでない場合に、BIFSノードの「ノード・データ型」は、「親ノード」のノード番号、親ノードの関連するプロパティ・フィールドのフィールド・インデックス、およびMPEG-4 Systems仕様で定義されたテーブルの組によって決定される。mp4bifs Node要素が、別のmp4bifs Node要素に従属する場合に、「親ノード」は、それが従属するmp4bifs Node要素によって定義される。mp4bifs Nodeが、mp4bifs bifsCommand要素に従属する場合に、「親ノード」は、それが従属するmp4bifs bifsCommand要素の「NodeId」属性または「parentId」属性によって決定される。

【0691】

次に、任意選択の「NodeId」属性3010の存在について、mp4bifs Node要素を検査する。このNode要素の「NodeId」属性の値が指定されていない場合に、単一の「0」ビットをmp4ファイルに書き込んで、条件「isUpdateable=偽」1716を示す。そうでない（すなわち、このNode要素のNodeId属性の値が指定されている）場合には、NodeId属性3010の値を、量「nodeID」に割り当て、このノードを「更新可能」ノードとして分類し、単一の「1」ビットをmp4ファイルに書き込む1716。これに、量「nodeID」1718の値が続く。量「nodeID」の整数値を表すのに使用されるビット数は、量「NodeIdBits」の値によって指定される。boolean量「bUseNames」の値が「真」である場合に、Node要素の「name」属性3016の値を文字単位でmp4ファイルにコピーし、これにヌル・バイト（8ビットのゼロ）1720を続ける。

【0692】

次に、単一の「1」ビットをmp4ファイルに書き込んで、BIFSノードの「マスク・ノード」形式が選択されたことを示す1722。これに、マスク・ビット1726のシーケンスと、多分プロパティ・フィールド値1728が続く。各マスク・ビット1726は、mp4bifs Node要素の要素名によって指定されるBIFSノードのタイプの「公開された」プロパティ・フィールドに対応する。公開されたプロパティ・フィールドは、MPEG-4 Systems仕様のテーブルによって定義される。指定されたBIFSノードのプロパティ・フィールドの順序付きの組の各メンバは、順番であるとみなされる。公開されたプロパティ・フィールドの1つに対応し、その属性値がmp4bifs Node要素で指定されるプロパティ・フィールドごとに、値「1」を有するマスク

・ビットをmp4ファイルに書き込み、これに、関連する属性値のバイナリ表現を続ける。各そのような属性値を表すのに使用される手段を、下で説明する。公開されたプロパティ・フィールドの1つに対応し、その属性値がmp4bifs Node要素で指定されないプロパティ・フィールドごとに、値「0」を有するマスク・ビットをmp4ファイルに書き込む。

【0693】

指定されたプロパティ・フィールド名および指定されたプロパティ・フィールド・データ型のほかに、BIFSノードの各タイプの各プロパティ・フィールドは、指定された特性を有し、この指定された特性によって、そのプロパティ・フィールドが、指定されたデータ型の単一の値からなる(SFFieldプロパティ)か、指定されたデータ型の複数の値からなる(MFFieldプロパティ・フィールド)かが決定される。各MFFieldプロパティ・フィールドは、図47(1760)および図48(1780)からわかるように、0個以上のSFFieldを含む。本発明のこの実施形態では、図47に示されたリスト形式(1760)が、必ず選択される。リスト形式またはベクトル形式の選択は、重要でなく、本発明は、MFField構造にベクトル形式1780を使用して同等によく実施することができる。

【0694】

7.2.3.20 SFField構造

各SFFieldデータ構造は、指定されたデータ型を表す。サポートされるデータ型に、boolean、int、float、string、およびcolorなどの「単純データ型」と、「複合データ型」が含まれる。複合データ型は、データ型「node」および「buffer」からなる。単純データ型を有する各プロパティ・フィールドのバイナリ表現は、mp4bifs Node要素の同一の名前の属性の値によって決定される。たとえば、「boolean」プロパティ・フィールドは、単一のビットによって表される。「integer」プロパティ・フィールドは、32ビット整数値によって表されるなどである。プロパティ・フィールドの各型を表すのに使用されるビット数は、MP EG-4 Systems仕様で定義されている。

【0695】

型「node」のプロパティ・フィールドの場合に、そのプロパティ・フィールドの値は、従属するmb4bifs Node要素によって表される。このプロパティ・フィールドに関連する属性の値は、従属するmp4bifs Node要素の要素名のリストからなる。各そのような従属するNode要素のバイナリ表現は、上でmp4bifs Node要素に関して指定した手順(SFNode構造)を再帰的に適用することによって作成される。

【0696】

型「buffer」のプロパティ・フィールドの場合に、従属要素は、mb4bifs bifsCommand要素からなる。このプロパティ・フィールドに関連する属性の値は、従属するmp4bifs bifsCommand要素の要素名のリストからなる。各そのような従属するbifsCommand要素のバイナリ表現は、上でmp4bifs commandFrame要素に関して指定したものと同一の手順を再帰的に適用することによって作成される。

【0697】

7.2.4 「mov」要素の処理

出力mp4ファイル2230の作成の第4ステップは、mp4file文書2300に含まれる単一のmov要素2320の処理からなる。

【0698】

標準XML手段を使用して、図59に示されたmp4file文書2300のトップ・レベル要素に従属する単一の「mov」要素2320を入手する。図103に示された手順を使用して、出力mp4ファイル内にアトムID「mov」を有するアトム712および758を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を

、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値760の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、atomId「moov」766を出力mp4ファイルに書き込む。「sizePos」の値を量「moovSizePos」に割り当てる。

【0699】

次の属性が、mp4bifs「moov」要素について定義されている：version、flags、creationTime、modifyTime、timeScale、duration、およびnextTrackID。これらの属性のそれぞれについて指定された値を、同一の名前の量（「プロパティ値」）に割り当てる。これらのプロパティ値のどれもが、「mvhd」アトム772が作成されるまで、出力mp4ファイルに書き込まれない。 10

【0700】

図103に示された手順を使用して、アトムID「mvhd」を有するアトム772を出力mp4ファイル内に作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムId「mvhd」を出力mp4ファイルに書き込む。「sizePos」の値を量「mvhdSizePos」に割り当てる。

【0701】

動作5040で、「moov」要素の属性から導出される次のプロパティ値を、出力mp4ファイルに書き込む。 20

1. version（8ビット整数）
2. flags（24ビット整数）
3. creationTime（32ビット整数）
4. modifyTime（32ビット整数）
5. timeScale（32ビット整数）
6. duration（32ビット整数）
7. reserved（76バイト）
8. nextTrackID（32ビット整数）

【0702】

「reserved」データ値は、バイト1、4、17、および33が値「1」を有し、バイト48が値「4」を有することを除いて、すべて0である。 30

【0703】

mvhdアトム772は、従属アトムを有しない5050。

【0704】

mvhdアトム772のプロパティ・フィールドを書き込んだ後に、mp4ファイル内のmvhdアトム772のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0705】

mvhdアトム772を完了した後に、量「trackNum」の値に0をセットする。次に、標準XML手段を使用して、「moov」要素2320に従属するmp4file要素のそれぞれを入手する。図59および60からわかるように、これらの従属要素に、単一の「mp4iods」要素2330、任意選択の「udta」（ユーザ・データ）要素2340、および1つまたは複数の「trak」要素2350が含まれる可能性がある。これらの従属要素のそれぞれを、下で説明するように動作5050で処理する。 40

【0706】

「moov」要素2320に従属するすべてのmp4file要素の処理を完了した後に、量「moovSizePos」の値を「sizePos」に割り当て、moovアトム758のアトム・サイズ760の値を、図103（動作5060から5095）に示されているように更新する。 50

【0707】

7. 2. 4. 1 mp4fiods要素の処理

標準XML手段を使用して、図59に示された「moov」要素2320に従属する「mp4fiods」要素2330を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「iods」を有するアトム778および800を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を量「sizePos」に割り当てる。動作5010で、アトム・サイズ値804の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムId「iods」808を出力mp4ファイルに書き込む。「sizePos」の値を量「iodsSizePos」に割り当てる。

10

【0708】

次の属性が、「mp4iods」要素について定義されている：version、objectDescriptorID 2370、url、includeInlineProfilesFlag、ODProfileLevelIndication、sceneProfileLevelIndication、audioProfileLevelIndication、visualProfileLevelIndication、およびgraphicsProfileLevelIndication。これらの属性のそれぞれについて指定される値を、同一の名前の量（プロパティ値）に割り当てる。

【0709】

量「version」の値を、32ビット整数812および816としてmp4ファイルに書き込む。この値は、iodsアトムの「バージョン」値812および「フラグ」値816の両方を表す。

20

【0710】

図104に示された手順を使用して、出力mp4ファイル内でMp4fInitObjDescrオブジェクト構造824を作成する。動作5100で、構造タグ「MP4_IOD_TAG」（値=16）828を8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を量「sizePos」に割り当て、「sizePos」の値を量「mp4fiodPos」に割り当てる。動作5120で、値「1」を、量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的なサイズ値832として出力mp4ファイルに書き込む。

30

【0711】

量「objectDescriptorID」の値を、10ビット整数836としてmp4ファイルに書き込む。「url」属性の値が指定されている場合に、単一の「1」ビットを、mp4ファイルに書き込む840。そうでない場合に、単一の「0」ビットをmp4ファイルに書き込む840。量「includeInlineProfilesFlag」の値が真である場合に、単一の「1」ビットをmp4ファイルに書き込む844。そうでない場合に、単一の「0」ビットをmp4ファイルに書き込む844。値「15」を、4ビット整数（バイナリ値=1111）848としてmp4ファイルに書き込む。

【0712】

「url」属性の値が指定されている場合に、量「url」の値を、「Pascalストリング」（文字数の1バイトと、それに続く指定された個数の文字バイト）としてmp4ファイルに書き込む。そうでない場合に、動作5140で、量ODProfileLevelIndication 852、sceneProfileLevelIndication 856、audioProfileLevelIndication 860、visualProfileLevelIndication 864、およびgraphicsProfileLevelIndication 868の値を、それぞれ8ビット整数としてmp4ファイルに書き込む。

40

【0713】

次に、標準XML手段を使用して、動作5050で、「mp4fiods」要素に従属

50

する各要素を入手する。図 60 からわかるように、「mp4fiods」要素 2360 は、1 つまたは複数の従属する「EsIdInc」要素 2380 を有すると期待され、各「EsIdInc」要素 2380 は、「trackID」属性を有する。

【0714】

「mp4fiods」要素 2360 に従属する「EsIdInc」要素 2380 ごとに、次の値からなる ES_ID_Inc データ構造 872 を、出力 mp4 ファイルに付加する。

1. 「ES_ID_IncTag」（値 = 14）880 の値を表す 1 バイト
2. 値「4」（「numBytes」）884 を表す 1 バイト
3. 32 ビット整数（「ES_ID」）888 として「trackID」属性の値を表す 4 バイト

【0715】

動作 5050 で「mp4fiods」要素 2360 に従属するすべての「EsIdInc」要素 2380 の処理を完了した後に、MP4_IOD サイズ 832 の値を、図 104（動作 5160 から 5195）に示されているように更新する。次に、量「iodsSizePos」の値を量「sizePos」に割り当て、iods アトム 800 のアトム・サイズ 804 を、図 103（動作 5060 から 5095）に示されているように更新する。

【0716】

7. 2. 4. 2 各 track 要素の処理

標準 XML 手段を使用して、図 59 に示された「moov」要素 2320 に従属する「trak」要素 2350 のそれぞれを入手する。図 103 に示された手順を使用して、出力 mp4 ファイル内でアトム ID「trak」を有するアトム 790 および 900 を作成する。動作 5000 で、出力 mp4 ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作 5010 で、アトム・サイズ値 903 の代わりに、値 0 を出力 mp4 ファイルに書き込む。動作 5020 で、アトム ID「trak」906 を出力 mp4 ファイルに書き込む。「sizePos」の値を量「trakSizePos」に割り当てる。

【0717】

次の要素が、「trak」要素について定義されている：version、flags、creationTime、modifyTime、trackID、および duration。これらの属性のそれぞれについて指定される値を、同一の名前の量（プロパティ値）に割り当てる。これらのプロパティ値のどれもが、「tkhd」アトム 910 が作成されるまで、出力 mp4 ファイルに書き込まれない。

【0718】

図 103 に示された手順を使用して、出力 mp4 ファイル内にアトム ID「tkhd」を有するアトム 910 を作成する。動作 5000 で、出力 mp4 ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作 5010 で、アトム・サイズ値の代わりに、値 0 を出力 mp4 ファイルに書き込む。動作 5020 で、アトム ID「tkhd」を出力 mp4 ファイルに書き込む。「sizePos」の値を量「tkhdSizePos」に割り当てる。

【0719】

動作 5040 で、次の量の値を mp4 ファイルに書き込む。

1. version（8 ビット整数）
2. flags（24 ビット整数）
3. creationTime（32 ビット整数）
4. modifyTime（32 ビット整数）
5. trackID（32 ビット整数）
6. reserved1（32 ビット、0）ここでファイル位置を「durationPos」として保存する

- 7. duration (32ビット整数)
- 8. reserved2 (56バイト)
- 9. reserved3 (32ビット、値=0x01400000)
- 10. reserved4 (32ビット、値=0x00f00000)

【0720】

「reserved2」データ値は、値「1」を有するバイト17および33と、値「4」を有するバイト48を除いて、すべて0である。

【0721】

tkhdアトム910は、従属アトムを有しない。動作5040で、tkhdアトム910のプロパティ・フィールドを書き込んだ後に、mp4バイナリ・ファイル内のtkhdアトム910のアトム・サイズの値を、図103（動作5060から5195）に示されているように更新する。

【0722】

tkhdアトム910を完了した後に、標準XML手段を使用して、「trak」要素2600に従属するmp4file要素のそれぞれを入手する。図68および69に示されているように、これらの従属要素に、単一の「mdia」要素2604、可能な「ref」（track reference）要素2636、および可能な「edts」（edit list）要素2644を含めることができる。これらの従属要素のそれぞれを、下で説明するように動作5050で処理する。

【0723】

「trak」要素2600に従属するすべてのmp4file要素の処理を完了した後に、量「trakSizePos」の値を「sizePos」に割り当て、trakアトム900のアトム・サイズ903の値を、図103（動作5060から5095）に示されているように更新する。量「trackNum」の値を1つ増分する。

【0724】

7. 2. 4. 3 mdia要素の処理

標準XML手段を使用して、図68に示された各「trak」要素2600に従属する単一の「mdia」要素2604を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「mdia」を有するアトム912を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「mdia」を出力mp4ファイルに書き込む。「sizePos」の値を量「mdiaSizePos」に割り当てる。

【0725】

次の属性が、「mdia」要素について定義されている：version、flags、creationTime、modifyTime、timeScale、duration、language、およびquality。これらの属性のそれぞれについて指定された値を、同一の名前の量（プロパティ値）に割り当てる。これらのプロパティ値のどれもが、「mdhd」アトム915が作成されるまで、出力mp4ファイルに書き込まれない。

【0726】

図103に示された手順を使用して、出力mp4ファイル内でアトムID「mdhd」を有するアトム915を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5040で、次の量の値をmp4ファイルに書き込む。

- 1. version (8ビット整数)
- 2. flags (24ビット整数)
- 3. creationTime (32ビット整数)
- 4. modifyTime (32ビット整数)

10

20

30

40

50

- 5. `timeScale` (32ビット整数)
- 6. `duration` (32ビット整数)
- 7. `language` (16ビット整数)
- 8. `quality` (16ビット整数)

【0727】

`mdhd`アトム915は、従属アトムを有しない。動作5040で、`mdhd`アトム915のプロパティ・フィールドを書き込んだ後に、`mp4`ファイル内の`mdhd`アトム915のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0728】

10

次に、メディア・データ2220を調べて、各「サンプル」（「アクセス単位」とも称する）のサイズおよび持続時間を決定する。リスト`StreamTypeForTrack`の量`trackNum`の値によって指定される項目の値を、量`streamType`に割り当てる。リスト`ObjectTypeForTrack`の量`trackNum`の値によって指定される項目の値を、量`objectType`に割り当てる。

【0729】

量`streamType`の値が「4」である場合に、`sampleSize`、`sampleTime`、および`syncSample`という名前の3つの新しいリストを作成する。各リストの項目数は、リスト`MediaSamples`の量`trackNum`の値によって指定される項目の値によって決定される。次に、メディア・データ・ファイルのデータを媒体依存の形で調べて、各サンプルのサイズおよび持続時間を判定する。その結果を、リスト`sampleSize`および`sampleTime`の項目に割り当てる。「ランダム・アクセス・サンプル」と決定された各サンプルのインデックスを、リスト「`syncSample`」の項目に割り当てる。

20

【0730】

量`streamType`の値が「5」である場合に、`sampleSize`および`sampleTime`という名前の2つの新しいリストを作成する。各リストの項目数は、リスト`MediaSamples`の量`trackNum`の値によって指定される項目の値によって決定される。次に、メディア・データ・ファイル内のデータを媒体依存の形で調べて、各サンプルのサイズおよび持続時間を判定する。その結果を、リスト`sampleSize`および`sampleTime`の項目に割り当てる。

30

【0731】

他の場合に（`streamType`が「4」でも「5」でもない）、メディア・データ・ストリーム全体が、1つの巨大なサンプルになると定義される。

【0732】

次に、標準XML手段を使用して、「`mdia`」要素2604に従属する各要素を入手する。図68からわかるように、この従属要素に、単一の「`hdlr`」（`handler`）要素2608、単一の「`minf`」（`media info`）要素2612、単一の「`stbl`」（`sample table`）要素2628、およびメディア・ヘッダ（「`*mhd`」）要素2632が含まれる。これらの従属要素のそれぞれを、下で説明するように動作5050で処理する。

40

【0733】

「`mdia`」要素2604に従属するすべての`mp4file`要素の処理が完了した後に、量「`mdiaSizePos`」の値を「`sizePos`」に割り当て、メディア・アトム912のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0734】

7. 2. 4. 4 `hdlr`要素の処理

標準XML手段を使用して、図68に示された「`mdia`」要素2604に従属する単一の「`hdlr`」要素2608を入手する。図103に示された手順を使用して、出力m

50

p 4 ファイル内でアトム ID「hdlr」を有するアトム 918 を作成する。動作 5000 で、出力 mp 4 ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作 5010 で、アトム・サイズ値の代わりに、値 0 を出力 mp 4 ファイルに書き込む。動作 5020 で、アトム ID「hdlr」を出力 mp 4 ファイルに書き込む。

【0735】

次の属性が、「hdlr」要素 2608 について定義されている：version、flags、handlerType、および name。これらの属性のそれぞれについて指定された値を、同一の名前の量の量に割り当てる。次に、動作 5040 で、次の量の値を mp 4 ファイルに書き込む。

1. version (8 ビット整数)
2. flags (24 ビット整数)
3. handlerType (32 ビット整数)
4. reserved (12 バイト、すべて 0)
5. name (ヌル終端文字ストリング)

【0736】

hdlr 要素 2608 は、従属する要素を有しない 5050。hdlr アトム 918 のプロパティ・フィールドを書き込んだ後に、hdlr アトム 918 のアトム・サイズ値を、図 103 (動作 5060 から 5095) に示されているように更新する。

【0737】

7. 2. 4. 5 minf 要素の処理

標準 XML 手段を使用して、図 68 に示された「mdia」要素 2604 に従属する単一の「minf」要素 2612 を入手する。図 103 に示された手順を使用して、出力 mp 4 ファイル内でアトム ID「minf」を有するアトム 920 を作成する。動作 5000 で、出力 mp 4 ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作 5010 で、アトム・サイズ値の代わりに、値 0 を出力 mp 4 ファイルに書き込む。動作 5020 で、アトム ID「minf」を出力 mp 4 ファイルに書き込む。「sizePos」の値を、量「minfSizePos」に割り当てる。

【0738】

「minf」要素 2612 の属性はなく、minf アトム 920 のプロパティ・フィールドはない (動作 5030 および 5040 参照)。

【0739】

次に、動作 5050 で、標準 XML 手段を使用して、「minf」要素 2612 に従属するすべての要素を入手する。図 68 からわかるように、この従属する要素に、「dinf」要素 2616、「stbl」要素 2628、およびメディア・ヘッダ (*mhd) 要素 2632 が含まれる可能性がある。「stbl」要素 2628 およびメディア・ヘッダ (*mhd) 要素 2632 を処理するのに使用される手段を、下で説明する。

【0740】

「dinf」要素 2616 の場合に、図 103 に示された手順を使用して、出力 mp 4 ファイル内でアトム ID「dinf」を有するアトム 924 を作成する。動作 5000 で、出力 mp 4 ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作 5010 で、アトム・サイズ値の代わりに、値 0 を出力 mp 4 ファイルに書き込む。動作 5020 で、アトム ID「dinf」を出力 mp 4 ファイルに書き込む。「sizePos」の値を、量「dinfSizePos」に割り当てる。

【0741】

「dinf」要素 2616 の属性はなく、dinf アトム 924 のプロパティ・フィールドはない (動作 5030 参照)。

【0742】

次に、動作 5040 で、標準 XML 手段を使用して、「dinf」要素 2616 に従属するすべての要素を入手する。図 68 からわかるように、この従属する要素に、「dref」要素 2620 が含まれる可能性がある。

10

20

30

40

50

【0743】

「d i n f」要素2616が、従属する「d r e f」要素2620を所有する場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「d r e f」を有するアトム927を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「d r e f」を出力mp4ファイルに書き込む。「s i z e P o s」の値を、量「d r e f S i z e P o s」に割り当てる。「d r e f」要素の属性はなく、d r e fアトムのプロパティ・フィールドはない。

【0744】

10

次に、標準XML手段を使用して、「d r e f」要素2620に従属するすべての要素を入手する。図68からわかるように、この従属する要素に、「u r l D a t a」要素2624が含まれる可能性がある。

【0745】

「d r e f」要素2620が、従属する「u r l D a t a」要素2624を所有する場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「u r l」を有するアトム930を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、4文字アトムID「u r l」（u-r-l-スペース）を出力mp4ファイルに書き込む。

20

【0746】

動作5030で、「u r l D a t a」要素の「v e r s i o n」、「f l a g s」、および「l o c a t i o n」属性の値を、同一の名前の量に割り当てる。

【0747】

動作5400で、次の値を出力mp4ファイルに書き込む。

1. 量「v e r s i o n」の値を表す8ビット整数
2. 量「f l a g s」の値を表す24ビット整数
3. 量「l o c a t i o n」の値を表すヌル終端文字ストリング

【0748】

「u r l D a t a」要素2624は、従属要素を有しない（動作5050参照）。これらのプロパティ値を出力mp4ファイルに書き込んだ後に、「u r l」アトム930のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

30

【0749】

量「d r e f S i z e P o s」の値を、「s i z e P o s」に割り当て、d r e fアトム927のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0750】

量「d i n f S i z e P o s」の値を、「s i z e P o s」に割り当て、d i n fアトム924のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

40

【0751】

「m i n f」要素2612に従属するすべてのmp4 f i l e要素の処理を完了した後に、量「m i n f S i z e P o s」の値を量「s i z e P o s」に割り当て、m i n fアトム920のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0752】

7. 2. 4. 6 s t b l要素の処理

標準XML手段を使用して、図68に示された「m i n f」要素2612に従属する単一の「s t b l」要素2628を入手する。図103に示された手順を使用して、出力m

50

p 4 ファイル内でアトム ID 「s t b l」を有するアトム 9 3 3 および 9 5 0 を作成する。動作 5 0 0 0 で、出力 m p 4 ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作 5 0 1 0 で、アトム・サイズ値 9 5 4 の代わりに、値 0 を出力 m p 4 ファイルに書き込む。動作 5 0 2 0 で、アトム ID 「s t b l」9 5 7 を出力 m p 4 ファイルに書き込む。「s i z e P o s」の値を量「s t b l S i z e P o s」に割り当てる。

【0 7 5 3】

「s t b l」要素 2 6 2 8 の属性はなく、s t b l アトムのプロパティ・フィールドはない（動作 5 0 3 0 および 5 0 4 0 参照）。

【0 7 5 4】

次に、標準 XML 手段を使用して、「s t b l」要素 2 6 5 2 に従属するすべての要素を入手する。図 6 9 からわかるように、この従属要素に、「s t s c」要素 2 6 5 6、「s t t s」要素 2 6 6 0、「s t c o」要素 2 6 6 4、「s t s z」要素 2 6 6 8、「s t s s」要素 2 6 7 2、および「s t s d」要素 2 6 7 6 が含まれる可能性がある。単一インスタンスが任意選択である「s t s s」要素 2 6 7 2 を除いて、これらの要素のそれぞれの 1 つが必要である。これらの要素のそれぞれを処理するのに使用される手段を、下の動作 5 0 5 0 で説明する。

【0 7 5 5】

「s t b l」要素 2 6 5 2 に従属するすべての m p 4 f i l e 要素の処理が完了した後に、量「s t b l S i z e P o s」の値を「s i z e P o s」に割り当て、s t b l アトム 9 5 0 のアトム・サイズ 9 5 4 の値を、図 1 0 3（動作 5 0 6 0 から 5 0 9 5）に示されているように更新する。

【0 7 5 6】

7. 2. 4. 7 s t s c 要素の処理

標準 XML 手段を使用して、図 6 9 に示された「s t b l」要素 2 6 5 2 に従属する単一の「s t s c」要素 2 6 5 6 を入手する。図 1 0 3 に示された手順を使用して、出力 m p 4 ファイル内でアトム ID 「s t s c」を有するアトム 9 6 0 を作成する。動作 5 0 0 0 で、出力 m p 4 ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作 5 0 1 0 で、アトム・サイズ値の代わりに、値 0 を出力 m p 4 ファイルに書き込む。動作 5 0 2 0 で、アトム ID 「s t s c」を出力 m p 4 ファイルに書き込む。

【0 7 5 7】

「s t s c」要素 2 6 5 6 は、属性「v e r s i o n」および「f l a g s」を有する。動作 5 0 3 0 で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作 5 0 4 0 で、量「v e r s i o n」の値を、8 ビット整数として m p 4 ファイルに書き込み、量「f l a g s」の値を、2 4 ビット整数として m p 4 ファイルに書き込む。

【0 7 5 8】

m p 4 ファイルのファイル位置を、量「n u m E n t r i e s P o s」に割り当て、3 2 ビットの 0 の値を m p 4 ファイルに書き込み、値 0 を量「n u m E n t r i e s」に割り当てる。

【0 7 5 9】

次に、動作 5 0 5 0 で、標準 XML 手段を使用して、「s t s c」要素 2 6 5 6 に従属するすべての要素を入手する。これらの従属要素は、1 つまたは複数の「s a m p l e T o C h u n k」要素からなる。各「s a m p l e T o C h u n k」要素は、属性「f i r s t C h u n k」、「n u m S a m p l e s」、および「s a m p l e D e s c」を所有する。

【0 7 6 0】

次の動作を、「s t s c」要素 2 6 5 6 に従属する「s a m p l e T o C h u n k」要素ごとに実行する。

(1) 「f i r s t C h u n k」属性の値を、3 2 ビット整数として m p 4 ファイルに書き込む。

10

20

30

40

50

(2) 「numSamples」属性の値を、32ビット整数としてmp4ファイルに書き込む。

(3) 「sampleDesc」属性の値を、32ビット整数としてmp4ファイルに書き込む。

(4) 量「numEntries」の値を1つ増分する。

【0761】

「stsc」要素2656に従属する要素のすべての処理を完了した後に、動作5060で、mp4ファイルのファイル位置を量「endPos」に割り当てる。mp4ファイルのファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numEntries」の値を、32ビット整数としてmp4ファイルに書き込む。次に、stscアトム960のアトム・サイズの値を、図103（動作5070から5095）に示されているように更新する。 10

【0762】

7. 2. 4. 8 stts要素の処理

標準XML手段を使用して、図69に示された「stbl」要素2652に従属する単一の「stts」要素2660を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「stts」を有するアトム963を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「stts」を出力mp4ファイルに書き込む。 20

【0763】

「stts」要素2660は、属性「version」および「flags」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5040で、量「version」の値を、8ビット整数としてmp4ファイルに書き込み、量「flags」の値を、24ビット整数としてmp4ファイルに書き込む。

【0764】

mp4ファイルのファイル位置を、量「numEntriesPos」に割り当て、32ビットの0の値をmp4ファイルに書き込み、値0を量「numEntries」に割り当てる。

【0765】

リストStreamTypeForTrackの項目「trackNum」の値を、量「streamType」に割り当てる。リストObjectTypeForTrackの項目「trackNum」の値を、量「objectType」に割り当てる。量「trackNum」の値は、手順「trak要素の処理」の一部として決定されたものである。 30

【0766】

量「streamType」の値が「1」である場合に、odsmデータがodsmのmdatアトムに入力された時に作成されたodsmサンプル時間値（OdsmSampleTime）のリストを使用して、各odsmサンプルの持続時間値を決定する。各odsmサンプルの持続時間は、このリストの連続する項目の値の間の差によって決定される。これらの値は、トラック時間単位で指定される。 40

【0767】

量「streamType」の値が「3」である場合に、sdsmデータがsdsmのmdatアトムに入力された時に作成されたsdsmサンプル時間値（SdsmSampleTime）のリストを使用して、各sdsmサンプルの持続時間を決定する。各sdsmサンプルの持続時間は、このリストの連続する項目の値の間の差によって決定される。秒単位の結果の持続時間に、この「stts」要素を含む「trak」要素の「timeScale」属性の値をかけて、トラック時間単位の持続時間値を決定する。

【0768】

量「streamType」の値が「4」である場合、または量「streamType 50

e」の値が「5」かつ量objectTypeの値が「64」または「107」である場合に、対応するメディア・データがmdatアトムに入力された時に作成されたメディア・サンプル時間値(sampleTime)のリストを使用して、各メディア・サンプルの持続時間値を決定する。各メディア・サンプルの持続時間は、このリストの対応する項目によって指定される。これらの値は、トラック時間単位で指定される。

【0769】

前の3つの場合のそれぞれで、同一の持続時間を有する連続するサンプルの個数を、量「numSamples」に割り当てる。サンプルの持続時間の値が前のサンプルの持続時間と異なるたびに、「numSamples」の値を32ビット整数としてmp4ファイルに書き込み、前のサンプルの持続時間の値を32ビット整数としてmp4ファイルに書き込み、値1を値「numSamples」に割り当て、量「numEntries」の値を1つ増分する。

10

【0770】

そうでない場合に、標準XML手段を使用して、「stts」要素2660に従属するすべての要素を入手する。この従属要素に、1つまたは複数の「timeToSample」要素が含まれる。各「timeToSample」要素は、属性「numSamples」および「duration」を所有する。

【0771】

次の動作を、「stts」要素2660に従属する「timeToSample」要素ごとに繰り返す。

20

(1) 「numSamples」属性の値を、32ビット整数としてmp4ファイルに書き込む。

(2) 「duration」属性の値を、32ビット整数としてmp4ファイルに書き込む。

(3) 量「numEntries」の値を1つ増分する。

【0772】

「stts」要素2660に従属するすべての要素の処理を完了した後に、動作5060で、mp4ファイルのファイル位置を量「endPos」に割り当てる。mp4ファイルのファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numEntries」の値を、32ビット整数としてmp4ファイルに書き込む。次に、sttsアトム963のアトム・サイズの値を、図103(動作5070から5095)に示されているように更新する。

30

【0773】

7.2.4.9 stco要素の処理

標準XML手段を使用して、図69に示された「stbl」要素2652に従属する単一の「stco」要素2664を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「stco」を有するアトム966を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「stco」を出力mp4ファイルに書き込む。

40

【0774】

「stco」要素2664は、属性「version」および「flags」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5040で、量「version」の値を、8ビット整数としてmp4ファイルに書き込み、量「flags」の値を、24ビット整数としてmp4ファイルに書き込む。

【0775】

mp4ファイルのファイル位置を、量「numEntriesPos」に割り当て、32ビットの0の値をmp4ファイルに書き込み、値0を量「numEntries」に割り当てる。

【0776】

50

次に、動作5050で、標準XML手段を使用して、「stco」要素2664に従属するすべての要素を入手する。これらの従属要素は、1つまたは複数の「chunkOffset」要素からなる。「chunkOffset」要素について、次の2つの属性が定義されている:「mdatId」および「mdatOffset」

【0777】

次の動作を、「stco」要素2664に従属する「chunkOffset」要素ごとに繰り返す。

(1)「mdatId」属性および「mdatOffset」属性について指定された値を、同一の名前の量に割り当てる。

(2)量「chunk」の値を、次の3つの条件によって決定する:

a. リストmdatIdForChunkの対応する項目が、量mdatIdの値と一致し、

b. リストtrackIdForChunkの対応する項目が、量trackIdの値と一致し、

c. numEntriesの値が、前の2つの条件を満足する項目の個数と一致する

(3)リストoffsetForChunkの項目「chunk」の値を、量「chunkOffset」に割り当てる。

(4)量「chunkOffset」の値を、32ビット整数としてmp4ファイルに書き込む。

(5)量「numEntries」の値を1つ増分する。

【0778】

「stco」要素2664に従属するすべての要素の処理を完了した後に、動作5060で、mp4ファイルのファイル位置を量「endPos」に割り当てる。mp4ファイルのファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numEntries」の値を、32ビット整数としてmp4ファイルに書き込む。次に、stcoアトム966のアトム・サイズの値を、図103（動作5070から5095）に示されているように更新する。

【0779】

7.2.4.10 stsz要素の処理

標準XML手段を使用して、図69に示された「stbl」要素2652に従属する単一の「stsz」要素2668を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「stsz」を有するアトム970を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「stsz」を出力mp4ファイルに書き込む。

【0780】

「stsz」要素2668は、属性「version」、「flags」、「sizeForAll」、および「numSamples」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5040で、量「version」の値を、8ビット整数としてmp4ファイルに書き込み、量「flags」の値を、24ビット整数としてmp4ファイルに書き込む。

【0781】

リストStreamTypeForTrackの項目「trackNum」の値を、量「streamType」に割り当てる。リストObjectTypeForTrackの項目「trackNum」の値を、量「objectType」に割り当てる。量「trackNum」の値は、手順「trak要素の処理」の一部として決定されたものである。

【0782】

量「streamType」の値が「1」であり、量「numOdsmSamples」の値が「1」である場合に、値「1」を量「numSamples」に割り当て、リス

10

20

30

40

50

トOdsmsSampleSizeの最初の項目の値を、量「sizeForAll」に割り当てる。

【0783】

量「streamType」の値が「3」であり、リストSdsmsSampleSizeの第2項目が負である場合に、値「1」を、量「numSamples」に割り当て、リストSdsmsSampleSizeの第1項目の値を、量「sizeForAll」に割り当てる。リストSdsmsSampleSizeの負の項目は、このリストの終りを示す。

【0784】

量「streamType」の値が「5」であり、量「objectType」の値が「193」であり、量「sizeForAll」の値が1未満である場合に、値「24」を量「numSamples」に割り当て、リストSdsmsSampleSizeの第1項目の値を、量「sizeForAll」に割り当てる。24バイトは、「193」のobjectTypeによって定義されるオーディオ・ストリームの各サンプルのサイズである。

【0785】

mp4ファイルのファイル位置を、量「sizeForAllPos」に割り当て、量「sizeForAll」の値を、32ビット整数としてmp4fileに書き込む。mp4ファイルのファイル位置を、量「numEntriesPos」に割り当て、量「numSamples」の値を、32ビット整数としてmp4fileに書き込む。値0を、量「numEntries」に割り当てる。

【0786】

「stsz」要素2668は、従属する「sampleSize」要素を有する場合があるが、この要素に含まれるデータ値が、現在のメディア・データ・ストリームと必ずしも一貫しないので、これらは無視される。その代わりに、すべてのサイズ値が、ストリームが作成される時(odsmおよびsdsm)またはmdiaアトムが作成される時(オーディオ・ストリームおよびビデオ・ストリーム)に、再計算されている。結果のサンプル・サイズ値が、リストOdsmsSampleSize、SdsmsSampleSize、またはsampleSizeのいずれか1つに入力されている。メディア・データが均一のサンプル・サイズを有する場合に、各サンプルの値は、量「sizeForAll」によって指定される。

【0787】

量「streamType」の値が「1」(odsm)であり、量「numOdsmsSamples」の値が1を超える場合に、量「numOdsmsSamples」の値を量「numEntries」に割り当てる。量「numOdsmsSamples」の値は、リストOdsmsSampleSizeの項目数を示す。リストOdsmsSampleSizeの各項目の値を、32ビット整数としてmp4ファイルに書き込む。量「numOdsmsSamples」の値が量「numSamples」の値と同一でない場合に、mp4ファイルのファイル位置を、量「mp4FilePos」に割り当て、ファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numOdsmsSamples」の値を、32ビット整数としてmp4ファイルに書き込み、ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0788】

量「streamType」の値が「3」(sdsm)であり、リストSdsmsSampleSizeの第2項目が負でない場合に、リストSdsmsSampleSizeの各項目の値を、負の項目が見つかるまで、32ビット整数としてmp4ファイルに書き込む。リストSdsmsSampleSizeの負でない項目ごとに、量「numEntries」の値を1つ増分する。このリストの最後の負の値は、mp4ファイルに書き込まれない。量「numEntries」の値が、量「numSamples」の値と同一でない場合には、mp4ファイルのファイル位置を、量mp4FilePosに割り当て、ファ

イル位置を、量「numEntriesPos」によって指定される値に変更し、量「numEntries」の値を、32ビット整数としてmp4ファイルに書き込み、ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0789】

量「streamType」の値が「1」でも「3」でもなく、量「sizeForAll」の値が正でなく、量「numSamples」の値が正でない場合に、量「numMediaSamples」の値を、量「numSamples」に割り当てる。この事例に、変化するサンプル・サイズを有するビデオ・メディア・データおよびオーディオ・メディア・データのほとんどが含まれる。量「numMediaSamples」は、mediaアトムが作成された時に作成されたリスト「sampleSize」の項目数を指定する。リストsampleSizeの各項目の値を、32ビット整数としてmp4ファイルに書き込む。mp4ファイルのファイル位置を、量「mp4FilePos」に割り当て、ファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numMediaSamples」の値を32ビット整数としてmp4ファイルに書き込み、ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0790】

量「streamType」の値が「1」でも「3」でもなく、量「sizeForAll」の値が正であり、量「numSamples」の値が0である場合に、量「numSamples」の値を、このトラックのメディア・データのサイズを量「sizeForAll」の値で割ることによって決定する。この事例に、均一のサンプル・サイズを有するいくつかのオーディオ・メディア・データが含まれる。mp4ファイルのファイル位置を、量「mp4FilePos」に割り当て、ファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numSamples」の値を32ビット整数としてmp4ファイルに書き込み、ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0791】

量「streamType」の値が、「1」でも「3」でもなく、量「sizeForAll」の値が、0であり、量「numSamples」の値が正である場合に、量「sizeForAll」の値を、このトラックのメディア・データのサイズを量「numSamples」の値で割ることによって決定する。この事例に、単一のイメージを表すメディア・データなど、均一なサンプル・サイズおよび通常は「1」である既知のサンプル・カウントを有するあるメディア・データが含まれる。mp4ファイルのファイル位置を、量「mp4FilePos」に割り当て、ファイル位置を、量「sizeForAllPos」によって指定される値に変更し、量「sizeForAll」の値を32ビット整数としてmp4ファイルに書き込み、ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0792】

どの場合でも、「stsz」要素2668の処理は、図103（動作5060から5095）に示されているように、stszアトム970のアトム・サイズの値を更新することによって完了する。

【0793】

7. 2. 4. 11 stss要素の処理

「stss」要素2672が存在する場合に、この要素に「sync sample table」が含まれる。このテーブルは、MPEG-4ビデオ・ストリームなどの、あるタイプのメディア・データ・ストリームに必要である。「stbl」要素2652に、必要な場合に限って、従属する「stss」要素2672が含まれる。標準XML手段を使用して、従属する「stss」要素2672が「stbl」要素2652内に存在するかどうかを判定する。

【0794】

従属する「s t s s」要素2 6 7 2が存在する場合に、標準XML手段を使用して、この要素を入手する。図1 0 3に示された手順を使用して、出力mp 4ファイル内でアトムID「s t s s」を有するアトム9 7 4を作成する。動作5 0 0 0で、出力mp 4ファイルの現在のファイル位置を、量「s i z e P o s」に割り当てる。動作5 0 1 0で、アトム・サイズ値の代わりに、値0を出力mp 4ファイルに書き込む。動作5 0 2 0で、アトムID「s t s s」を出力mp 4ファイルに書き込む。

【0 7 9 5】

「s t s s」要素2 6 7 2は、属性「v e r s i o n」、「f l a g s」、および「n u m E n t r i e s」を有する。動作5 0 3 0で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5 0 4 0で、量「v e r s i o n」の値を、8ビット整数としてmp 4ファイルに書き込み、量「f l a g s」の値を、2 4ビット整数としてmp 4ファイルに書き込む。 10

【0 7 9 6】

mp 4ファイルのファイル位置を、量「n u m E n t r i e s P o s」に割り当て、3 2ビットの0の値をmp 4ファイルに書き込む。

【0 7 9 7】

図1 0 7に示された手段を使用して、出力mp 4ファイル内でs y n c s a m p l e t a b l eを構成する。この手順によれば、動作5 4 0 0で、リスト「M e d i a S a m p l e s」の項目「t r a k N u m」に量「i M e d i a S a m p l e s」の値を割り当てる。この量は、現在のt r a kアトム7 9 0および9 0 0に関連するメディア・データ・ストリームを表すサンプル（または「アクセス単位」）の個数を示す。この量の値は、リスト「s y n c S a m p l e」の項目数の上限も提供する。リスト「i S y n S a m p l e」は、サンプル・インデックス番号の単調増加する組からなり、メディア・データ・ストリームの最初のサンプルが、サンプル・インデックス値「1」によって表される。このリストの項目数がi M e d i a S a m p l e sの値未満である場合に、ストリームの最後のサンプル・インデックス値に、値0を有する項目が続く。 20

【0 7 9 8】

動作5 4 1 0で、値0を量「i S y n c S a m p l e」および「n u m S y n c S a m p l e s」の値に割り当てる。

【0 7 9 9】

動作5 4 2 0で、n u m S y n c S a m p l e sの値をi M e d i a S a m p l e sの値と比較する。 30

【0 8 0 0】

量「n u m S y n c S a m p l e s」の値が量「i M e d i a S a m p l e s」の値未満でない場合に、この処理は、動作5 4 3 0で完了する。

【0 8 0 1】

そうでない場合すなわち、量「n u m S y n c S a m p l e s」の値が量「i M e d i a S a m p l e s」の値未満である場合に、動作5 4 4 0で、量「i S y n c S a m p l e」の値を量「p r e v i o u s S a m p l e」の値に割り当てる。動作5 4 5 0で、項目「n u m S y n c S a m p l e s」の値を量「i S y n c S a m p l e」に割り当て、動作5 4 6 0で、量「i S y n c S a m p l e」の結果の値を量「i S y n c S a m p l e」の値と比較する。 40

【0 8 0 2】

量「i S y n c S a m p l e」の値が量「i S y n c S a m p l e」の値を超えない場合に、この処理は、動作5 4 7 0で完了する。そうではなく、量「i S y n c S a m p l e」の値が量「i S y n c S a m p l e」の値を超える場合に、動作5 4 8 0で、量「i S y n c S a m p l e」の値を3 2ビット整数として出力mp 4ファイルに書き込む。動作5 4 9 0で、量「n u m S y n c S a m p l e s」の値を1つ増分し、動作5 4 2 0での量「n u m S y n c S a m p l e s」の値と量「i M e d i a S a m p l e s」の値の比較を繰り返す。 50

【0803】

図107に示された手順を完了した後に、動作5060で、mp4ファイルのファイル位置を量「endPos」に割り当てる。mp4ファイルのファイル位置を、量「numEntriesPos」によって指定される値に変更し、量「numSyncSamples」の値を、32ビット整数としてmp4ファイルに書き込む。次に、stssアトム974のアトム・サイズの値を、図103（動作5070から5095）に示されているように更新する。

【0804】

7. 2. 4. 12 stsd要素の処理

標準XML手段を使用して、図69に示された「stbl」要素2652に従属する単一の「stsd」要素2676を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「stsd」を有するアトム978を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「stsd」を出力mp4ファイルに書き込む。 10

【0805】

「stsd」要素2676は、属性「version」および「flags」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5040で、量「version」の値を、8ビット整数としてmp4ファイルに書き込み、量「flags」の値を、24ビット整数としてmp4ファイルに書き込む。 20

【0806】

mp4ファイルのファイル位置を、量「numEntriesPos」に割り当て、32ビットの0の値をmp4ファイルに書き込み、値0を、量「numEntries」に割り当てる。量「sizePos」の値を、量「stsdSizePos」に割り当てる。

【0807】

動作5050で、標準XML手段を使用して、「stsd」要素2676に従属するすべての要素を入手する。そのような従属要素の組は、型「mp4s」、「mp4a」、または「mp4v」の単一の要素からなると期待される。これらの要素型を、図69に示されているように、集合的に「mp4*」2680と記述する。これらの従属要素のそれぞれは、「サンプル記述テーブル」の1項目を表し、量「numEntries」の値を、そのような従属要素ごとに1つ増分する。 30

【0808】

要素のこれらの型（「mp4s」、「mp4a」、または「mp4v」）のそれぞれが、「dataRefIndex」という名前の単一の属性を有する。

【0809】

型「mp4s」の従属要素ごとに、図103に示された手順を使用して、出力mp4ファイル内でアトムID「mp4s」を有するアトム982を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当て、量「sizePos」の値を、量「mp4xSizePos」に割り当てる。5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「mp4s」を出力mp4ファイルに書き込む。 40

【0810】

値0を、量「dataRefIndex」に割り当てる。動作5030で、mp4s要素の「dataRefIndex」属性の値が指定されている場合に、この属性の値を量「dataRefIndex」に割り当てる。動作5040で、次の値をmp4ファイルに書き込む。

1. 値0の6つのバイト

2. 量「dataRefIndex」の値を表す16ビット整数

【0811】

型「mp4a」の従属要素ごとに、図103に示された手順を使用して、出力mp4ファイル内でアトムID「mp4a」を有するアトム982を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当て、量「sizePos」の値を、量「mp4xSizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「mp4a」を出力mp4ファイルに書き込む。

【0812】

値0を、量「dataRefIndex」に割り当てる。動作5030で、mp4a要素の「dataRefIndex」属性の値が指定されている場合に、この属性の値を量「dataRefIndex」に割り当てる。動作5040で、次の値をmp4ファイルに書き込む。 10

1. 値0の6つのバイト
2. 量「dataRefIndex」の値を表す16ビット整数
3. 値0の2つの32ビット整数
4. 値「2」を表す16ビット整数
5. 値「16」を表す16ビット整数
6. 値0を有する32ビット整数
7. 「mdia」要素2604の対応する属性から得られた量「timeScale」の値を表す16ビット整数
8. 値0を有する16ビット整数 20

【0813】

型「mp4v」の従属要素ごとに、図103に示された手順を使用して、出力mp4ファイル内でアトムID「mp4v」を有するアトム982を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当て、量「sizePos」の値を、量「mp4xSizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「mp4v」を出力mp4ファイルに書き込む。

【0814】

値0を、量「dataRefIndex」に割り当てる。動作5030で、mp4v要素の「dataRefIndex」属性の値が指定されている場合に、この属性の値を量「dataRefIndex」に割り当てる。動作5040で、次の値をmp4ファイルに書き込む。 30

1. 値0の6つのバイト
2. 量「dataRefIndex」の値を表す16ビット整数
3. 値0を有する4つの32ビット整数
4. 値「320」を有する16ビット整数
5. 値「240」を有する16ビット整数
6. 値「72」を有する16ビット整数
7. 値0を有する16ビット整数
8. 値「72」を有する16ビット整数 40
9. 値0を有する16ビット整数
10. 値0を有する32ビット整数
11. 値「1」を有する16ビット整数
12. 値0を有する32ビット整数
13. 値「24」を有する16ビット整数
14. 値「-1」を有する16ビット整数

【0815】

型「mp4s」、「mp4a」、または「mp4v」2680の各要素は、図69に示されているように、型「esds」の単一の従属する要素2684を有すると期待される。型「esds」の従属要素ごとに、図103に示された手順を使用して、出力mp4フ 50

ファイル内でアトムID「e s d s」を有するアトム986を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「s i z e P o s」に割り当て、量「s i z e P o s」の値を、量「e s d s S i z e P o s」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「e s d s」を出力mp4ファイルに書き込む。

【0816】

型「e s d s」の要素は、属性「v e r s i o n」および「f l a g s」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。動作5040で、量「v e r s i o n」の値を、8ビット整数としてmp4ファイルに書き込み、量「f l a g s」の値を、24ビット整数としてmp4ファイルに書き込む。

10

【0817】

型「e s d s」の各要素は、図69に示されているように、型「E S _ D e s c r」の単一の従属する要素2688を有すると期待される。この従属「E S _ D e s c r」要素を、下で説明する手段を使用して、動作5050で処理する。

【0818】

「e s d s」要素2684に従属する「E S _ D e s c r」要素2688の処理を完了した後に、動作5060で、mp4ファイルのファイル位置を量「e n d P o s」に割り当てる。量「e s d s S i z e P o s」の値を、量「s i z e P o s」に割り当て、e s d sアトム986のアトム・サイズ値を、図103（動作5070から5095）に示されているように更新する。

20

【0819】

量「m p 4 x S i z e P o s」の値を、量「s i z e P o s」に割り当て、「m p 4 s」、「m p 4 a」、または「m p 4 v」のアトム・サイズの値を、図103（動作5070から5095）に示されているように更新する。

【0820】

mp4ファイルのファイル位置を、量「n u m E n t r i e s P o s」によって指定される値に変更し、量「n u m E n t r i e s」の値を、32ビット整数としてmp4ファイルに書き込む。量「s t s d S i z e P o s」の値を、「s i z e P o s」に割り当てる。s t s dアトム978のアトム・サイズの値を、図103（動作5070から5095）に示されているように更新する。

30

【0821】

7. 2. 4. 13 E S _ D e s c r 要素の処理

「e s d s」要素2684に従属する「E S _ D e s c r」要素2688および2700ごとに、図104に示された手順を使用して、出力mp4ファイル内でE S _ D e s c r構造990および1000を作成する。動作5100で、構造タグ「E S _ D e s c r T a g」（値=3）1004を8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を量「s i z e P o s」に割り当て、「s i z e P o s」の値を量「f i l e P o s 1」に割り当てる。動作5120で、値「1」を、量「n u m S i z e B y t e s」に割り当てる。動作5130で、値0を、予備的なサイズ値1008として出力mp4ファイルに書き込む。

40

【0822】

次の属性が、「E S _ D e s c r」要素について定義されている：「E S _ I D」および「p r i o r i t y」。0の値を、「E S _ I D」および「p r i o r i t y」という名前の量に割り当てる。どちらかの属性について値が指定されている場合には、動作5135で、指定された値を、同一の名前の量に割り当てる。

【0823】

動作5140で、次の値をmp4ファイルに書き込む。

1. 量「E S _ I D」の値を表す16ビット整数
2. 値0を有する3ビット整数
3. 量「p r i o r i t y」の値を表す5ビット整数

50

【0824】

次に、標準XML手段を使用して、「ES_Descr」要素2688および2700に従属する各XML要素を入手する。これらの従属要素は、図70に示されているように、「DecoderConfigDescriptor」という名前の1つの要素2710と、「SLConfigDescriptor」という名前の1つの要素2760を含むと期待される。

【0825】

「ES_Descr」要素2700に従属する「DecoderConfigDescriptor」要素2710ごとに、図104に示された手順を使用して、出力mp4ファイル内でDecoderConfigDescriptorオブジェクト構造1024および1032を作成する。動作5100で、構造タグ「DecoderConfigDescrTag」（値=4）1036を8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を量「sizePos」に割り当て、「sizePos」の値を量「filePos2」に割り当てる。動作5120で、値「1」を、量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的なサイズ値1040として出力mp4ファイルに書き込む。

【0826】

次の属性が、「DecoderConfigDescriptor」要素2710について定義されている:「objectType」、「streamType」、「upstream」、「maxBitrate」、および「avgBitrate」。これらの属性のいずれかについて値が指定されている場合に、動作5135で、各指定された値を同一の名前の量に割り当てる。

【0827】

量「bufferSize」の値を、対応するメディア・データ・ストリーム内の最大のサンプルのサイズによって決定する。量「streamType」の値が「1」である場合に、リスト「OdsmsSampleSize」の最大の項目の値を、量「bufferSize」に割り当てる。量「streamType」の値が「3」である場合に、リスト「SdsmsSampleSize」の最大の項目の値を、量「bufferSize」に割り当てる。量「streamType」の値が「4」であり、量「objectType」の値が「32」であるか、量「streamType」の値が「5」であり、量「objectType」の値が「64」であるか、量「streamType」の値が「5」であり、量「objectType」の値が「107」である場合に、リスト「sampleSize」の最大の項目の値を、量「bufferSize」に割り当てる。量「streamType」の値が「5」であり、量「objectType」の値が「193」である場合に、値「2400」を、量「bufferSize」に割り当てる。そうでない場合に、メディア・データの値は、リストmediaDataSizeの項目のうちで、リストtrackIdForChunkの対応する項目が現在のトラックの量「trackId」の値と一致するものの合計によって決定される。

【0828】

次に、動作5140で、次の値をmp4ファイルに書き込む。

1. 量「objectType」1044の整数値を表す1バイト
2. 量「streamType」1048の整数値を表す6バイト
3. 量「upstream」1052のboolean値を表す1ビット（「真」の場合に「1」、そうでない場合に「0」）
4. 値「1」を有する（「予約済み」としての）1ビット1056
5. 量「bufferSize」1060（「bufferSizeDB」としての）の整数値を表す3バイト
6. 量「maxBitrate」1064の値を表す32ビット整数
7. 量「avgBitrate」1068の値を表す32ビット整数

【0829】

10

20

30

40

50

次に、標準XML手段を使用して、「DecoderConfigDescriptor」要素2710に従属するすべての要素を入手する。これには、図70に示されているように次の型の要素の1つが含まれる可能性がある：「BIFS__DecoderConfig」2720、「JPEG__DecoderConfig」2730、「VisualConfig」2740、または「AudioConfig」2750。これらの従属要素のいずれか1つが見つかった場合に、図104に示された手順を使用して、出力mp4ファイル内でDecoderSpecificInfoオブジェクト構造1072および1076を作成する。動作5100で、構造タグ「DecoderSpecificInfoTag」（値=5）1080を8ビット整数として出力mp4ファイルに書き込む。動作5110で、出力mp4ファイルの現在のファイル位置を量「sizePos」に割り当てる。動作5120で、値「1」を、量「numSizeBytes」に割り当てる。動作5130で、値0を、予備的なサイズ値1084として出力mp4ファイルに書き込む。

【0830】

デコーダ固有情報要素の各型のプロパティを、下で説明する。動作5150で、これらのプロパティを処理した後に、DecoderSpecificInfoオブジェクト構造1076のサイズ値（numBytes）1084を、図104（動作5160から5195）に示されているように更新する。次に、量「filePos2」の値を量「sizePos」に割り当て、DecoderConfigDescriptorオブジェクト構造1032のサイズ値（numBytes）1040を、図104（動作5160から5195）に示されているように更新する。

【0831】

「ES__Descr」要素2700に従属する「SLConfigDescriptor」要素2760ごとに、次の手順を使用して、出力mp4ファイル内でSLConfigDescriptorオブジェクト構造1028および1088を作成する。次の属性が、「SLConfigDescriptor」要素2760について定義されている：「predefined」。属性「predefined」の値が指定されている場合に、この属性の値を、量「predefined」に割り当てる。そうでない場合に、値「2」を、量「predefined」に割り当てる。

【0832】

次の値を、mp4ファイルに書き込む。

1. 「SLConfigDescrTag」（値=6）1090の値を有する1バイト
2. 値「1」（numBytes）1094を有する1バイト
3. 量「predefined」1098の値を表す1バイト

【0833】

これらの従属要素のそれぞれを処理した後に、量「filePos1」の値を、量「sizePos」に割り当て、ES__Descr構造1000のサイズ値（numBytes）1008を、図104（動作5160から5195）に示されているように更新する。

【0834】

7. 2. 4. 14 BIFS__DecoderConfig要素の処理

「BIFS__DecoderConfig」要素2720は、量「objectType」の値に応じて、「version 1」プロパティまたは「version 2」プロパティを有することができる。量「objectType」の値が「2」でない場合に、次の属性が、「BIFS__DecoderConfig」要素（version 1）について定義される：「nodeIdBits」、「routeIdBits」、「pixelMetric」、「pixelWidth」、「および「pixelHeight」。

【0835】

動作5135で、これらの属性のそれぞれについて指定された値を、同一の名前の量に

割り当てる。次に、動作 5140 で、次の値を mp4 ファイルに書き込む。

1. 量「nodeIdBits」の整数値を表す 5 ビット
2. 量「routeIdBits」の整数値を表す 5 ビット
3. 値「1」を有する 1 ビット
4. 量「pixelMetric」の boolean 値によって決定される 1 ビット（「真」の場合に「1」、そうでない場合に「0」）
5. 値「1」を有する 1 ビット
6. 量「pixelWidth」の整数値を有する 16 ビット整数
7. 量「pixelHeight」の整数値を有する 16 ビット整数
8. 値 0 を有する 3 ビット

10

【0836】

量「objectType」の値が「2」である場合に、次の属性が「BIFS_DecoderConfig」要素（version 2）について定義される。「nodeIdBits」、「routeIdBits」、「protoIdBits」、「use3DMeshCoding」、「usePredictiveMFField」、「pixelMetric」、「pixelWidth」、および「pixelHeight」。

【0837】

動作 5135 で、これらの属性のそれぞれについて指定された値を、同一の名前の量に割り当てる。次に、動作 5140 で、次の値を mp4 ファイルに書き込む。

20

1. 量「use3DMeshCoding」の boolean 値によって決定される 1 ビット（「真」の場合に「1」、そうでない場合に「0」）
2. 量「usePredictiveMFField」の boolean 値によって決定される 1 ビット（「真」の場合に「1」、そうでない場合に「0」）
3. 量「nodeIdBits」の整数値を表す 5 ビット
4. 量「routeIdBits」の整数値を表す 5 ビット
5. 量「protoIdBits」の整数値を表す 5 ビット
6. 値「1」を有する 1 ビット
7. 量「pixelMetric」の boolean 値によって決定される 1 ビット（「真」の場合に「1」、そうでない場合に「0」）
8. 値「1」を有する 1 ビット
9. 量「pixelWidth」の整数値を有する 16 ビット整数
10. 量「pixelHeight」の整数値を有する 16 ビット整数
11. 値 0 を有する 4 ビット

30

【0838】

7. 2. 4. 15 JPEG_DecoderConfig 要素の処理

次の属性が、「JPEG_DecoderConfig」要素について定義されている：「headerLength」、「Xdensity」、および「Ydensity」。

【0839】

40

同一の名前の量に、0、1、および 1 のデフォルト値を割り当てる。動作 5135 で、これらの属性のそれぞれについて指定された値を、同一の名前の量に割り当てる。

【0840】

「1」または「3」の値を、このトラックに関連するメディア・データの内容に基づいて、量「numComponents」に割り当てる。動作 5135 で、関連するメディア・データがグレイスケール・イメージを表す場合に、値「1」を量「numComponents」に割り当て、そうでない場合には、値「3」を量「numComponents」に割り当てる。

【0841】

動作 5140 で、次の値を mp4 ファイルに書き込む。

50

1. 量「headerLength」の値を表す16ビット整数
2. 量「Xdensity」の値を表す16ビット整数
3. 量「Ydensity」の値を表す16ビット整数
4. 量「numComponents」の整数値を表す1バイト

【0842】

7. 2. 4. 16 VisualConfig要素の処理

次の属性が、「VisualConfig」要素について定義されている：「profile_and_level_indication」

【0843】

同一の名前の属性に、1のデフォルト値を割り当てる。この属性の値が指定されている場合には、動作5135で、指定された値を、同一の名前の量に割り当てる。 10

【0844】

このタイプのデコーダ固有情報のさらなる処理は、このトラックのmdatアトムが作成された時に対応するメディア・データから抽出された「mediaHeader」データに依存する。

【0845】

メディア・ヘッダ・データが、このトラックについて入手可能である場合に、「ビジュアル・オブジェクト・シーケンス開始コード」(0x000001b0)および「ビジュアル・オブジェクト・シーケンス終了コード」(0x000001b1)の存在について、メディア・データをテストする。 20

【0846】

メディア・ヘッダ・データが、このトラックについて入手可能であるが、メディア・データに、「ビジュアル・オブジェクト・シーケンス開始コード」が含まれない場合、またはメディア・ヘッダ・データが、このトラックについて入手可能でない場合に、動作5140で、次の値をmp4ファイルに書き込む。

1. 値0x000001b0(ビジュアル・オブジェクト・シーケンス開始コード)を有する32ビット整数

2. 量「profile_and_level_indication」の値を表す1バイト

3. 値0x000001b5(ビジュアル・オブジェクト開始コード)を表す32ビット整数 30

4. 値「9」を有する1バイト

【0847】

メディア・ヘッダ・データが、このトラックについて入手可能でない場合に、動作5140で、次の値をmp4ファイルに書き込む。

5. 値0x00000100(ビジュアル・オブジェクト開始コード)を有する32ビット整数

【0848】

メディア・ヘッダ・データが、このトラックについて入手可能であり、メディア・ヘッダ・データの最後の4バイトが、「ビジュアル・オブジェクト・シーン終了コード」でない場合に、動作5140で、メディア・ヘッダ・データの残りをmp4ファイルにコピーする。 40

【0849】

メディア・ヘッダ・データが、このトラックについて入手可能であり、メディア・ヘッダ・データの最後の4バイトが、「ビジュアル・オブジェクト・シーケンス終了コード」からなる場合に、動作5140で、最後の4バイトを除くメディア・ヘッダ・データの残りを、mp4ファイルにコピーする。

【0850】

7. 2. 4. 17 AudioConfig要素の処理

動作5135で、「AudioConfig」要素の属性は指定されない。mp4ファ 50

イルに書き込まれるすべてのデータ値は、このトラックのメディア・データから導出されたメディア・ヘッダ・データから導出される。

【0851】

メディア・ヘッダ・データの第1バイトを、量「audioObjectType」に割り当てる。メディア・ヘッダ・データの第2バイトを、量「sampleRateIndex」に割り当てる。メディア・ヘッダ・データの第3バイトを、量「channelConfig」に割り当てる。動作5140で、次の値をmp4ファイルに書き込む。

1. 量「audioObjectType」の整数値を表す5ビット
2. 量「sampleRateIndex」の整数値を表す4ビット
3. 量「channelConfig」の整数値を表す4ビット
4. 値0を有する3ビット

10

【0852】

量「channelConfig」の値が0である場合に、「プログラム構成要素」(PCE)を、デコーダ固有情報に追加する。PCEの構造は、オーディオ・データのエンコードに関してMPEG-4仕様で定義されている。PCEを作成するのに必要なデータは、メディア・ヘッダ・データ配列に含まれる。このデータは、対応するmdatアトムが作成された時にこの配列に保管されたデータである。

【0853】

7. 2. 4. 18 メディア・ヘッダ要素の処理

標準XML手段を使用して、図68に示された「mdia」要素2604に従属する単一のメディア・ヘッダ(「*mhd」)要素2632を入手する。メディア・ヘッダ要素2632は、「nmhd」要素(sdsmまたはodsm)、「smhd」要素(オーディオ・ストリーム)、または「vmhd」要素(ビデオ・ストリーム)である可能性がある。

20

【0854】

メディア・ヘッダ要素2632が「nmhd」要素である場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「nmhd」を有するアトム936を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「nmhd」を出力mp4ファイルに書き込む。

30

【0855】

「nmhd」要素は、属性「version」および「flags」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。

【0856】

動作5040で、次の量の値を出力mp4ファイルに書き込む。

1. 量「version」の値を表す8ビット整数
2. 量「flags」の値を表す24ビット整数

【0857】

メディア・ヘッダ要素2632が「smhd」要素である場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「smhd」を有するアトム936を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「smhd」を出力mp4ファイルに書き込む。

40

【0858】

「smhd」要素は、属性「version」、「flags」、および「balance」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。

【0859】

50

動作5040で、次の量の値を出力mp4ファイルに書き込む。

1. 量「version」の値を表す8ビット整数
2. 量「flags」の値を表す24ビット整数
3. 量「balance」の値を表す16ビット整数
4. 0の値を有する16ビット整数。

【0860】

メディア・ヘッダ要素2632が「vmhd」要素である場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「vmhd」を有するアトム936を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「vmhd」を出力mp4ファイルに書き込む。 10

【0861】

「vmhd」要素は、属性「version」、「flags」、「transferMode」、「opColorRed」、「opColorGreen、および「opColorBlue」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。

【0862】

動作5040で、次の量の値を出力mp4ファイルに書き込む。

1. 量「version」の値を表す8ビット整数 20
2. 量「flags」の値を表す24ビット整数
3. 量「transferMode」の値を表す16ビット整数
4. 量「opColorRed」の値を表す16ビット整数
5. 量「opColorGreen」の値を表す16ビット整数
6. 量「opColorBlue」の値を表す16ビット整数

【0863】

動作5050で、メディア・ヘッダ要素2632は、従属要素を有しない。メディア・ヘッダ・アトムのプロパティ値を書き込んだ後に、メディア・ヘッダ・アトム936のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。 30

【0864】

7. 2. 4. 19 tref要素の処理

標準XML手段を使用して、図68に示された「trak」要素2600におそらく従属する単一の「tref」要素2636を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「tref」を有するアトム940を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「tref」を出力mp4ファイルに書き込む。「sizePos」の値を、量「trefSizePos」に割り当てる。

【0865】

動作5030および5040で、「tref」要素は属性を有しない。 40

【0866】

標準XML手段を使用して、「tref」要素2636に従属する各要素を入手する。これには、図68に示された、「mpod」要素2640が含まれる可能性がある。「dpnd」要素または「sync」要素あるいはこの両方を含む他のタイプの要素も、「tref」要素2636への従属要素として現れる可能性がある。

【0867】

「tref」要素2636に従属する「mpod」、「dpnd」、または「sync」要素2640ごとに、図103に示された手順を使用して、出力mp4ファイル内で同一の値のアトムIDを有するアトム942を作成する。動作5000で、出力mp4ファ 50

イルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「mpod」、「dpnd」、または「sync」を出力mp4ファイルに書き込む。

【0868】

各「mpod」、「dpnd」、または「sync」要素は、「trackID」という名前の1つの属性を有する。この属性は、動作5030で、trackID値のリストからなる。動作5040で、このリストの各trackID値のそれぞれを、32ビット整数として出力mp4ファイルに書き込む。また、「mpod」要素の場合に、各trackID値を、リスト「TrackIdForOdId」の項目に割り当てる。

10

【0869】

「mpod」、「dpnd」、または「sync」要素は、動作5050で、従属要素を有しない。この要素のtrackID属性を処理した後に、mp4ファイルの対応するアトム942のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0870】

「tref」要素2636に従属するすべての要素2640の処理を完了した後に、量「trefSizePos」の値を「sizePos」に割り当て、trefアトム940のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

20

【0871】

7. 2. 4. 20 edts要素の処理

標準XML手段を使用して、図68に示された「trak」要素2600におそらく従属する単一の「edts」要素2644を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「edts」を有するアトム945を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「edts」を出力mp4ファイルに書き込む。「sizePos」の値を量「edtsSizePos」に割り当てる。

【0872】

「edts」要素2644は、動作5030および5040で属性を有しない。

30

【0873】

標準XML手段を使用して、図68に示された「edts」要素2644に従属する単一の「elst」要素2648を入手する。図103に示された手順を使用して、出力mp4ファイル内でアトムID「elst」を有するアトム948を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「elst」を出力mp4ファイルに書き込む。

【0874】

「elst」要素は、属性「version」および「flags」を有する。動作5030で、これらの属性のそれぞれの値を、同一の名前の量に割り当てる。

40

【0875】

動作5040で、次の量の値を出力mp4ファイルに書き込む。

1. 量「version」の値を表す8ビット整数
2. 量「flags」の値を表す24ビット整数

【0876】

次に、標準XML手段を使用して、「elst」要素2648に従属する各要素を入手する。「elst」要素2648に従属する要素の組は、2つの「segment」要素からなると期待される。各「segment」要素は、3つの属性「duration」、「startTime」、および「rate」を有する。

50

【0877】

「e l s t」要素2648に従属するsegment要素ごとに、次の動作を実行する。

1. 属性「duration」、「startTime」、および「rate」のそれぞれの値を、同一の名前の量に割り当てる

2. 量「duration」の値を、32ビット整数として出力mp4ファイルに書き込む

3. 量「startTime」の値を、32ビット整数として出力mp4ファイルに書き込む

4. 量「rate」の浮動小数点値に、 $256 * 256$ をかけ、整数に変換し、その結果を32ビット整数としてmp4ファイルに書き込む。 10

【0878】

「e l s t」要素2648に従属するすべての「segment」要素を処理した後に、e l s tアトム948のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0879】

量「edtsSizePos」の値を「sizePos」に割り当て、edtsアトム945のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0880】

7. 2. 5 任意選択のユーザ・データ要素の処理
出力mp4ファイル2230作成の第5ステップは、「moov」要素2320に含まれるすべての任意選択の「ユーザ・データ」（udta）要素2340の処理からなる。標準XML手段を使用して、図59に示されたmp4file文書2300の「moov」要素2320に従属するすべての「udta」要素2340を入手する。次の手段を使用して、各そのような「udta」要素を処理する。

【0881】

図103に示された手順を使用して、出力mp4ファイル内でアトムID「udta」を有するアトム784を作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「udta」を出力mp4ファイルに書き込む。「sizePos」の値を、量「udtaSizePos」に割り当てる。 30

【0882】

「udta」要素は、動作5030および5040で属性を有しない。

【0883】

各「udta」要素は、mp4ファイルに著作権メッセージを埋め込むのに使用できる「cprt」要素などの従属要素を有する場合がある。認識されない従属要素は、すべて無視することができる。

【0884】

「udta」要素が、従属する「cprt」要素を有することがわかった場合に、図103に示された手順を使用して、出力mp4ファイル内でアトムID「cprt」を有するアトムを作成する。動作5000で、出力mp4ファイルの現在のファイル位置を、量「sizePos」に割り当てる。動作5010で、アトム・サイズ値の代わりに、値0を出力mp4ファイルに書き込む。動作5020で、アトムID「cprt」を出力mp4ファイルに書き込む。 40

【0885】

「cprt」要素は、「version」、「flags」、および「language」という名前の属性を所有することができる。これらの属性のそれぞれの値を、動作5030で同一の名前の量に割り当てる。動作5040で、「cprt」要素は、メッセー 50

ジを著すテキスト・ストリングを含む従属する「テキスト・ノード」も所有する。

【0886】

動作5040で、次の量の値を出力mp4ファイルに書き込む。

1. 量「version」の値を表す8ビット整数
2. 量「flags」の値を表す24ビット整数
3. 量「language」の値を表す16ビット整数
4. 従属するテキスト・ノードの値を表す文字のシーケンスと、それに続くヌル・バイト

【0887】

「cprt」要素の属性を完了した後に、cprtアトムのアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。 10

【0888】

「udta」要素2340に従属するすべてのmp4file要素の処理を完了した後に、量「udtaSizePos」の値を「sizePos」に割り当て、対応するudtaアトム784のアトム・サイズの値を、図103（動作5060から5095）に示されているように更新する。

【0889】

7. 2. 6 odsmバッファ・サイズの更新

出力mp4ファイル2230作成の最後のステップは、odsmバッファ・サイズの更新からなる。 20

【0890】

出力mp4ファイル2300に、メディア・ストリームごとのtrakアトム790が含まれる。図21および22からわかるように、各trakアトム900に、ES_Descrオブジェクト構造990が含まれる。各ES_Descrオブジェクト構造990および1000に、DecoderConfigDescriptorオブジェクト構造1024および1032が含まれる。図23から26からわかるように、各DecoderConfigDescriptorオブジェクト構造に、プロパティ「bufferSizeDB」1060が含まれる。ほとんどの場合に、このプロパティの値は、関連するメディア・データ・ストリームの最大のサンプルのサイズによって決定される。odsm1900の場合に、各サンプル1920および1940に、ES_Descr構造1000への参照を有するEsIdRef構造2160を含めることができ、odsmサンプル・バッファは、埋め込まれたEsIdRef2160構造のそれぞれを対応するES_Descr構造1000に置換できるようにするのに十分なサイズを有しなければならない。ES_Descr構造1000を含むバイト数は、一般に、対応するEsIdRef構造2160より多い。その結果、odsmの最小バッファ・サイズを増やして、EsIdRef構造2160を対応するES_Descr構造1000によって置換できるようにしなければならない。これは、下で説明する手段によって達成される。 30

【0891】

これらの動作を実行する前に、各trakアトム900が、上で説明したように構成されている。odsmのtrakアトムに、bufferSizeDBプロパティの予備的な値が含まれる。この予備的な値を出力mp4ファイルに書き込む前に、mp4ファイル位置の値が、量「OdsmBufferSizePos」に割り当てられた。さらに、上で説明した動作の一部として、各odsmサンプルのサイズが、リスト「OdsmSampleSize」の項目に割り当てられ、各trakアトムのES_Descr構造のサイズが、リスト「EsDescrSizeForTrack」の項目に割り当てられ、各trakアトムの「trackID」プロパティの値が、リスト「TrackIdForTrack」の項目に割り当てられ、各オブジェクトに関連するtrackIDの値が、リスト「TrackIdForOdId」の項目に割り当てられた。 40

【0892】

上で説明した先行ステップを完了した後に、次の手段を使用して、odsmのプロパティ 50

ィbufferSizeDB1060の値を改訂する。

1. 標準XML手段を使用して、mp4file文書2300の各「mdat」要素2310を入手する。
2. 標準XML手段を使用して、各「mdat」要素2310および2400に従属する各「odsm」要素2420を入手する
3. 標準XML手段を使用して、各「odsm」要素2420および2460に従属する各「odsmChunk」要素2470を入手する。
4. 標準XML手段を使用して、各「odsmChunk」要素2470および2500に従属する各「odsmSample」要素(2510)を入手する。
5. 量「ithOdsmSample」を用いて各「odsmSample」要素2510を列挙する。 10
6. 標準XML手段を使用して、各「odsmSample」要素2520および2510に従属する各「ObjectDescrUpdate」odsm-command要素2530および2540を入手する。
7. 標準XML手段を使用して、各「ObjectDescrUpdate」要素2540に従属する各「ObjectDescriptor」要素2550を入手する。
8. 「ObjectDescriptor」要素2550の「ODID」属性の値を、量「OdId」に割り当てる。
9. リスト「TrackIdForOdId」の項目OdId-1の値を、量「trackID」に割り当てる。 20
10. リスト「TrackIdForTrack」を使用して、「trackID」のこの値の「トラック」インデックスを決定する。
11. リスト「EsDescrSizeForTrack」の項目「track」の値を、量「EsDescrSize」の値に割り当てる。
12. 量「EsDescrSize」の値を、リスト「OdsmSampleSize」の項目「IthOdsmSample」に加算する。
13. リスト「OdsmSampleSize」の最大の項目を判定し、その結果を量OdsmBufferSizeに割り当てる。
14. 現在のmp4ファイル位置を、量「mp4FilePos」に割り当てる。
15. 現在のmp4ファイル位置を、量「OdsmBufferSizePos」によって指定される値に変更する。 30
16. 量OdsmBufferSizeの値を24ビット整数として表す3バイトを、出力mp4ファイルに書き込む。
17. mp4ファイル位置を、量「mp4FilePos」によって指定される値に復元する。

【0893】

量OdsmBufferSizeの値は、odsmバッファ・サイズに必要な値を過大評価する可能性があるが、これは許容可能である。この手段によって、出力mp4ファイルの作成が完了する。

【0894】

本発明の前述の説明は、例示および説明のために提示されたものである。網羅的であることまたは本発明を開示された形態に制限することは意図されておらず、他の修正形態および変形形態が、上の教示に鑑みて可能である。たとえば、XMT-Aファイルの正確な定義が、時と共に変化または進化する可能性がある。同様に、MPEG-4 Intermedia Formatの正確な定義が、時と共に変化または進化する可能性がある。本明細書に記載の発明は、上に示した文書で指定される特定の定義に制限されない。したがって、本発明の原理は、他の関連しないデータ構造にも適用することができる。SFNodeデータ構造の他の拡張形式も、MPEG-4 Systems仕様で定義されている。そのような事例を含むために次の実施形態で説明されるように本発明を拡張する手段が、当業者に明白になる。したがって、開示された実施形態は、本発明の原理およびその 40 50

実用的な応用例を最もよく説明し、これによって、当業者が、企図される特定の使用に適するようにさまざまな実施形態およびさまざまな修正形態で本発明を最もよく利用できるようにするために、選択され、説明された。請求項が、従来技術によって制限される範囲を除いて、本発明の他の代替実施形態を含むように解釈されることが意図されている。

【図面の簡単な説明】

【0895】

【図1】本発明の1実施形態によって使用される例示的なXMT-A文書を示す図である。

【図2】例示的なXMT-A Initial Object Descriptorを示す図である。

10

【図3】例示的なXMT-A par要素を示す図である。

【図4】例示的なXMT-A odsmコマンド要素を示す図である。

【図5】例示的なXMT-A Insertコマンドを示す図である。

【図6】例示的なXMT-A Deleteコマンドを示す図である。

【図7】例示的なXMT-A Replaceコマンドを示す図である。

【図8】例示的なXMT-A BIFS Node要素を示す図である。

【図9】例示的なXMT-A BIFS Nodeを示す図である。

【図10】例示的な再利用されるXMT-A BIFS Nodeを示す図である。

【図11】例示的なXMT-A Object Descriptorを示す図である。

【図12】例示的なXMT-A ES_Descriptorを示す図である。

20

【図13】sdsm(BIFS)に関する例示的なDecoder Specific Infoを示す図である。

【図14】本発明の1実施形態によって生成される例示的なmp4バイナリ・ファイルを示す図である。

【図15】例示的なmdatアトムを示す図である。

【図16】例示的なチャンクを示す図である。

【図17】例示的なmoovアトムを示す図である。

【図18】例示的なmp4ファイルiodsアトムを示す図である。

【図19】例示的なMp4fInitObjectDescrを示す図である。

【図20】例示的なES_ID_Incを示す図である。

30

【図21】例示的なtrakアトムを示す図である。

【図22】例示的なsample tablesアトムを示す図である。

【図23】例示的なバイナリES記述子を示す図である。

【図24】例示的なデコーダ構成記述子を示す図である。

【図25】例示的なデコーダ固有情報記述子を示す図である。

【図26】例示的なバイナリSL構成記述子を示す図である。

【図27】例示的なsdsmバイナリ・チャンクを示す図である。

【図28】例示的なsdsmコマンド・フレームを示す図である。

【図29】例示的なBIFS挿入コマンドを示す図である。

【図30】例示的なBIFS削除コマンドを示す図である。

40

【図31】例示的なBIFS置換コマンドを示す図である。

【図32】例示的なBIFSシーン置換コマンドを示す図である。

【図33】例示的なNode挿入コマンドを示す図である。

【図34】例示的なIndexedValue挿入コマンドを示す図である。

【図35】例示的なRoute挿入コマンドを示す図である。

【図36】例示的なNode削除コマンドを示す図である。

【図37】例示的なIndexedValue削除コマンドを示す図である。

【図38】例示的なRoute削除コマンドを示す図である。

【図39】例示的なNode置換コマンドを示す図である。

【図40】例示的なField置換コマンドを示す図である。

50

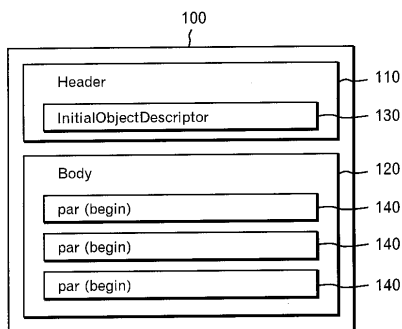
- 【図 4 1】 例示的な Indexed Value 置換コマンドを示す図である。
- 【図 4 2】 例示的な Route 置換コマンドを示す図である。
- 【図 4 3】 例示的な B I F S S c e n e を示す図である。
- 【図 4 4】 例示的な S F N o d e （再利用される）を示す図である。
- 【図 4 5】 例示的な S F N o d e （マスク・ノード）を示す図である。
- 【図 4 6】 例示的な S F N o d e （リスト・ノード）を示す図である。
- 【図 4 7】 例示的な M F F i e l d （リスト形式）を示す図である。
- 【図 4 8】 例示的な M F F i e l d （ベクトル形式）を示す図である。
- 【図 4 9】 例示的な R o u t e s （リスト形式）を示す図である。
- 【図 5 0】 例示的な R o u t e s （ベクトル形式）を示す図である。 10
- 【図 5 1】 例示的な R o u t e を示す図である。
- 【図 5 2】 例示的な o d s m バイナリ・チャンクを示す図である。
- 【図 5 3】 例示的な o d s m バイナリ・サンプルを示す図である。
- 【図 5 4】 例示的な O b j e c t D e s c r i p t o r U p d a t e コマンドを示す図である。
- 【図 5 5】 例示的な O b j e c t D e s c r i p t o r R e m o v e コマンドを示す図である。
- 【図 5 6】 例示的な バイナリ・オブジェクト記述子を示す図である。
- 【図 5 7】 例示的な バイナリ E s I d R e f 記述子を示す図である。
- 【図 5 8】 本発明によって企図される、例示的な X M T - A から M P E G - 4 への中間フ 20
ァイル・コンバータを示す図である。
- 【図 5 9】 例示的な m p 4 f i l e 文書を示す図である。
- 【図 6 0】 例示的な m p 4 f i o d s 文書を示す図である。
- 【図 6 1】 例示的な m d a t 要素を示す図である。
- 【図 6 2】 例示的な s d s m 要素を示す図である。
- 【図 6 3】 例示的な o d s m 要素を示す図である。
- 【図 6 4】 例示的な m e d i a F i l e 要素を示す図である。
- 【図 6 5】 例示的な o d s m C h u n k 要素を示す図である。
- 【図 6 6】 例示的な o d s m S a m p l e 要素を示す図である。
- 【図 6 7】 例示的な o d s m コマンド要素を示す図である。 30
- 【図 6 8】 例示的な t r a k 要素を示す図である。
- 【図 6 9】 例示的な s t b l 要素を示す図である。
- 【図 7 0】 例示的な E S _ D e s c r を示す図である。
- 【図 7 1】 例示的な m p 4 b i f s 文書を示す図である。
- 【図 7 2】 例示的な m p 4 b i f s c o m m a n d F r a m e 要素を示す図である。
- 【図 7 3】 例示的な m p 4 b i f s b i f s C o m m a n d 要素を示す図である。
- 【図 7 4】 例示的な m p 4 b i f s R e p l a c e S c e n e 要素を示す図である。
- 【図 7 5】 例示的な m p 4 b i f s のオリジナル N o d e 要素を示す図である。
- 【図 7 6】 例示的な m p 4 b i f s C o n d i t i o n a l N o d e 要素を示す図で 40
ある。
- 【図 7 7】 例示的な m p 4 b i f s の再利用される N o d e 要素を示す図である。
- 【図 7 8】 X M T - A 文書の例示的な処理を示す流れ図である。
- 【図 7 9】 X M T - A H e a d e r の例示的な処理を示す流れ図である。
- 【図 8 0】 X M T - A D e s c r 要素の例示的な処理を示す流れ図である。
- 【図 8 1】 X M T - A e s D e s c r 要素の例示的な処理を示す流れ図である。
- 【図 8 2】 X M T - A E S _ D e s c r の例示的な処理を示す流れ図である。
- 【図 8 3】 例示的な m d a t 要素作成を示す流れ図である。
- 【図 8 4】 例示的な t r a k 要素作成を示す流れ図である。
- 【図 8 5】 例示的な s t b l 要素作成を示す流れ図である。
- 【図 8 6】 例示的な e s d s 要素作成を示す流れ図である。 50

- 【図 87】 例示的な B I F S 構成処理を示す流れ図である。
- 【図 88】 例示的なオブジェクト・テーブルを示す図である。
- 【図 89】 例示的な B I F S NodeID テーブルを示す図である。
- 【図 90】 例示的な B I F S RouteID テーブルを示す図である。
- 【図 91】 例示的な ReplaceScen タイム・テーブルを示す図である。
- 【図 92】 例示的なソート済みオブジェクト・テーブルを示す図である。
- 【図 93】 XMT-A Body 要素の例示的な処理（パス 1 またはパス 2）を示す流れ図である。
- 【図 94】 XMT-A par 要素の例示的な処理（パス 1）を示す流れ図である。
- 【図 95】 XMT-A コマンド要素の例示的な処理（パス 1）を示す流れ図である。
- 【図 96】 XMT-A par 要素の例示的な処理（パス 2）を示す流れ図である。
- 【図 97】 Insert コマンドの例示的な処理を示す流れ図である。
- 【図 98】 Delete コマンドの例示的な処理を示す流れ図である。
- 【図 99】 Replace コマンドの例示的な処理を示す流れ図である。
- 【図 100】 create ReplaceScene コマンドの例示的な処理を示す流れ図である。
- 【図 101】 XMT-A B I F S ノードの例示的な処理を示す流れ図である。
- 【図 102】 odsm の XML 表現の例示的な処理を示す流れ図である。
- 【図 103】 例示的な mp4 アトム構造作成の流れ図である。
- 【図 104】 例示的な mp4 オブジェクト構造作成の流れ図である。
- 【図 105】 md at 要素の例示的な処理を示す流れ図である。
- 【図 106】 mediaFile 要素の例示的な処理を示す流れ図である。
- 【図 107】 例示的な sync sample table 作成を示す流れ図である。

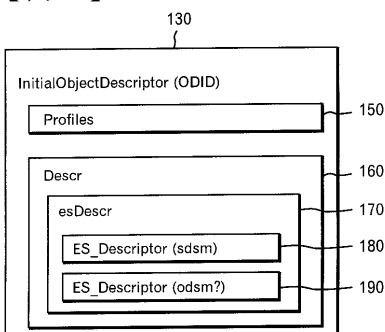
10

20

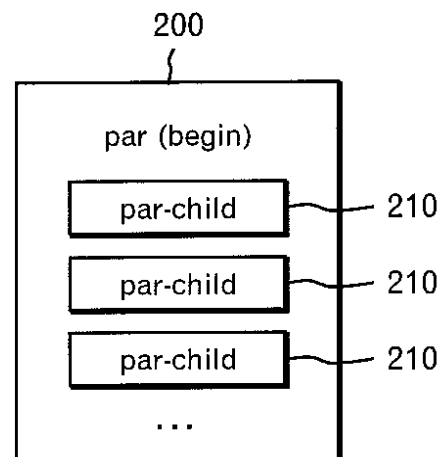
【図 1】



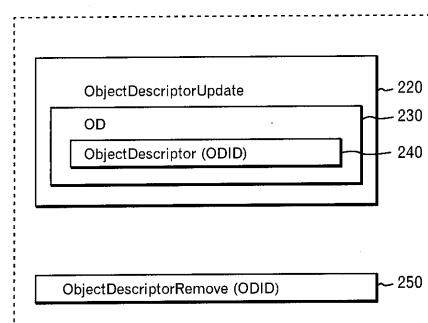
【図 2】



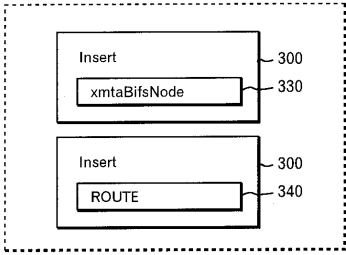
【図 3】



【図 4】



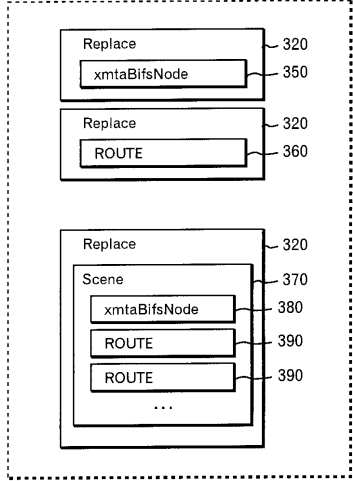
【図 5】



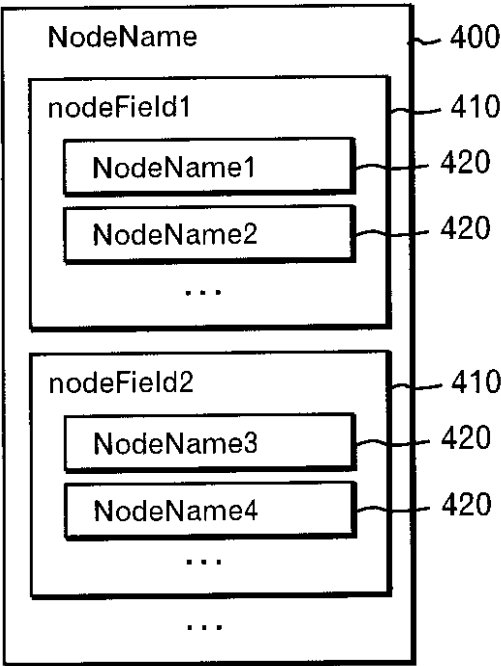
【図 6】



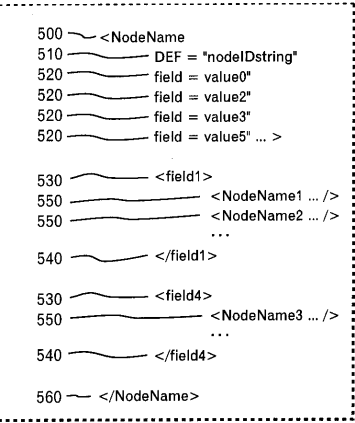
【図 7】



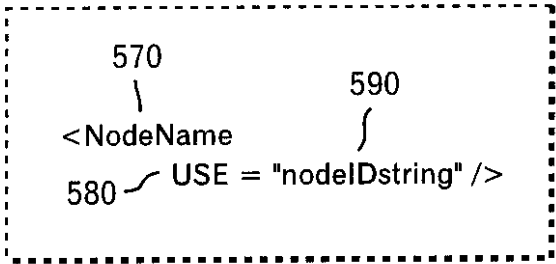
【図 8】



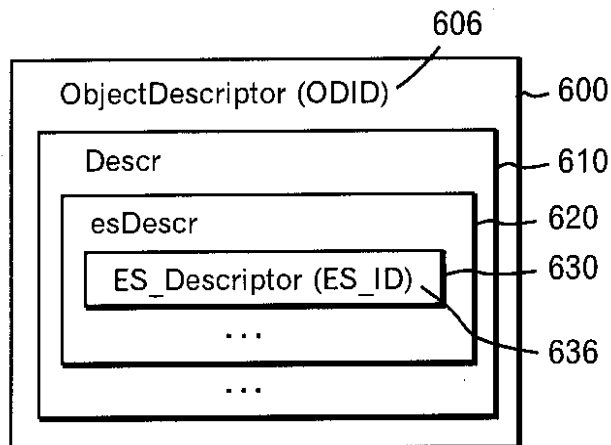
【図 9】



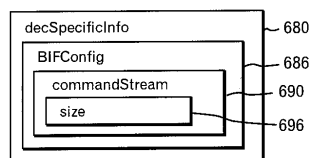
【図 10】



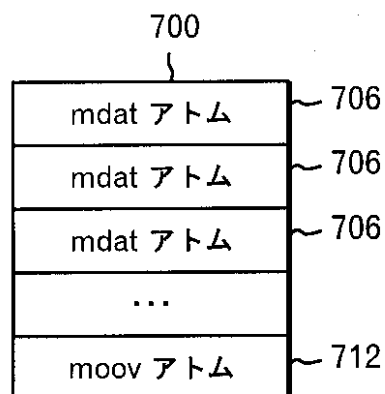
【図 1 1】



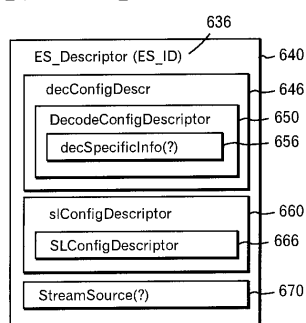
【図 1 3】



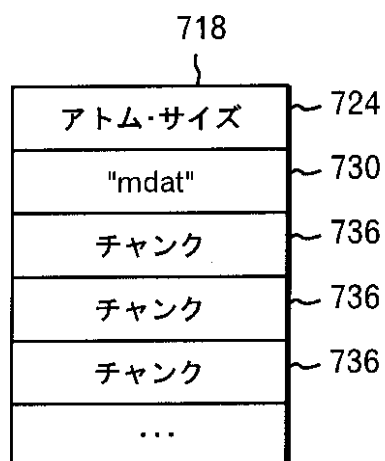
【図 1 4】



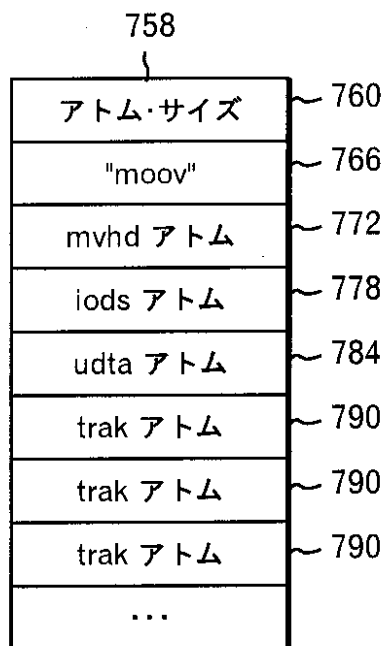
【図 1 2】



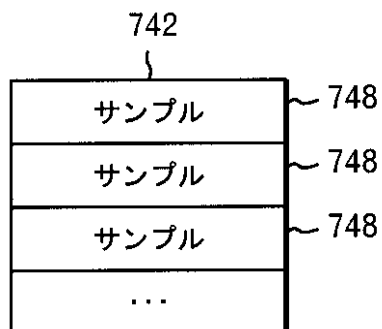
【図 1 5】



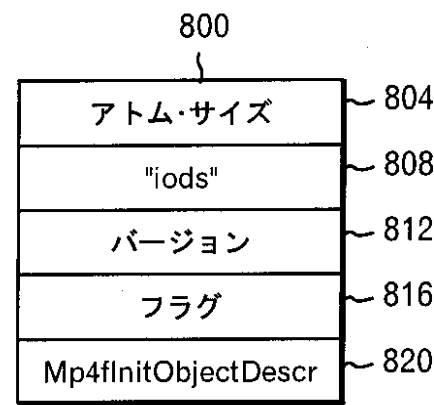
【図 1 7】



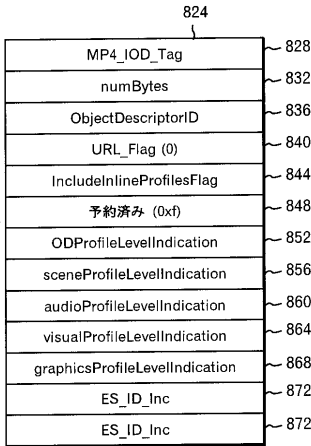
【図 1 6】



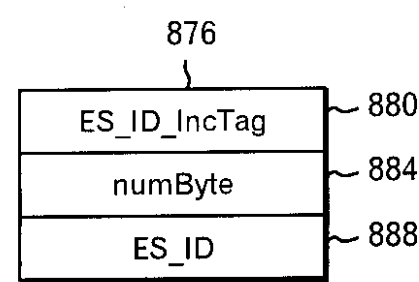
【図 18】



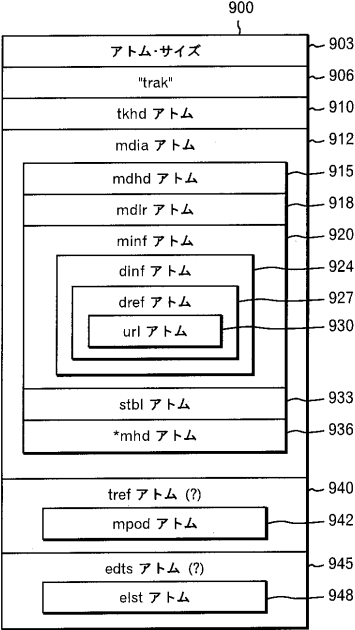
【図 19】



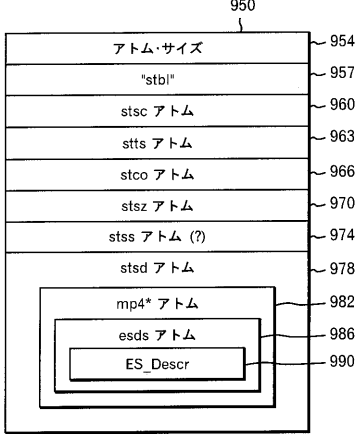
【図 20】



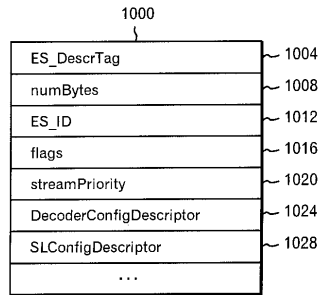
【図 21】



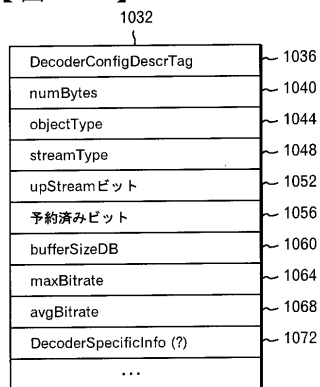
【図 22】



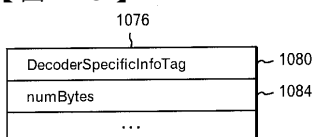
【図 23】



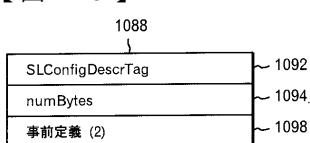
【図 2 4】



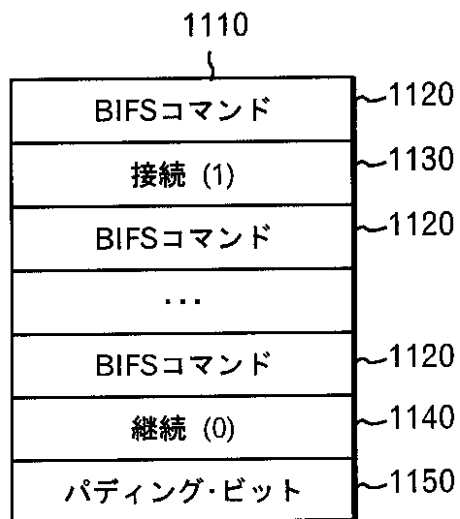
【図 2 5】



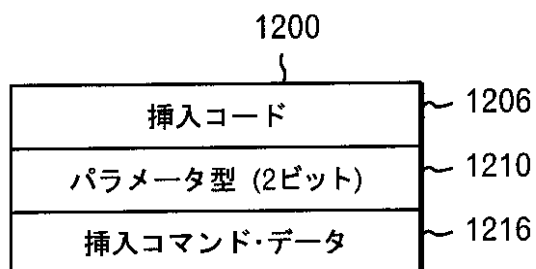
【図 2 6】



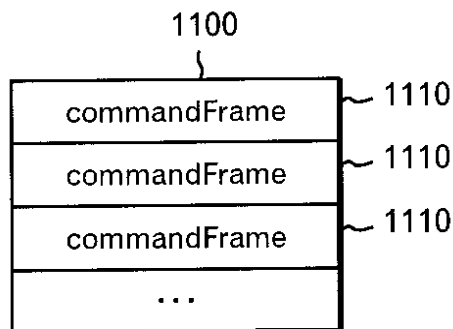
【図 2 8】



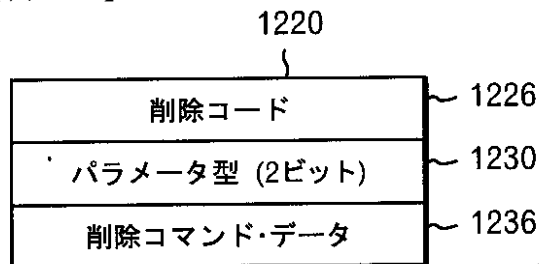
【図 2 9】



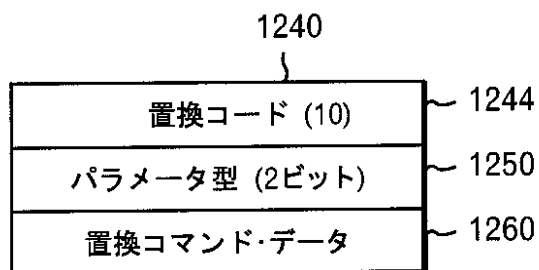
【図 2 7】



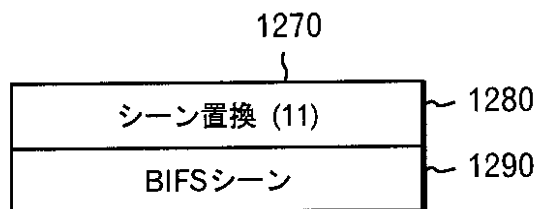
【図 3 0】



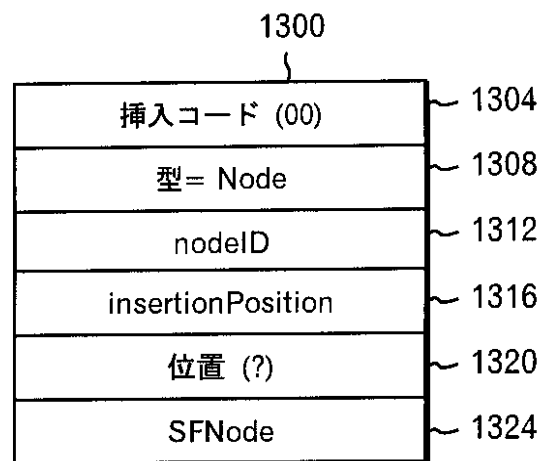
【図 3 1】



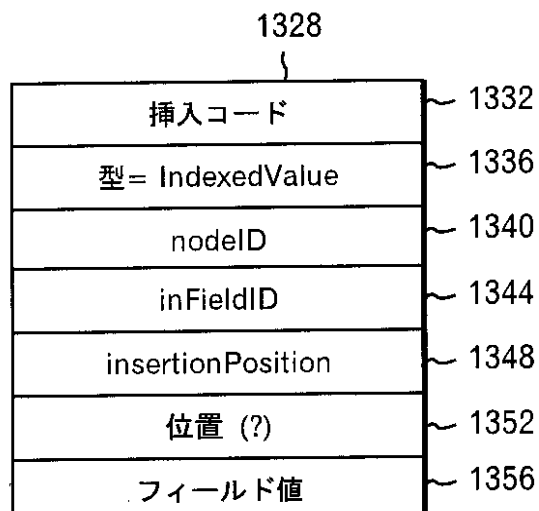
【図 3 2】



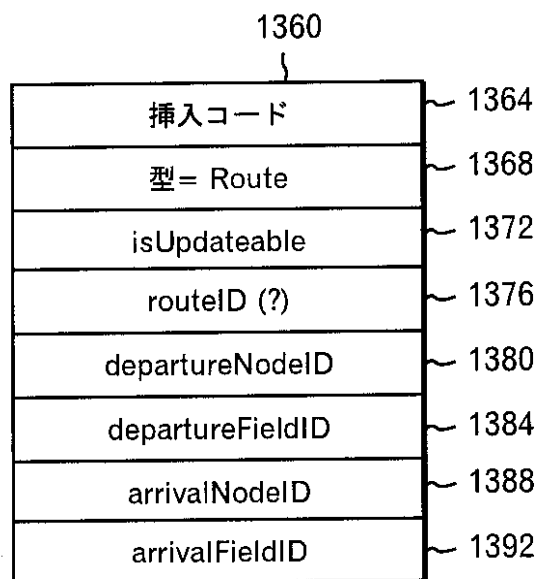
【図 3 3】



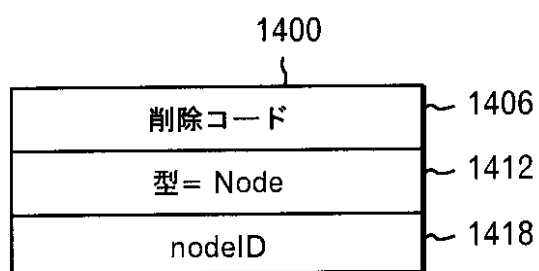
【図 3 4】



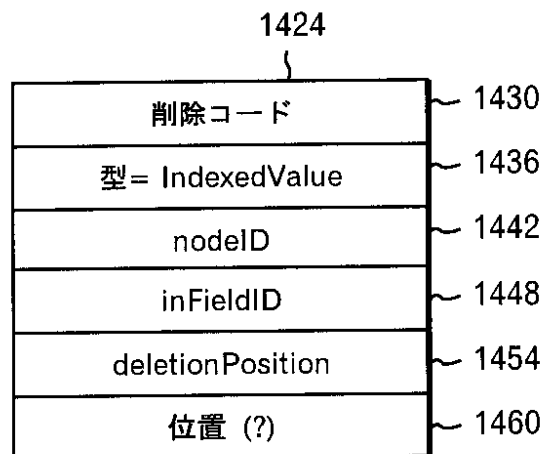
【図 3 5】



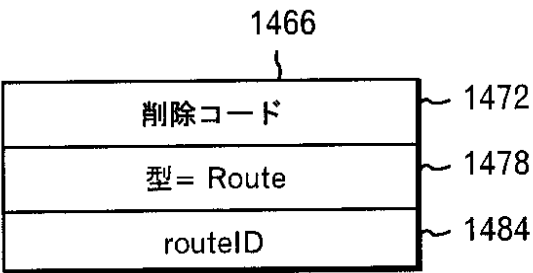
【図 3 6】



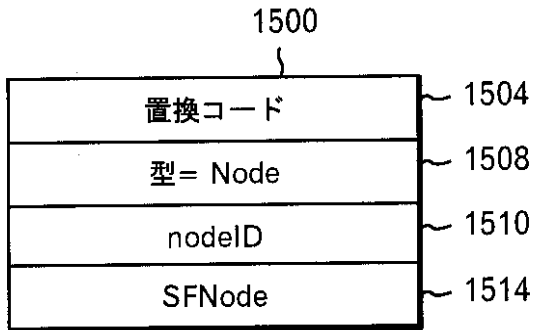
【図 3 7】



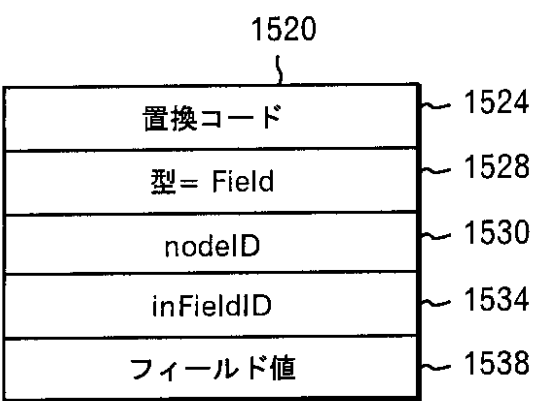
【図 3 8】



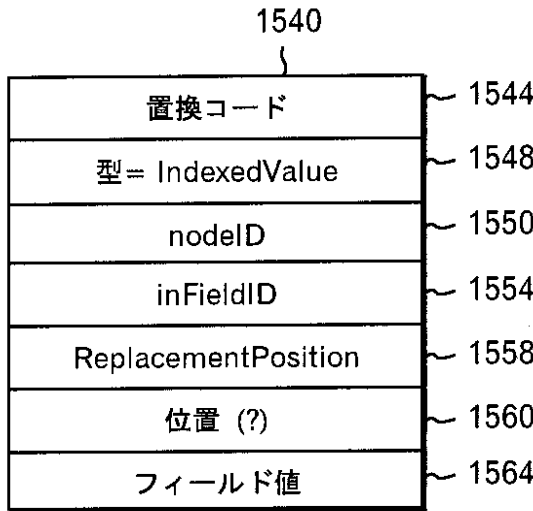
【図 3 9】



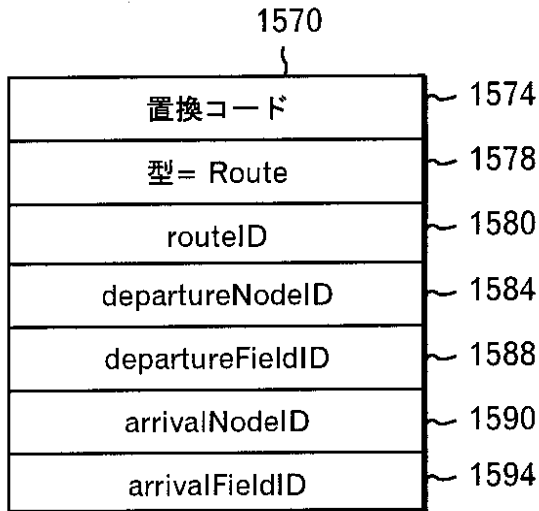
【図 4 0】



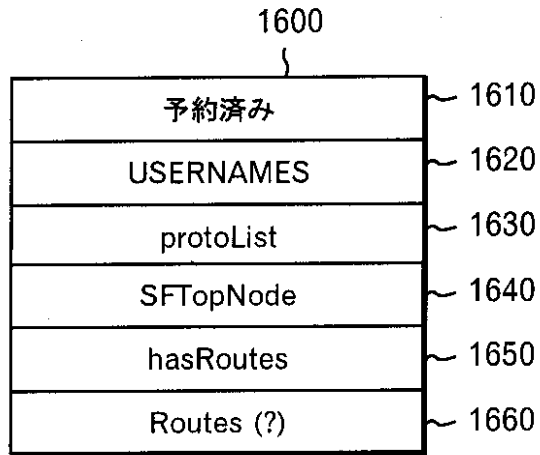
【図 4 1】



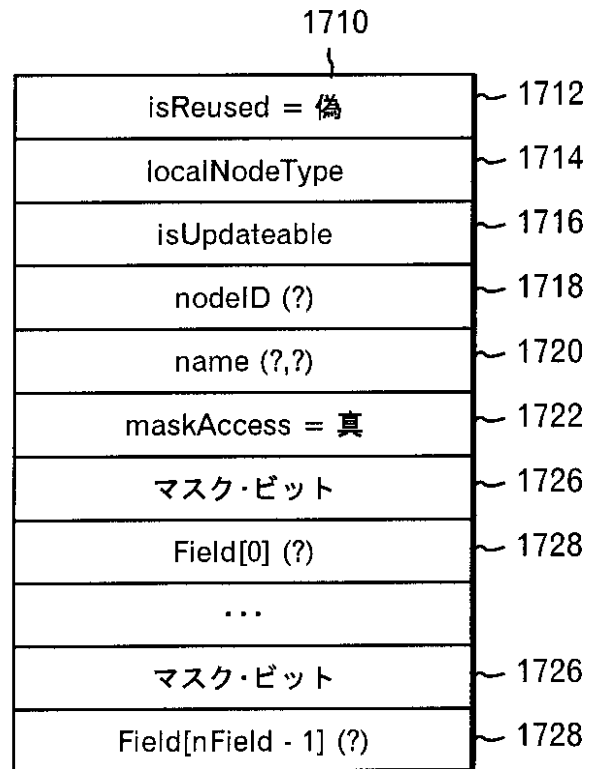
【図 4 2】



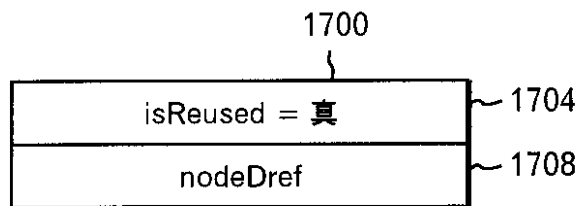
【図 4 3】



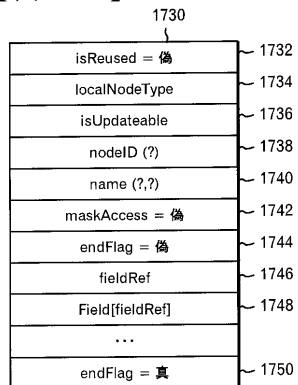
【図 4 5】



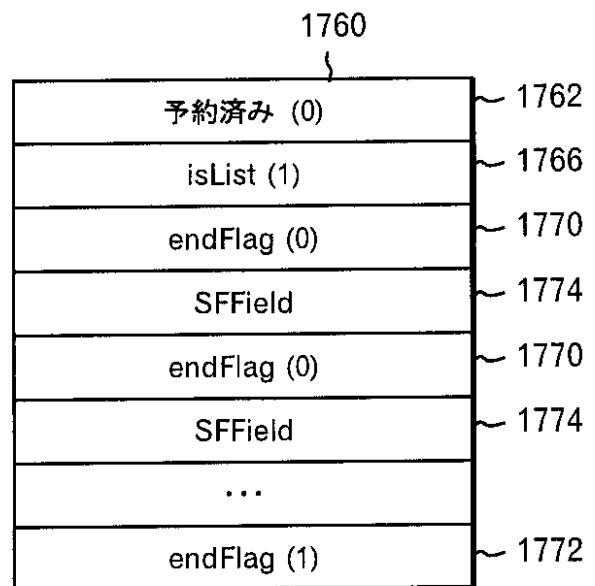
【図 4 4】



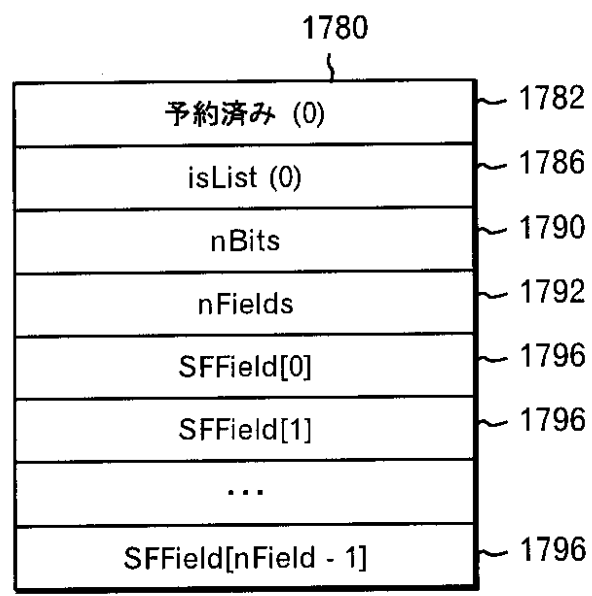
【図 4 6】



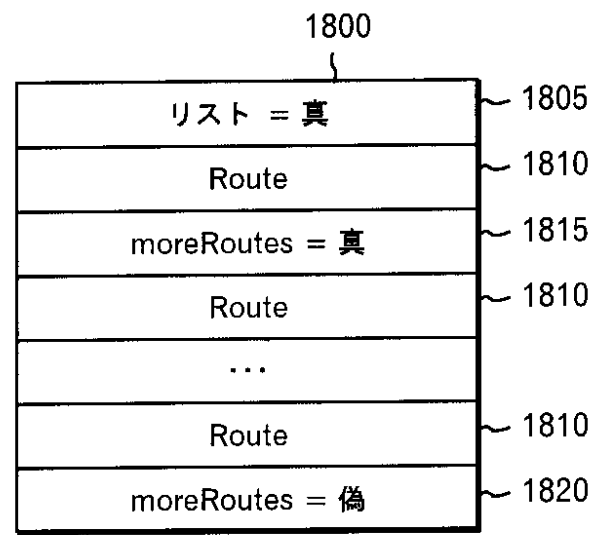
【図 4 7】



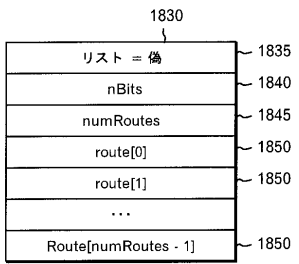
【図 4 8】



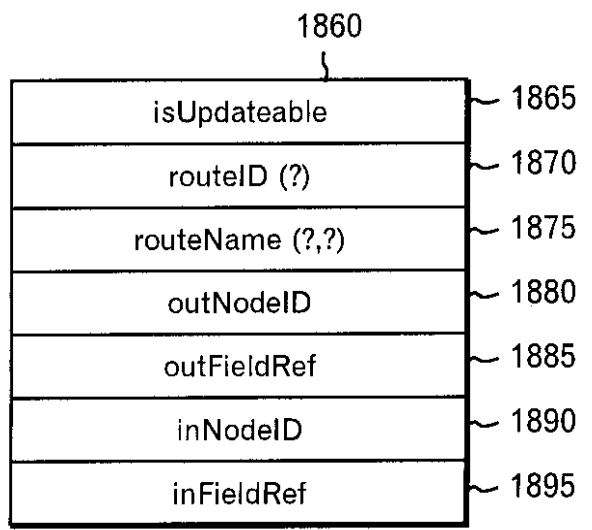
【図 4 9】



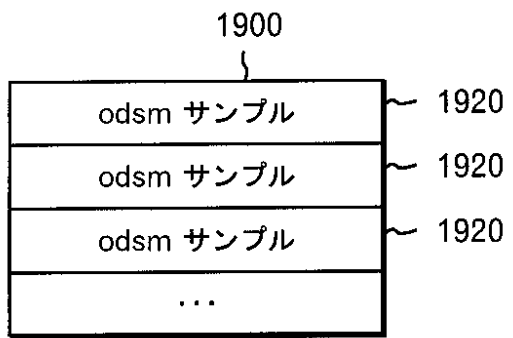
【図 5 0】



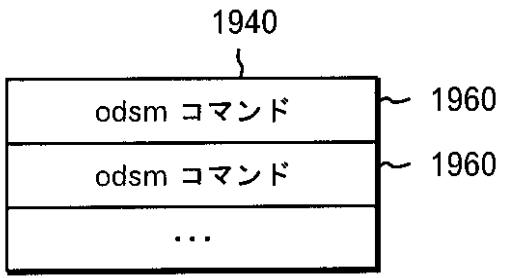
【図 5 1】



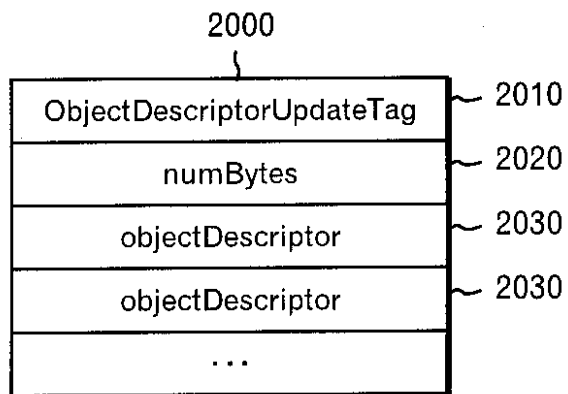
【図 5 2】



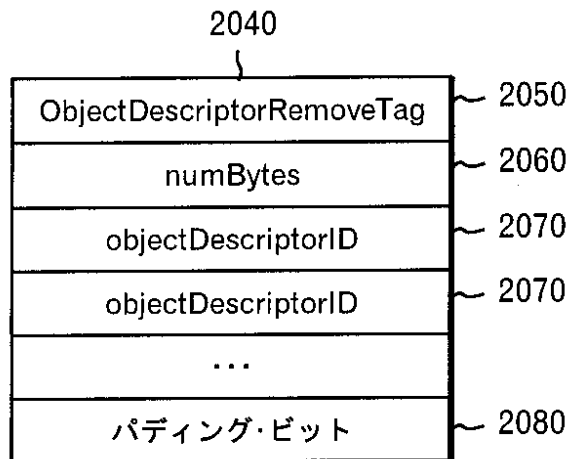
【図 5 3】



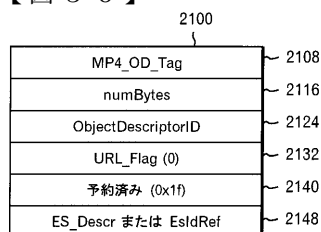
【図 5 4】



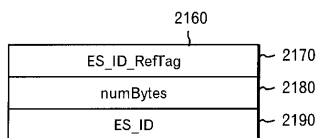
【図 5 5】



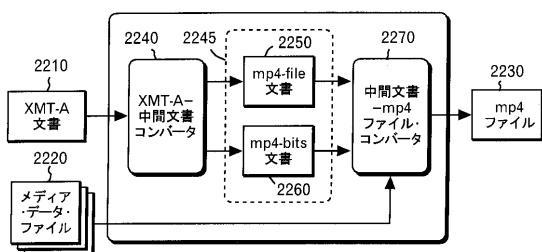
【図 5 6】



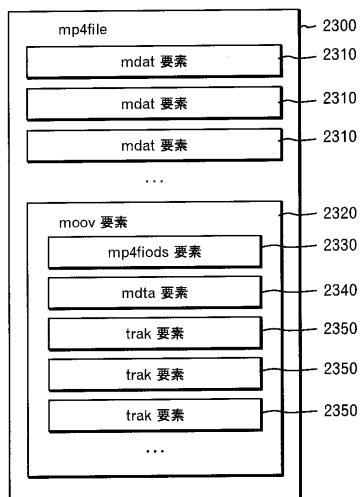
【図 5 7】



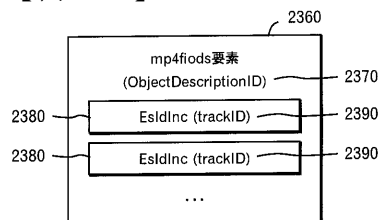
【図 5 8】



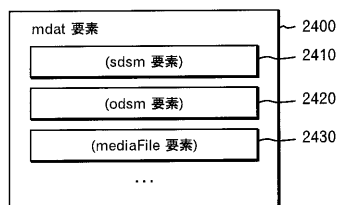
【図 5 9】



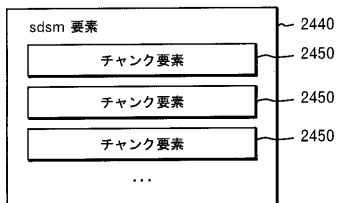
【図 6 0】



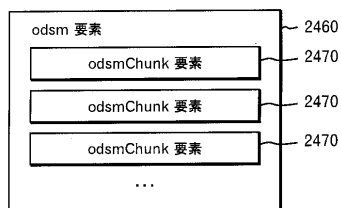
【図 6 1】



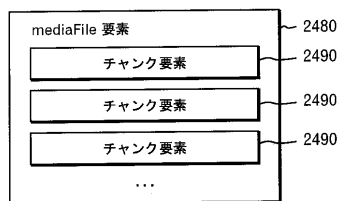
【図 6 2】



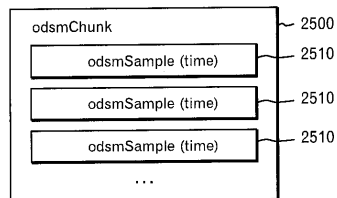
【図 6 3】



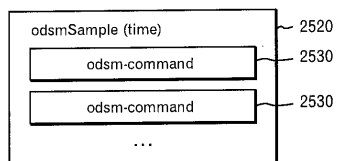
【図 6 4】



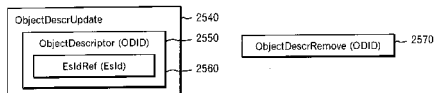
【図 6 5】



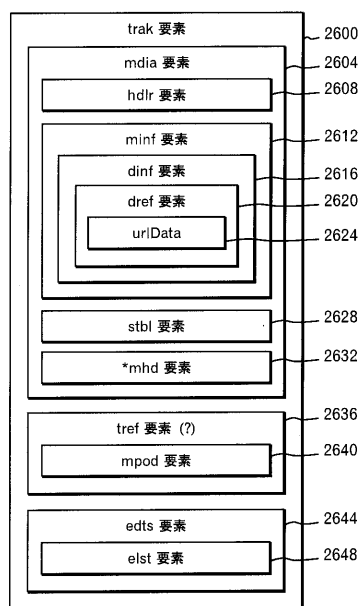
【図 6 6】



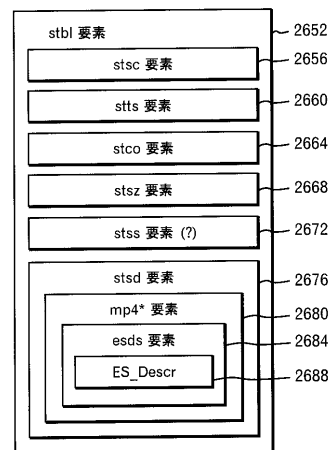
【図 6 7】



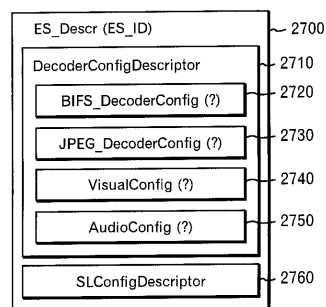
【図 6 8】



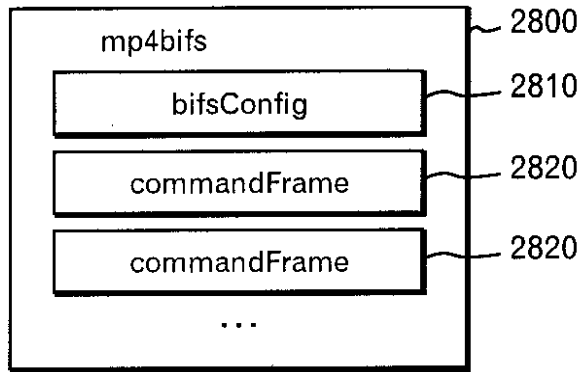
【図 6 9】



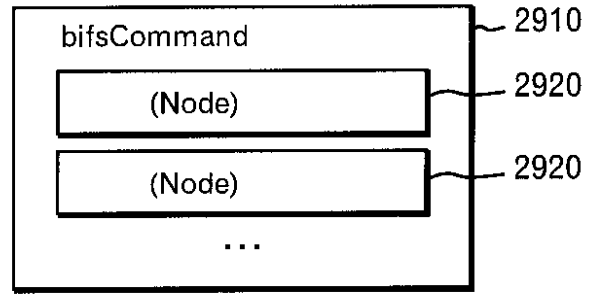
【図 7 0】



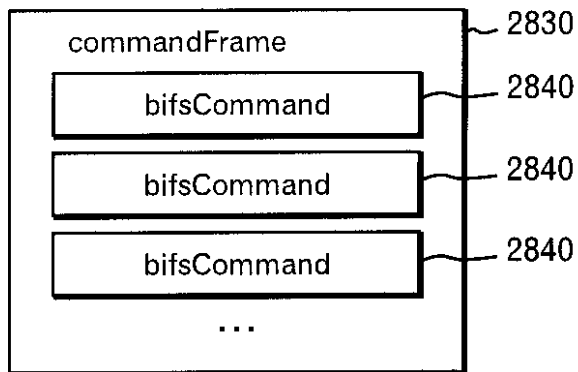
【図 7 1】



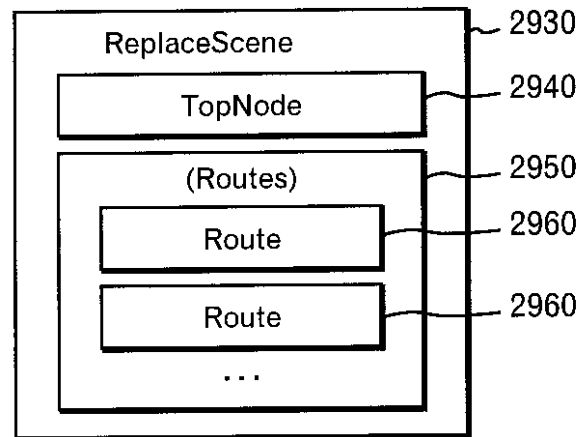
【図 7 3】



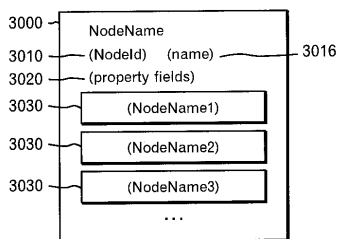
【図 7 2】



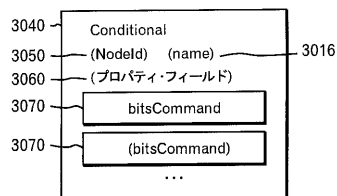
【図 7 4】



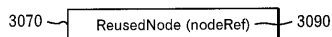
【図 7 5】



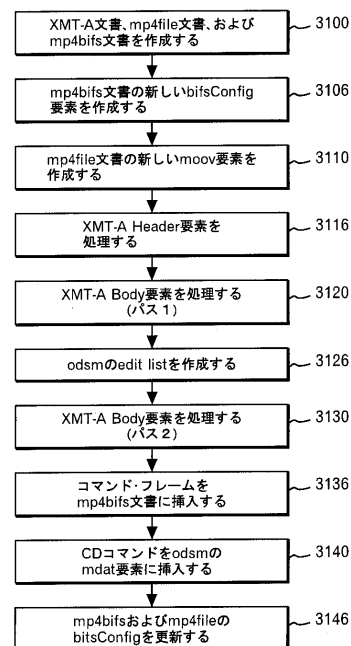
【図 7 6】



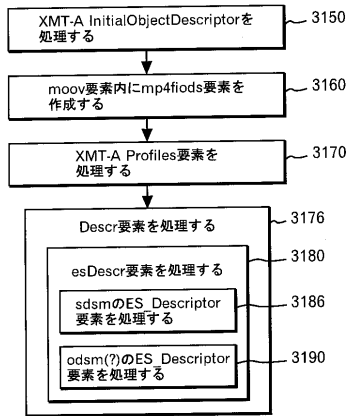
【図 7 7】



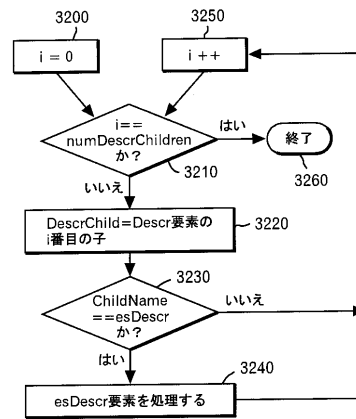
【図 7 8】



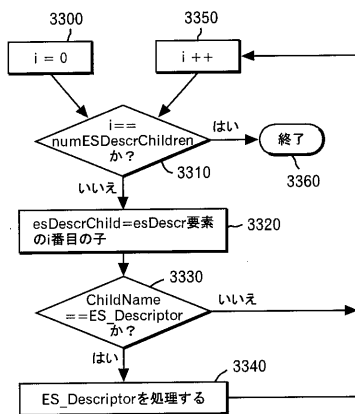
【図 79】



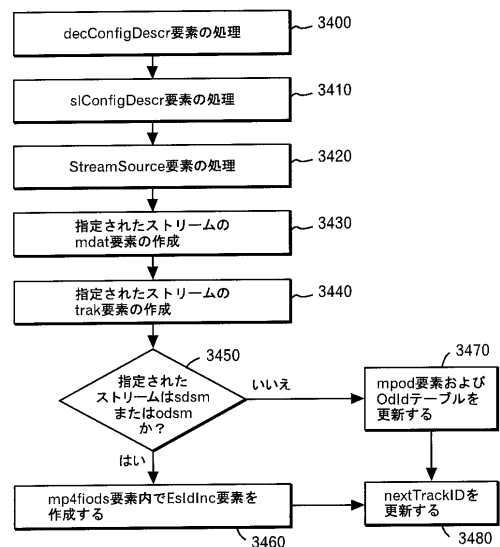
【図 80】



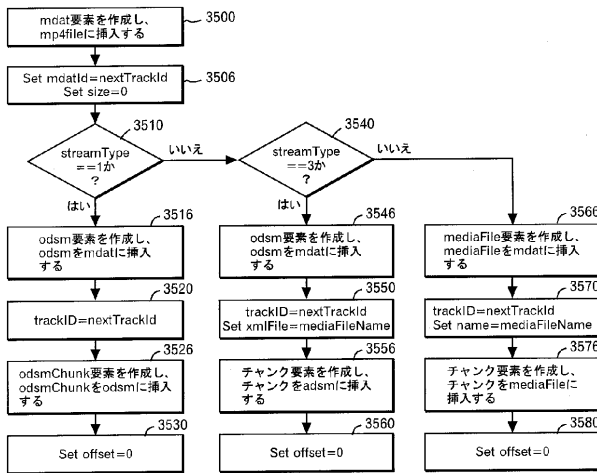
【図 81】



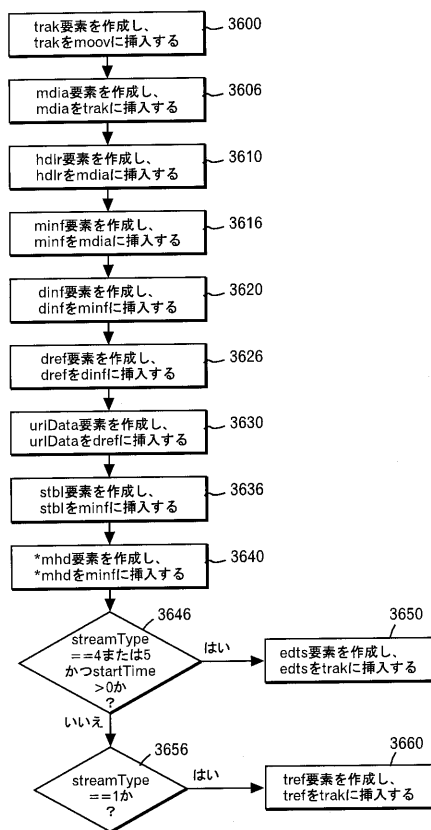
【図 82】



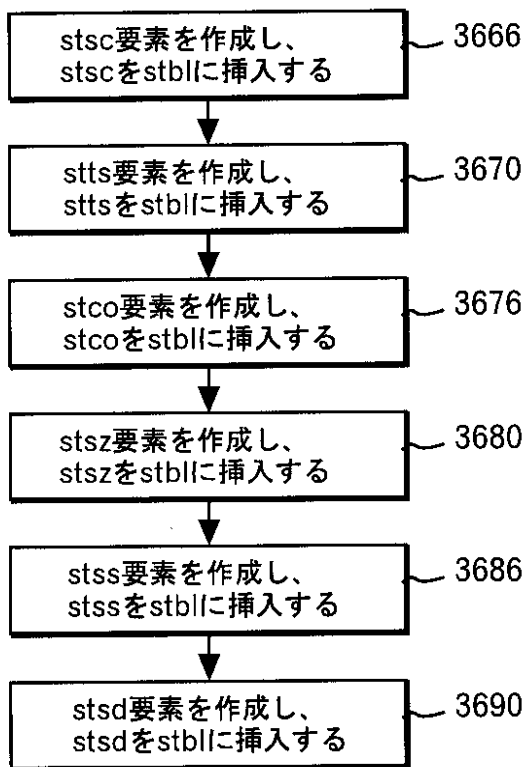
【図 8 3】



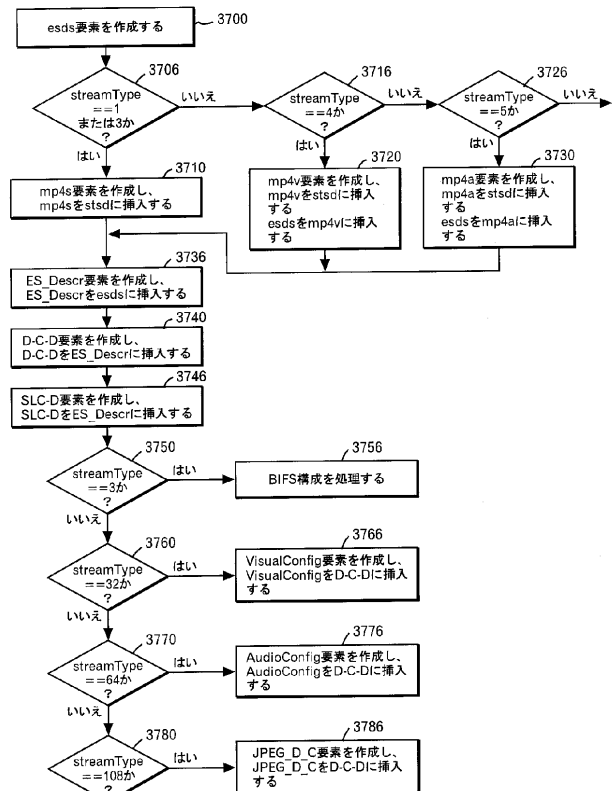
【図 8 4】



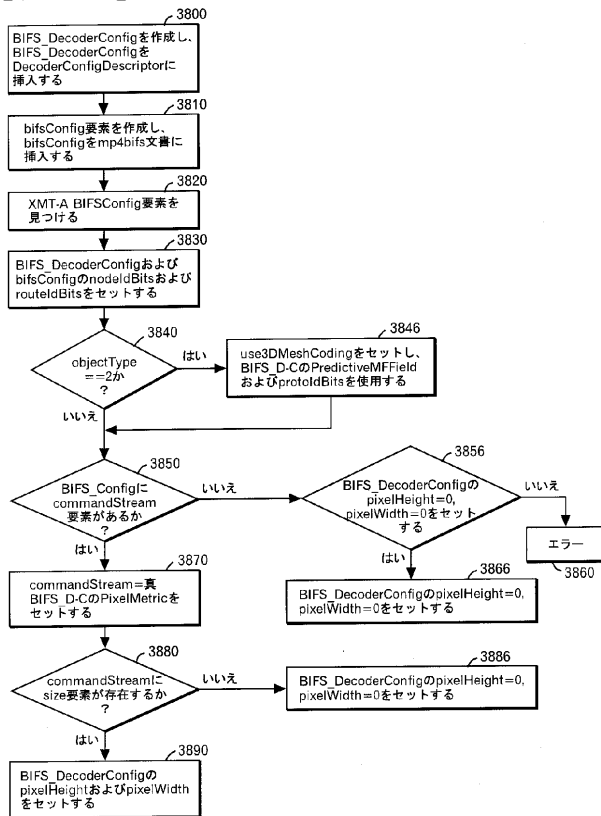
【図 8 5】



【図 8 6】



【図 87】



【図 88】

3900 position	3910 ObjectDescriptionID	3920 OddId	3930 startTime	3940 stopTime	3950 EsId
0	OdString0	1	parTime1	parTime2	esid1
1	OdString1	2	parTime3	parTime4	esid2
...	...	...	...	...	...

【図 89】

3960 position	3966 Node String
0	NodeString0
1	NodeString1
...	...

【図 90】

3970 position	3976 Route String
0	RouteString0
1	RouteString1
...	...

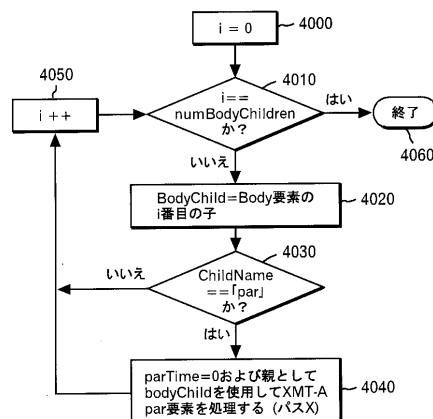
【図 92】

3990 entry	3992 OdId	3994 OdId	3996 OdId
0	OdId1	time0	真
1	OdId2	time1	真/偽
...	...	...	...

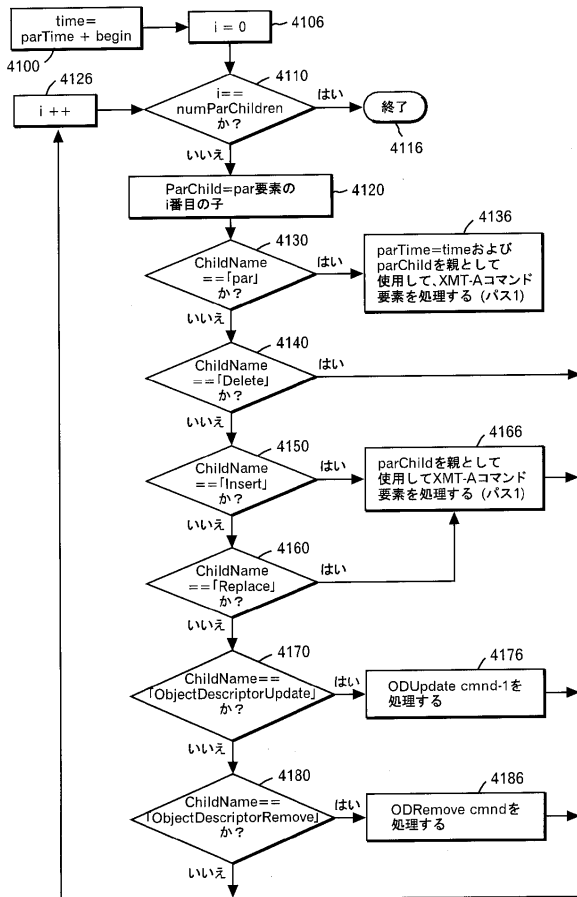
【図 91】

3980 position	3986 ReplaceSceneTime
0	time0
1	time1
...	...

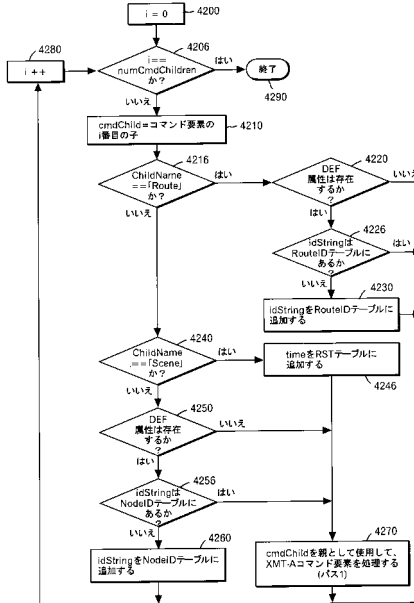
【図 93】



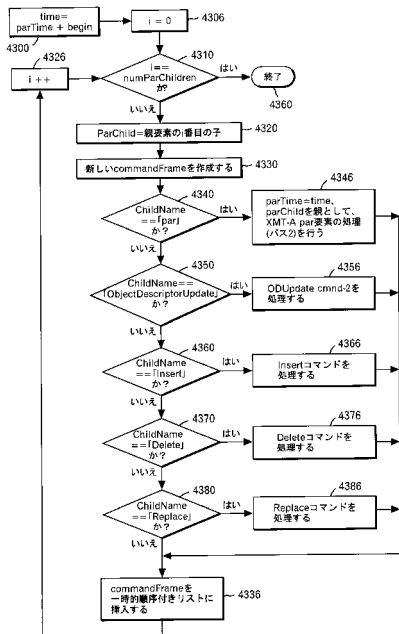
【図 9 4】



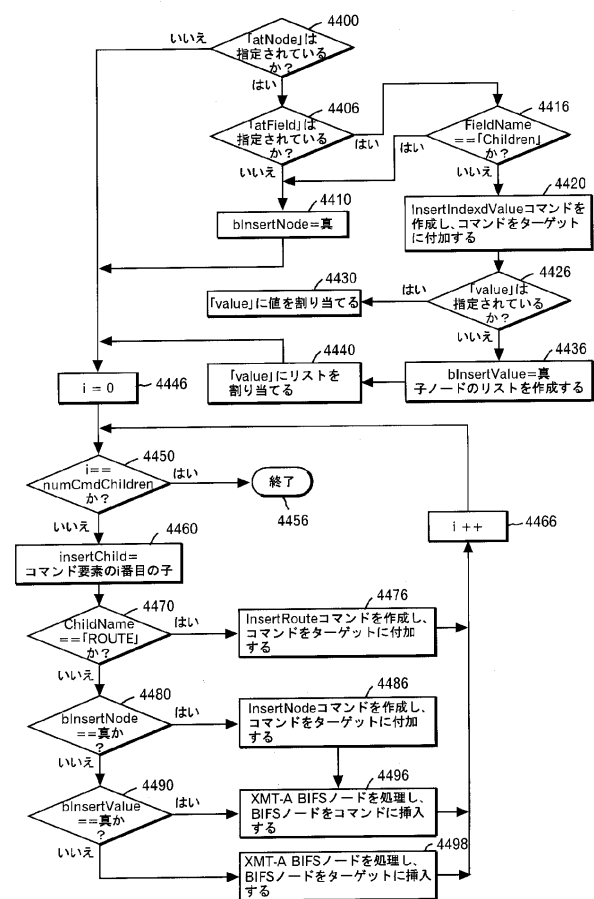
【図 9 5】



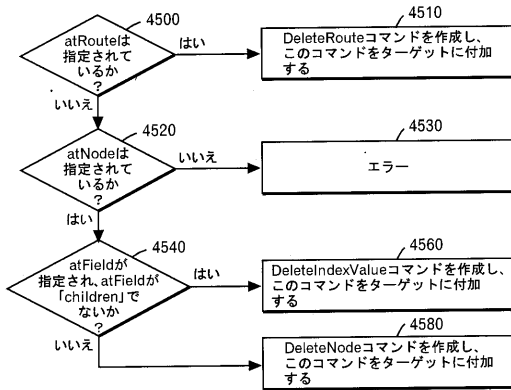
【図 9 6】



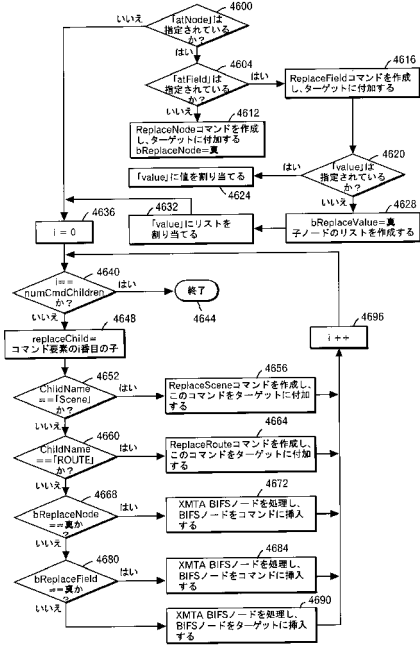
【図 9 7】



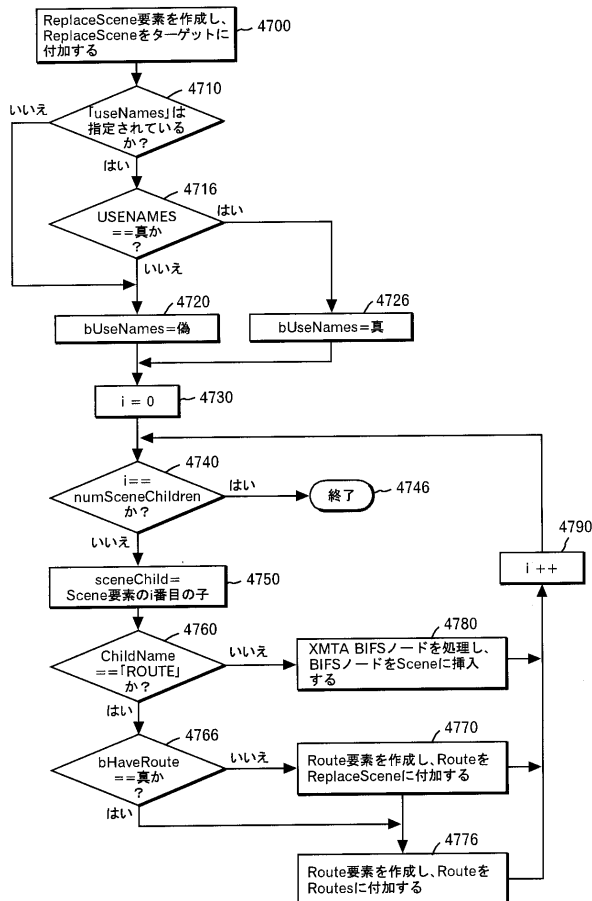
【図 98】



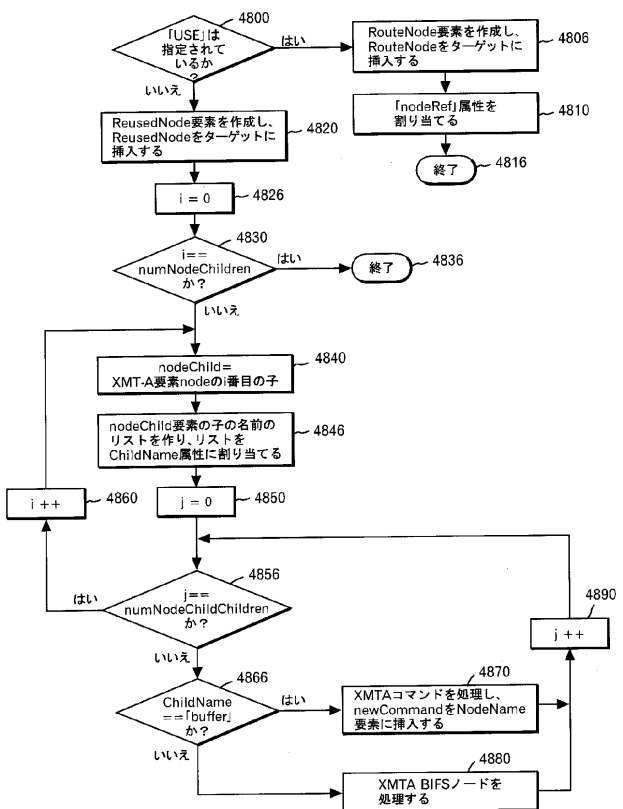
【図 99】



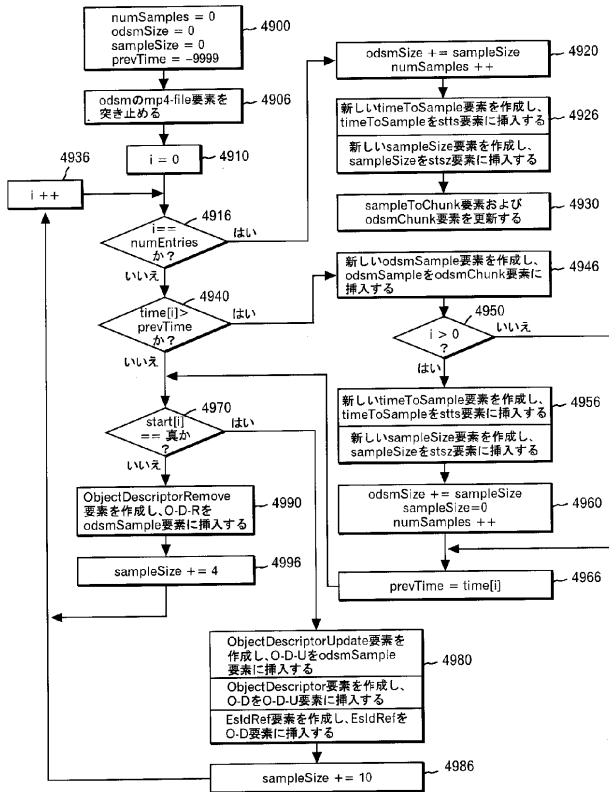
【図 100】



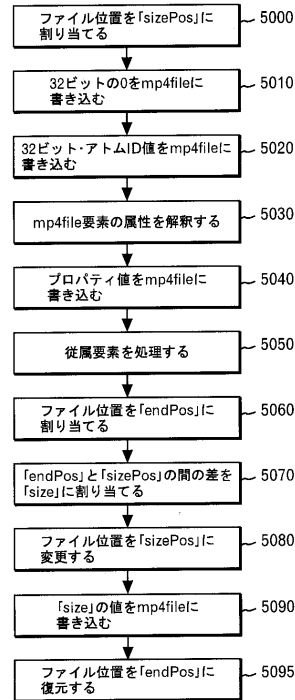
【図 101】



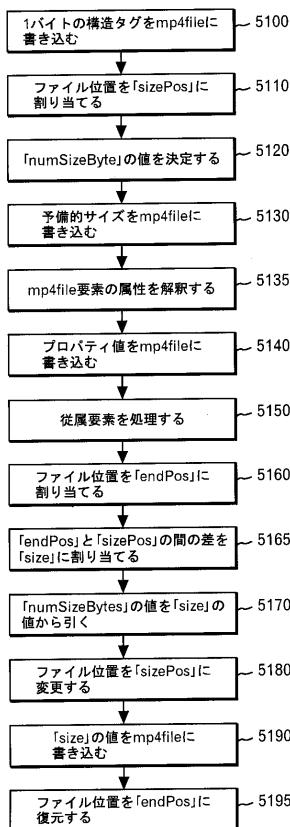
【図102】



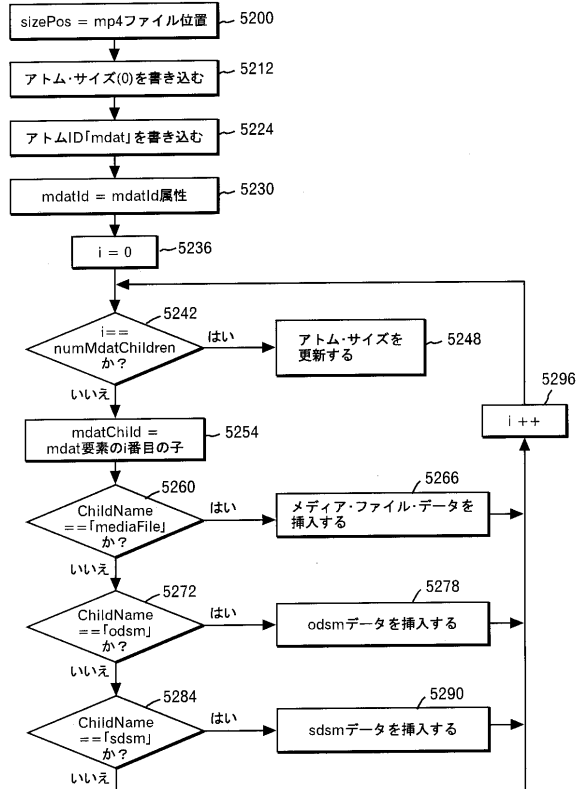
【図103】



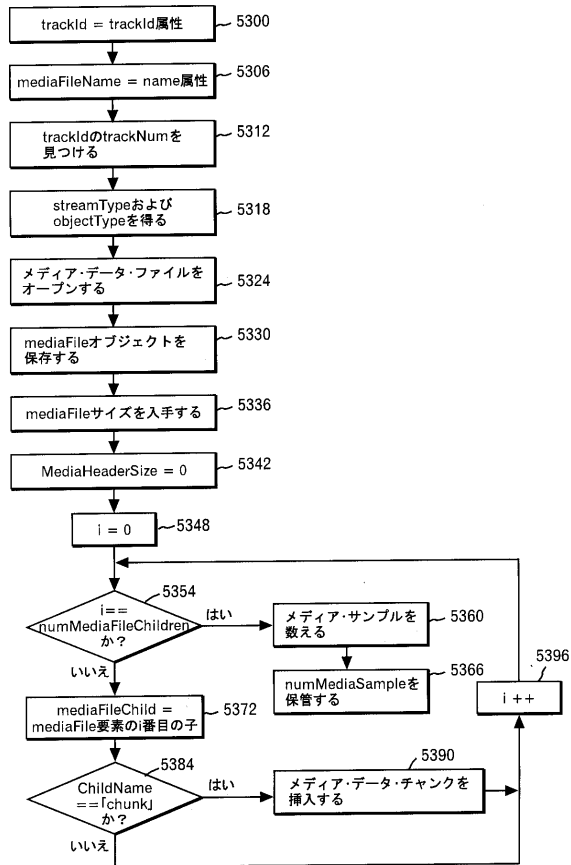
【図104】



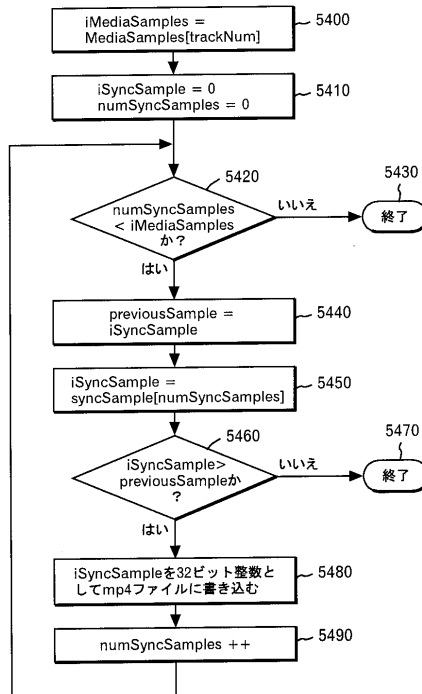
【図105】



【図 106】



【図 107】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US03/38137

A. CLASSIFICATION OF SUBJECT MATTER												
IPC(7) : G06F 17/21 US CL : 715/513 According to International Patent Classification (IPC) or to both national classification and IPC												
B. FIELDS SEARCHED												
Minimum documentation searched (classification system followed by classification symbols) U.S. : 715/513, 500.1, 501.1, 515, 530												
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Google NPL Search: MP4 and XMT												
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)												
C. DOCUMENTS CONSIDERED TO BE RELEVANT												
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.										
X	"Avipix products", [http://web.archive.org/web/20011219115055/http://avipix.com/products.html], posted on the Internet on 19 December, 2001, pages 1-3.	1-32										
A	KIM et al., "Extensible MPEG-4 Textual Format (XMT)", [http://www1.acm.org/sigs/sigmm/MM2000/ep/michelle/index.html], Association for Computing Machinery, 2000, pages 1-11.	1-32										
A,P	CONCOLATO et al., "MPEG-4 BIFS and XMT Tutorial", [http://gpac.sourceforge.net/tutorial/bifs_intro.htm], last modified and published on the Internet on 9 OCTOBER 2003, pages 1-2.	1-32										
A	"MPEG-4 Tools by ENST", [http://web.archive.org/web/20021205045740/http://www.comelec.enst.fr/~dufourd/mpeg-4/tools.html], published on the Internet on 2 DECEMBER 2001, pages 1-15.	1-32										
☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.												
* Special categories of cited documents: <table border="0"> <tr> <td>"A" document defining the general state of the art which is not considered to be of particular relevance</td> <td>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</td> </tr> <tr> <td>"E" earlier application or patent published on or after the international filing date</td> <td>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</td> </tr> <tr> <td>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</td> <td>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</td> </tr> <tr> <td>"O" document referring to an oral disclosure, use, exhibition or other means</td> <td>"&" document member of the same patent family</td> </tr> <tr> <td>"P" document published prior to the international filing date but later than the priority date claimed</td> <td></td> </tr> </table>			"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention	"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art	"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family	"P" document published prior to the international filing date but later than the priority date claimed	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention											
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone											
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art											
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family											
"P" document published prior to the international filing date but later than the priority date claimed												
Date of the actual completion of the international search 21 October 2004 (21.10.2004)		Date of mailing of the international search report 16 NOV 2004										
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450 Alexandria, Virginia 22313-1450 Facsimile No. (703) 305-3230		Authorized officer Joseph H Feild <i>Pageot Harrod</i> Telephone No. 703-305-3900										

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

PCT/US03/38137

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	"MPEG-4 Systems(XMT-A)-based Authoring Tool", [http://web.archive.org/web/20030628002759/http://www.etri.re.kr/e_etri/intro/newtech/etri_21c_32_2003.html], published on the Internet on 28 JUNE 2003, pages 1-2.	1-32
A,E	US 6,751,623 B1 (BASSO et al.) 15 June 2004 (15.06.2004), column 2 line 35 - column 3 line 60.	1-32

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW

(特許庁注：以下のものは登録商標)

U N I X

(72)発明者 ルケン、ウィリアム

アメリカ合衆国 1 0 5 9 8 ニューヨーク州ヨークタウン・ハイツ ファーム・ウォーク・ロード
2 9 2 4

(72)発明者 ロイ、エティーネ

フランス F-7 7 4 5 0 エスブリ リュ・ガンベッタ 2 8

Fターム(参考) 5B009 QA06 TA11

5B082 GA02