

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4824266号
(P4824266)

(45) 発行日 平成23年11月30日 (2011.11.30)

(24) 登録日 平成23年9月16日 (2011.9.16)

(51) Int. Cl.

F I

G O 6 F 17/21 (2006.01)

G O 6 F 12/00 (2006.01)

G O 6 F 13/00 (2006.01)

G O 6 F 17/21 5 7 0 L

G O 6 F 17/21 5 0 1 T

G O 6 F 17/21 5 9 6 Z

G O 6 F 12/00 5 4 5 M

G O 6 F 12/00 5 4 6 R

請求項の数 7 (全 20 頁) 最終頁に続く

(21) 出願番号 特願2002-563369 (P2002-563369)
 (86) (22) 出願日 平成14年1月30日 (2002.1.30)
 (65) 公表番号 特表2004-519038 (P2004-519038A)
 (43) 公表日 平成16年6月24日 (2004.6.24)
 (86) 国際出願番号 PCT/IB2002/000336
 (87) 国際公開番号 W02002/063494
 (87) 国際公開日 平成14年8月15日 (2002.8.15)
 審査請求日 平成17年1月28日 (2005.1.28)
 審判番号 不服2008-25640 (P2008-25640/J1)
 審判請求日 平成20年10月6日 (2008.10.6)
 (31) 優先権主張番号 01/01530
 (32) 優先日 平成13年2月5日 (2001.2.5)
 (33) 優先権主張国 フランス (FR)
 (31) 優先権主張番号 01402815.3
 (32) 優先日 平成13年10月30日 (2001.10.30)
 (33) 優先権主張国 欧州特許庁 (EP)

(73) 特許権者 590000248
 コーニンクレッカ フィリップス エレク
 トロニクス エヌ ヴィ
 オランダ国 5 6 2 1 ベーアー アイン
 ドーフェン フルーネヴァウツウェッハ
 1
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (72) 発明者 アミエル・カプリオリオ, ミリアン
 オランダ国, 5 6 5 6 アーアー アイン
 ドーフェン, プロフ・ホルストラーン 6
 (72) 発明者 シャンピオン, マティウ
 オランダ国, 5 6 5 6 アーアー アイン
 ドーフェン, プロフ・ホルストラーン 6

最終頁に続く

(54) 【発明の名称】 フォーマットが適合したオブジェクトの転送方法

(57) 【特許請求の範囲】

【請求項 1】

サーバからクライアントへオブジェクトの適合バージョンを含むファイルを転送する方法であって、
 該サーバは、

マークアップ言語で書かれ、前記オブジェクトの基本バージョンを有する基本プログレッシブ・ファイルが記述される基本文書と、一つ以上の変換された文書を取得するため前記基本文書に適用される1以上の所定の変換規則とを保持するメモリと、

プロセッサと、
 を有し、

前記方法は、前記プロセッサによって実行される、

前記クライアントのプロファイルの関数として前記1以上の所定の変換規則から所定の変換規則を選択する手順と、

変換された文書を作成するため前記選択された変換規則を前記基本文書に適用する変換手順と、

該変換された文書に基づいて該オブジェクトの該適合バージョンを格納する転送ファイルを作成する手順と、

前記転送ファイルを前記サーバから前記クライアントに転送する手順と、
 を有し、

前記基本文書には、前記基本プログレッシブ・ファイルの構造が記述され、

前記変換手順は、前記基本プログレッシブ・ファイルを復号化することなく別のフォーマットのプログレッシブ・ファイルに変換することを特徴とするファイルを転送する方法。

【請求項 2】

前記基本文書がタグによって区切られ、一つ以上の属性を有することが可能な一つ以上の要素を格納するとき、前記変換手順は、一つ以上の要素を抑制し、一つ以上の属性及び／又は一つ以上の要素の値を変更することを有する、ことを特徴とする請求項 1 記載の方法。

【請求項 3】

前記所定の変換規則は、前記マークアップ言語で記述された文書を該マークアップ言語で記述された別の文書に変換するための規則を定義することができる変換言語で書かれていることを特徴とする、請求項 1 記載の方法。

【請求項 4】

請求項 1 記載の方法の各手順を実行するための命令を有するプログラム。

【請求項 5】

前記サーバと前記クライアントが相互に離れているとき、該クライアントは、オブジェクト転送に係る要求を該サーバへ送信するものであり、

前記サーバは、

サーバ・モジュールと、

処理モジュールと、

変換モジュールと、

を有し、

前記サーバ・モジュールは、前記要求を受信する機能と前記要求に対する応答を送信する機能を有し、

前記処理モジュールは、前記要求に対応した基本文書を検出する機能と、該検出した基本文書に係る要求を前記サーバ・モジュールに送信する機能と、前記クライアントへ転送される前記オブジェクトの適合バージョンを格納するファイルを作成する機能と、前記作成されたファイルを、前記クライアントへ送信するため、前記サーバ・モジュールへ送信する機能を有し、

前記変換モジュールは、前記基本文書に係る要求を前記サーバ・モジュールから受信する機能と、前記変換規則を選択し、前記選択した変換規則を前記基本文書に適用する機能と、前記適用する機能によって変換された文書を前記処理モジュールへ送信するため、前記変換された文書を前記サーバ・モジュールへ送信する機能と、を有する、ことを特徴とする請求項 1 記載のファイルを転送する方法。

【請求項 6】

前記要求は、前記サーバが前記クライアントのプロファイルを決定することを可能にさせ、

前記変換モジュールによって適用される変換規則は、前記プロファイルの関数である、ことを特徴とする請求項 5 記載のファイルを転送する方法。

【請求項 7】

前記プロファイルは、前記サーバと前記クライアントの間の交渉段階を経て決定されることを特徴とする請求項 6 記載のファイルを転送する方法。

【発明の詳細な説明】

【0001】

本発明は、オブジェクトの適合バージョンを含むファイルを、送信元エンティティから宛先エンティティへ転送する方法に関する。本発明は、オブジェクトの適合バージョンを含むファイルを作成する方法にも関する。

【0002】

本発明は、更に、このようなファイル作成方法を実施するために適した機器と、このようなファイル転送方法を実施するために適した伝送システムと、に関する。

10

20

30

40

50

【 0 0 0 3 】

更に、本発明は、
マークアップ言語で書かれた文書と、
プログレッシブ・ファイルを格納する少なくとも一つのメイン・ボックス、及び、マークアップ言語で書かれた文書を格納する少なくとも一つのサイド・ボックスを含むボックスにフォーマットされたファイルと、
このような文書、又は、このようなファイルが保存される記憶媒体と、
に関する。

【 0 0 0 4 】

本発明は、インターネットの分野で特に興味深いアプリケーションが見出される。インターネットにアクセスできる機器の台数は増加し続けている。例えば、パーソナルコンピュータ、携帯電話機、携帯情報端末（PDA）、ポータブルコンピュータ、ポケットコンピュータなどは、そのような機器の例である。これらの機器は、リソース（画面、計算能力、記憶容量など）に多少の制限があり、速度が変化するリンク（無線リンク、有線リンク、光リンクなど）を用いてインターネットに接続される。転送されるオブジェクトは、受信機のプロファイルに応じて適合させられるべきである。これにより、特に、受信機が使用できないようなデータを無駄に送信することが回避され、通過帯域を節約することが可能である。

【 0 0 0 5 】

1999年8月19日に公開された国際特許出願公開番号W099/41734号には、プログレッシブ・セクションに編成されたグラフィック・ファイルと、このようなファイルを選択的に転送する方法とが記載されている。この方法では、受信側は、プログレッシブ・ファイル内で必要なセクションを指定する。

【 0 0 0 6 】

この方法では、受信側がグラフィック・ファイルの構造を知っていること、及び、受信側が獲得したい解像度若しくは品質に応じて、再生されるべきセクションを決定する方法を知っていること、が前提とされている。

【 0 0 0 7 】

本発明の目的は、特に、ユーザからの特定の情報を必要とはせずに、適合バージョンのオブジェクトを含むファイルを転送する方法を提案することである。

【 0 0 0 8 】

この目的を達成するため、冒頭の段落で説明した転送方法は、送信元エンティティがマークアップ言語で書かれた基本文書を保持するメモリを具備し、基本文書は基本バージョンのオブジェクトを格納する基本プログレッシブ・ファイルを記述し、当該方法は、上記適合バージョンに応じて所定の変換を選択する手順と、
選択された変換を上記基本文書に適用し、変換された文書を作成する変換手順と、
上記変換された文書に基づいて上記オブジェクトの上記適合バージョンを格納するファイルを作成する手順と、
作成されたファイルを転送する手順と、
を有することを特徴とする。

【 0 0 0 9 】

このようにバージョンの適合は、送信元エンティティ側で行われる。従って、宛先エンティティ側で特別の知識が要求されることは無い。

【 0 0 1 0 】

本発明によれば、送信元エンティティは、マークアップ言語で書かれた基本文書を使用し、この基本文書は転送されるべき基本バージョンのオブジェクトを格納する基本プログレッシブ・ファイルを記述する。送信元エンティティは、一つ以上の変換された文書を獲得するため、この基本文書を適用対象とする一つ上の所定の変換を備えている。転送されるファイルは、獲得された変換後の文書から作成される。転送されるファイルは、転送される適合バージョンのオブジェクトを格納する。この適合バージョンは、基本バージョンと

10

20

30

40

50

は異なる。

【 0 0 1 1 】

本発明は、特に、オブジェクトをマークアップ言語で記述し、記述された内容に変換を適用する。以下では、この記述された内容を文書と呼ぶ。このようなマークアップ言語を用いることによって、基本プログレッシブ・ファイルの構造を記述できるようになる。即ち、基本プログレッシブ・ファイルの構造は、得られた文書の中に現れる。従って、これらの文書に当てはまる変換を定義することが可能である。これらの変換は簡単な様式で表現され、これらの変換を使用するために必要な計算は少ない。

【 0 0 1 2 】

基本プログレッシブ・ファイルの構造が現れている文書に関して変換を実行することにより、基本プログレッシブ・ファイルを別のフォーマットに再符号化するため、基本プログレッシブ・ファイルを復号化する必要が無くなる。これにより、計算負荷がかなりの程度で制限される。

10

【 0 0 1 3 】

更に、本発明によれば、基本文書を送信元エンティティに保存するだけで十分であり、一部の実施例では、基本文書に由来する基本プログレッシブ・ファイル、及び、所定の変換も送信元エンティティに保存すれば足りる。転送されるファイルは、必要に応じて、適合した変換を適用することによって基本文書から作成される。したがって、転送されるファイルを送信元エンティティのメモリに保存する必要はない。かくして、送信元エンティティのレベルで保存されるべき情報量は非常に削減される。

20

【 0 0 1 4 】

本発明の好ましい一実施例では、基本文書は、タグによって区切られ、一つ以上の属性が与えられ得る一つ以上の要素を格納し、上記変換は、一つ以上の要素を抑制し、一つ以上の属性と一つ以上の要素のうちの少なくとも一方の値を変更する。

【 0 0 1 5 】

所定の変換は、有利的には、マークアップ言語で記述された文書を当該マークアップ言語で記述された別の文書に変換するための規則を定義することができる変換言語で書かれている。

【 0 0 1 6 】

一例として、マークアップ言語はXML言語（拡張マークアップ言語）であり、変換言語はXSL言語（拡張スタイル・シート言語）であり、XML言語とXSL言語の両方はW3Cコンソーシアムによって規定されている。

30

【 0 0 1 7 】

本発明のもう一つの目的は、所与の文字の組を用いて、マークアップ言語で書かれた文書を提案することであり、この文書は、バイナリ・データ、マーカー、及び、一つ以上のマーカーに関連付けられたパラメータを格納するバイナリ・ファイルを記述する。本発明によれば、このような文書は、

- ・コンテンツと、一つ以上の属性と、一つ以上のサブ要素とのうちの少なくとも一つを格納する要素を含み、

- ・各マーカーに関連付けられた要素、上記要素のサブ要素若しくは属性を構成する上記マーカーに関連付けられる可能性のあるパラメータ、並びに、上記文字の形式である一つ以上の要素のコンテンツは、上記バイナリ・データと結び付けられている。

40

【 0 0 1 8 】

第1の実施例では、上記文書のコンテンツの少なくとも一部は、上記文字の組に変換された上記バイナリ・ファイルのバイナリ・データに対応する。

【 0 0 1 9 】

この第1の実施例では、文書は、バイナリ・ファイルとは独立している。なぜならば、文書自体がバイナリ・ファイルのデータを格納しているからである。しかし、バイナリ・データを文字に変換することは、これらのデータのある種の拡張を意味する。

【 0 0 2 0 】

50

第2の実施例では、上記文書のコンテンツの少なくとも一部は、上記バイナリ・ファイル内での上記バイナリ・データへの一つ以上のポイントに対応する。

【0021】

この第2の実施例では、文書は、データにアクセスするためにバイナリ・ファイルに戻る。文書が記述するバイナリ・ファイルは文書に添付されていると役に立つ。本実施例では、データの拡張は無い。

【0022】

本発明のもう一つの目的は、本発明によるファイル転送方法を実現するため有利に用いられる特定のバイナリ・ファイルを提案することである。

【0023】

このようなバイナリ・ファイルは、ボックスの形式にフォーマットされ、バイナリ・ファイルは、少なくとも、
プログレッシブ・ファイルを含むメイン・ボックスと、
マークアップ言語で書かれ、上記プログレッシブ・ファイルを記述する文書を格納する少なくとも一つの補助的なボックスと、
を含む。補助的なボックスは、有利的には、上記文書を適用対象とする一つ以上の変換を格納する。

【0024】

かくして、オブジェクトに関するすべての情報は、単一のファイルに再編され、これは、ファイル管理の観点から見ると有利である。

【0025】

このようなファイルは、例えば、JPEG2000ファイルである。本例の場合、補助的なボックスは、例えば、JPEG2000標準に規定されたオプションのXMLボックスによって構成される。補助的なボックスは、新しい専用のボックスによって構成してもよい。

【0026】

送信元エンティティと宛先エンティティが互いに離れている場合、宛先エンティティは、有利的には、オブジェクト転送に関する要求を送信元エンティティへ送信しようとする。宛先エンティティは、
要求と要求に対する応答を送受信するサーバ・モジュールと、
オブジェクト転送に関係した要求を受信し、上記要求に対応した基本文書を見つけ出し、上記基本文書に関係した要求を上記サーバ・モジュールに送信し、転送されるオブジェクトの適合バージョンを格納するファイルを作成し、作成されたファイルを、宛先エンティティへ送信するため、サーバ・モジュールへ送信する処理モジュールと、
上記基本文書に関係する上記要求を上記サーバ・モジュールから受信し、変換を選択し、変換された文書を上記処理モジュールへ送信するため、変換された文書を上記サーバ・モジュールへ送信する変換モジュールと、
を含む。

【0027】

更に、宛先エンティティによって送信された要求は、送信元エンティティが宛先エンティティのプロファイルを決定できるようにさせ、変換モジュールによって適用された変換は、上記プロファイルの関数である。例えば、上記要求は、送信元エンティティが宛先エンティティのプロファイルを決定できるようにするため、送信元エンティティと宛先エンティティの間の交渉フェーズを含む。

【0028】

本発明は、送信元エンティティから宛先エンティティへある特定のバージョンのオブジェクトを転送する方法に関する。本発明による転送方法は、この転送のため使用されるバージョン（のオブジェクト）を作成するステップを含む。このバージョンの作成は、基本文書に基づいて行われる。基本文書は、マークアップ言語で書かれ、上記オブジェクトの基本バージョンを格納する基本プログレッシブ・ファイルを記述する。このステップは、変

10

20

30

40

50

換された文書を作成することができる。最終的に、変換された文書は、適合バージョンと呼ばれるオブジェクトの別のバージョンを格納するファイルを作成するため使用される。このファイルが宛先エンティティへ転送されるファイルである。

【 0 0 2 9 】

図 1 は、本発明によるファイル転送方法のステップを概略的に説明するブロック図である。ブロック B 1 は、オブジェクト O のある特定のバージョンに対応する基本プログレッシブ・ファイル F (O) を表す。ブロック B 2 は、マークアップ言語で書かれ、基本プログレッシブ・ファイルを記述する基本文書 D O C (O) を表す。ブロック B 3 - 1 乃至 B 3 - N は、変換された文書を作成するため、基本文書 D O C (O) に適用されることになっている変換 T R i (O) を表す。選択ステップ S は、オブジェクト O に関するバージョンと、選択されたバージョンの関数である適用される変換と、を選択できるようにする。ブロック B 4 i は、変換 T R i (O) を基本文書 D O C (O) に適用することにより作成された変換後の文書 D O C i (O) を表現する。最後に、ブロック B 5 i は、変換後の文書 D O C i (O) から作成されたファイル F i (O) を表現し、ファイル F i (O) はオブジェクト O の別のバージョンを格納する。最終的に、ステップ T において、ファイル F i (O) は宛先エンティティへ送信される。

10

【 0 0 3 0 】

このファイル転送方法は、基本ファイルのプログレッシブ性による利益を得る。このファイル転送方法は、オブジェクトの適合バージョンを作成するため操作できるようにするため、プログレッシブ基本ファイルのシンタックス (構文) を文書に記述する。

20

【 0 0 3 1 】

このような方法は、有利的には、送信元エンティティと宛先エンティティが互いに離れている本発明による伝送システムに使用される。このような伝送システムの一例は、図 2 に示されている。このシステムは、送信元エンティティと、少なくとも一つの宛先エンティティとを含み、送信元エンティティと宛先エンティティの各々は、インターネットにアクセスすることができ、これらのアクセスは同図に参照番号 4 及び 5 で示されている。

【 0 0 3 2 】

實際上、送信元エンティティは、1 台若しくは多数の機械に拡散される。多数の機械に拡散されることにより、タスクを分散させ、送信元エンティティのレベルでの過負荷を防止することが可能になる。以下では、説明を簡略化するため、送信元エンティティは、サーバと呼ばれる 1 台の機械に収容されている場合を考える。

30

【 0 0 3 3 】

サーバ 1 は、メモリ 1 0 及びプロセッサ組立体 1 2 を含む。メモリ 1 0 は、例えば、リード・オンリー・メモリ R O M、ランダム・アクセス・メモリ R A M、ハードディスク H D の形式であり、場合によっては、フレキシブルディスクタイプの着脱式記憶媒体 F D、C D - R O M、D V D などの形式であり、これらの媒体を読み取る機能を備えたリーダーに挿入される。メモリ 1 0 は、特に、基本プログレッシブ・ファイル F (O) と、基本プログレッシブ・ファイルを記述する基本文書 D O C (O) と、基本プログレッシブ・ファイルを適用対象とする少なくとも一つの変換 T R i (O) と、本発明によるファイル転送方法を実現するためプロセッサ 1 2 による実行対象であるプログラム P 1 0 の組と、を含む。

40

【 0 0 3 4 】

クライアント機器 2 は、メモリ 2 0 と、プロセッサ組立体 2 2 と、画面 2 3 と、例えば、キーパッド、音声コントロール、マウスなどを備えたユーザ・インタフェース 2 4 と、を含む。メモリ 2 2 は、例えば、リード・オンリー・メモリ R O M と、ランダム・アクセス・メモリ R A M とにより構成される。ある種のケースでは、メモリは、ハードディスク H D の形式や、着脱式記憶媒体 F D、C D - R O M、D V D などの形式でもよく、着脱式記憶媒体は媒体に記憶されたデータを読み取る機能を備えたリーダーに挿入される。メモリ

50

２２は、特に、本発明による転送方法を実現するためプロセッサ２２によって実行されるプログラムＰ２０の既知の組を格納する。

【００３５】

クライアント機器２は、通常の通り、インターネットに接続され、リソース識別子を含むＨＴＴＰタイプの要求を送出する（ＨＴＴＰは、ＩＥＴＦのＲＦＣ２６１６に規定されたネットワーク・プロトコルであり、リソース識別子のフォーマットは、ＩＥＴＦのＲＦＣ２３９６に規定されている。）。このリソースを収容する機器は、当該クライアント機器に対して、ＨＴＭＬ文書を転送することによって応答する。このＨＴＭＬ文書は、文書のコンテンツを機器の画面に表示させるため、クライアント機器によって解釈される（ＨＴＭＬは、コンテンツを記述するための言語であり、インターネット上で普及している言語であり、Ｗ３Ｃコンソーシアムによって規定されている。）。よく知られているように、ＨＴＭＬ文書は画面に表示する対象のページを記述し、このようなページは、ユーザがある特定のオブジェクトを格納しているファイルをダウンロードするため選択するリンク（例えば、ボタン若しくはアイコンの形式）を格納している。ユーザがこのようなリンクを選択したとき、クライアント機器は、ファイルの転送を依頼するためＨＴＴＰタイプの要求を送信する。

10

【００３６】

図３には、送信元エンティティ１のソフトウェア構造の一例が示されている。図３によれば、送信元エンティティ１は、サーバ・モジュール３０と、処理モジュール３２と、変換モジュール３４の三つのソフトウェア・モジュールを含む。サーバ・モジュール３０は、例えば、ＡＰＡＣＨＥ（ＡＰＡＣＨＥは、Apacheソフトウェア財団によって開発されたｈｔｔｐサーバである。）という名称で知られているモジュールの形式である。サーバ・モジュールは、ある種の要求を他のモジュールへ送信するように構成される。サーバ・モジュール３０は、プログレッシブ・フォーマット（例えば、ＪＰＥＧ２００ファイル）を有するファイルに関係した要求を処理モジュール３２へ送信し、マークアップ言語で書かれたファイル（例えば、ＸＭＬファイル）に関係する要求を変換モジュール３４へ送信する。変換モジュール３４は、例えば、ＣＯＣＯＯＮという名称で知られているモジュールにより形成される（ＣＯＣＯＯＮは、ＡＰＡＣＨＥ ＸＭＬプロジェクトにおいてＪａｖａ（Ｒ）言語で開発されたアプリケーションである）。

20

【００３７】

処理モジュール３２は、特に、宛先エンティティによる送信要求に示されたプログレッシブ・ファイル・フォーマットに対応した基本文書ＤＯＣ（Ｏ）を見つけ出す機能と、この基本文書ＤＯＣ（Ｏ）に関係する要求をサーバ・モジュール３０へ送信する機能と、を備えている。サーバ・モジュール３０は、この要求を受信すると、変換モジュール３４によって処理されるべきマークアップ・フォーマット（例えば、ＸＭＬ）を認識し、その要求を変換モジュール３４へ送信する。

30

【００３８】

変換モジュール３４は、変換された文書ＤＯＣｉ（Ｏ）を生成するため、基本文書ＤＯＣ（Ｏ）を復元し、基本文書ＤＯＣ（Ｏ）に適用される変換ＴＲｉ（Ｏ）を選択する機能を備えている。変換モジュール３４は、変換された文書ＤＯＣｉ（Ｏ）をサーバ・モジュール３０へ返し、サーバ・モジュール３０は、変換された文書ＤＯＣｉ（Ｏ）を処理モジュール３２へ送信する。

40

【００３９】

処理モジュール３２は、転送されるオブジェクトの適合バージョンを格納するバイナリ・ファイルＦｉ（Ｏ）を作成し、サーバ・モジュール３０がそのオブジェクトを宛先エンティティ２へ送信できるようにそのオブジェクトをサーバ・モジュール３０へ返す機能を備えている。

【００４０】

図３には、送信元エンティティ１から宛先エンティティ２へ適合バージョンを転送するために必要な種々のステップが、番号付きの矢印で示されている。

50

【 0 0 4 1 】

矢印 F 1 は、宛先エンティティ 2 によるオブジェクトの転送に関する要求の送信を表現する。

【 0 0 4 2 】

ステップ F 2 では、サーバ・モジュール 3 0 は、宛先エンティティから送信された要求を受信し、プログレッシブ・フォーマットを認識し、その要求を処理モジュール 3 2 へ送信する。

【 0 0 4 3 】

ステップ F 3 では、処理モジュールは、プログレッシブ・フォーマットに対応した基本文書を見つけ出し、当該文書に関する要求 `http GET` を出す。基本文書を復元するため、処理モジュールは、暗黙的な規則を適用するか（例えば、転送されるオブジェクトを含むファイルの名前が `image.jp2` であるとき、関連した `XML` 文書の名前は `image.xml` であると決定する）、又は、明示的な規則を適用する（例えば、各基本プログレッシブ・ファイルに関連付けられた基本文書の名前を格納した対応関係のテーブルが利用できる。）。

10

【 0 0 4 4 】

ステップ F 4 では、サーバ・モジュール 3 0 は、基本文書に関する要求を変換モジュール 3 4 へ送信する。

【 0 0 4 5 】

ステップ F 5 では、変換モジュール 3 4 は、基本文書を復元し、変換を選択し、変換された文書を作成するため選択された変換を適用し、変換後の文書をサーバ・モジュール 3 0 へ返す。

20

【 0 0 4 6 】

ステップ F 6 では、サーバ・モジュール 3 0 は、変換モジュール 3 4 によって返された変換後の文書を処理モジュール 3 2 へ送信する。

【 0 0 4 7 】

ステップ F 7 では、処理モジュール 3 2 は、変換後の文書に基づいて、転送されるオブジェクトの適合バージョンを格納したバイナリ・ファイルを作成し、作成されたバイナリ・ファイルをサーバ・モジュール 3 0 へ返す。

【 0 0 4 8 】

最後に、ステップ F 8 において、サーバ・モジュール 3 0 は、オブジェクトの適合バージョンを格納したファイルを宛先エンティティ 2 へ送信する。

30

【 0 0 4 9 】

3 種類のモジュールが別々の機械に収容されている場合、有利的には、三つのモジュールのすべては、本発明による転送方法を実現するために必要なデータ、即ち、より具体的には、

基本プログレッシブ・ファイル `F (0)` と、

これらの基本プログレッシブ・ファイルを記述する基本文書 `DOC (0)` と、

基本プログレッシブ・ファイルを適用対象とする変換 `TR i (0)` と、

が記憶されている同じディスクにアクセスすることができる。

40

【 0 0 5 0 】

処理モジュール 3 2 及び変換モジュール 3 4 は、例えば、「サーブレット」、即ち、サーバ上で実行されることが予定されている `Java (R)` 言語で開発されたアプリケーションである（例えば、`c o c o o n` はサーブレットである。）。

【 0 0 5 1 】

次に、図 3 を参照して説明したサーバ・クライアント型アーキテクチャが使用されている本発明によるファイル転送方法の様々なアプリケーションを説明する。

【 0 0 5 2 】

図 4 に示されたアプリケーションの第 1 例によると、クライアント 2 によって送信された要求 J 1 は、サーバ 1 がクライアント 2 のプロファイル `PP` を決定できるようにさせる。

50

第1の実施例(図示せず)では、要求J1は、クライアント2のプロファイルPPを直接的に指定する。図4に示された第2の実施例では、サーバ1が要求J1を受信したとき、サーバ1は、交渉フェーズJ2を開始することによって応答する。サーバ1(より厳密には、サーバ1の変換モジュール34)は、クライアントのプロファイルPPに応じて、適切なファイルを作成するため、基本文書DOC(O)に適用される変換を選択する。一例として、このような交渉フェーズは、1998年にIETFによって発行された、"Transparent Content Negotiation in HTTP"という名称のRFC2295に記載されているタイプの交渉フェーズ、又は、1999年10月10日に無線アプリケーション・プロトコル・フォーラム(Wireless Application Protocol Forum)によって発行された文献"Wireless Application Group, User Agent Profile Specification"に記載されているタイプの交渉フェーズである。

10

【0053】

アプリケーションの第2例は、ウェブ・ページが低解像度のオブジェクト(屢々、画像)を含み、クライアントが同じオブジェクトの完全な解像度のものをダウンロードするため、そのオブジェクトをクリックする場合に関係する。この場合、サーバ1の変換モジュール34は、クライアントが既に使用したデータを補完するデータだけを格納するファイルを作成するため、特定の変換を適用する。

【0054】

アプリケーションの第3例は、同じウェブサイト上で、画像が、異なるフォーマットで、或いは、異なる解像度で、数回に亘って(例えば、異なるページに)提供される場合に関係する。クライアントがページ間で行き来するとき、サーバは、同じ画像の様々なバージョンを次から次へと送信させられる。変換モジュール34は、望ましいバージョンを作成するために適切な変換を毎回適用するので、送信された様々な画像バージョンをサーバに保存する必要はない。

20

【0055】

本発明は、図3に示されたタイプのクライアント・サーバ型アーキテクチャを使用する形態に限定されない。本発明は、ユーザによってローカルに使用されるオブジェクトの適合バージョンを格納するファイルを作成する方法にも関係する。例えば、このようなファイル作成方法は、有利的には、大きい画像に基づいて小さい画像を作成しようとするユーザによって使用される。

30

【0056】

本発明は、例えば、マルチメディア・オブジェクト、即ち、固定画像、音声又は映像に適用される。

【0057】

このタイプのオブジェクトのあるバージョンを格納しているファイルは、例えば、JPEG2000フォーマット、GIFフォーマット、又は、MPEG4フォーマットのファイルである。以下の説明中、これらのファイルは、バイナリ・ファイルとしてみなされる。なぜならば、物理的に、これらのファイルは、任意のビットシーケンスとして現れるからである。

【0058】

プログレッシブ・ファイルは、ファイルの一部だけを復元することによってオブジェクトの低品質バージョンを獲得できるように編成される。プログレッシブ性には様々な種類が存在する。より多くのデータを復元することにより、例えば、信号の品質を高め、より大きいサイズの画像を取得し、階調画像ではなくカラー画像を取得し、映像のフレーム周波数を増加させることなどが可能になる。例えば、JPEG2000ファイルは、プログレッシブ・ファイルである。

40

【0059】

画像ファイルのある種のフォーマットは、画像ファイルに格納されている画像の低品質バージョン(屢々、小サイズのバージョン)である「サムネイル」を画像ファイルに取り込むことが可能である。例えば、Exif2.1ファイル("Digital Still Camera Image File For

50

mat Standard; Exchangeable image file format for Digital Still Cameras: Exif, Version 2.1", 1998年6月12日、日本電子工業振興協会(JEIDA)) は、その一例である。このタイプのファイルは、本発明におけるプログレッシブ・ファイルである。このタイプのファイルに関連した基本文書は、一方で、サムネイルを記述し、他方で、画像を記述する。例えば、基本文書からサムネイルに関係した一部を抽出できる変換が定義されるので、サムネイルを記述する変換後の文書が作成され、変換後の文書は当該サムネイルを格納したバイナリ・ファイルを作成するために使用されることになる。

【 0 0 6 0 】

一般的に、マークアップ言語は、データセットの構文を記述するために、タグを利用し、これらのタグの用法を規定する言語を意味していると考えられる。このような言語は、データセットを構造化し、データセットの構造をデータセットの内容から分離することが可能である。

10

【 0 0 6 1 】

X M L は、このようなマークアップ言語の典型的な例である。X M L の長所は、インターネットの世界で広範に亘って使用されていることである。X M L の別の長所は、ある程度の数のツール、特に、X M L 文書に適用可能な変換を定義することができる X S L T という名称のツールを提供することである。変換の定義は、X S L T スタイル・シートを用いて行われる。X S L T 変換の場合、X S L T プロセッサは、X M L 文書と X S L T スタイル・シートを読み、別の文書、例えば、X M L 文書を作成する。X M L 及び X S L T は、このように、本発明を実施するために特に適している。

20

【 0 0 6 2 】

以下の説明では、具体的な例を挙げるため、J P E G 2 0 0 0 フォーマットのファイルを記述する文書を作成するために X M L 言語が使用される。しかし、このような例に限定されるものではない。別のタイプのマークアップ言語と他のプログレッシブ・ファイル・フォーマットを使用してもよい。

【 0 0 6 3 】

J P E G 2 0 0 0 標準は、"JPEG 2000 Final Draft International Standard"というタイトルの文書 ISO/IEC FCD 15444-1 に記載されている。

【 0 0 6 4 】

主要な J P E G 2 0 0 0 符号化ステップは、

30

a) Y がルミナンスを表し、C r がクロミナンス成分赤を表し、C b がクロミナンス成分青を表すとき、空間 Y - C r - C b での画像の表現を得るため入力画像をオプション的に変換するステップと、

b) 画像成分のウェーブレット変換ステップと、

c) 獲得された係数の定量化ステップと、

d) 符号化ステップと、

e) ある種のプログレッシブ方式に従って、獲得されたデータをビットストリームに挿入するステップと、

を有する。

40

【 0 0 6 5 】

符号化すべき画像が非常に大きい場合、予備ステップは、画像を、相互に独立して符号化される様々なタイルに分割することができる。以下では、説明を簡単化するため、画像が単一のタイルにより構成されている場合に限定する。しかし、このような場合に限定されるものではない。

【 0 0 6 6 】

J P E G 2 0 0 0 は、解像度と、品質と、空間位置と、成分とを用いた 4 通りの基本プログレッシブ方式を使用する。J P E G 2 0 0 0 は、これらの基本方式の 5 通りの組み合わせを提案する。各プログレッシブ方式は、データをビットストリーム内で順序付ける特定の方式に対応する。

【 0 0 6 7 】

50

ビットストリームは、パケットヘッダと、符号化により得られた有効なデータを収容するパケットを含む。有効なデータの各パケットは、成分 (i) と、レイヤ (j) と、解像度のレベル (k) と、パーティション位置 (m) とに対応する。ビットストリームは、四つのループを用いて構築される。ループの順番は、使用されるプログレッシブ方式を決定する。

【 0 0 6 8 】

成分によるプログレッシブ性を備えたビットストリームの場合、4 個のループの順番は、例えば、成分毎、解像度毎、レイヤ毎、パーティション位置毎の順番で、パケットをビットストリームに書き込む。解像度によるプログレッシブ性を備えたビットストリームの場合、4 個のループの順番は、例えば、解像度毎、成分毎、レイヤ毎、パーティション位置毎の順番で、パケットをビットストリームに書き込む。以下の説明では、これらの 2 通りのプログレッシブ性の例に限定されているが、本発明は、あらゆるタイプのプログレッシブ性に適用可能である。

10

【 0 0 6 9 】

J P E G 2 0 0 0 の用語では、画像に関係するすべての情報は、コードストリームに合成される。コードストリームは、メイン・ヘッダと、一つ以上のタイル部ヘッダとを含み、各タイルヘッダ部は、後にビットストリームが続き、エンドマーカによって終了する。メイン・ヘッダ及びタイル部ヘッダは、マーカーとマーカー・セグメントに編成される。最終的に、コードストリーム自体は、バイナリ・ファイルに格納される。

【 0 0 7 0 】

図 5 は、J P E G 2 0 0 0 コードストリームの一例の構造の説明図である。このコードストリーム C S は、メイン・ヘッダ 1 0 0 から始まり、メイン・ヘッダ 1 0 0 は、マーカー S O C (コードストリームの始まり) と、マーカー・セグメント S I Z (画像及びタイルのサイズ) と、C O D (符号化スタイルデフォルト) と、Q C D (量子化デフォルト) と、を含む。コードストリーム C S は、次に、タイル部ヘッダ 1 1 0 を含み、タイル部ヘッダ 1 1 0 は、マーカー・セグメント S O T (タイル部の始まり) と、マーカー S O D (データの始まり) と、を含む。コードストリーム C S は、最後に、ビットストリーム 1 2 0 を含む。ビットストリーム 1 2 0 は、オプションとして、パケットの先頭にマークを付けるため使用されるマーカー S O P (パケットの始まり) が先行する種々のパケットの形式である。コードストリーム C S は、エンドマーカー E O C (コードストリームの終わり) によって終了する。ここで使用されている名称は、J P E G 2 0 0 0 勧告で使用されている名称である。より詳細には、マーカーは、2 個のオクテットからなるコードである。マーカー・セグメントは、マーカーと、マーカーに関連付けられたパラメータと、を含む。

20

30

【 0 0 7 1 】

次に、X M L 言語ベースについて説明する。この言語は、1998年2月10日に発行された W 3 C コンソーシアムの勧告 REC-xml-19980210 に記載されている。

【 0 0 7 2 】

物理的な観点から、X M L 文書は、解析可能なエンティティと、解析できないエンティティと、を含む。解析可能なエンティティは、テキスト、即ち、タグ若しくはテキストデータを表現し、所定の文字集団に属する文字のシーケンスを格納する。解析できないエンティティは、テキスト以外のもの、或いは、X M L テキストではないテキストを含む。

40

【 0 0 7 3 】

論理的な観点から、X M L 文書は、一つ以上の要素を含み、要素の境界は、オープニングタグとクローズングタグによってマークされている。要素は、相互にインターリーブしていても構わない。各要素は名前によって識別される。要素はコンテンツを収容してもよい。属性詳細の組を要素に関連付けることも可能である。各属性詳細は、名前と値を含む。同一のスタートタグに同じ属性名が複数回現れることはない。

【 0 0 7 4 】

次に、X M L 言語を用いて J P E G 2 0 0 0 コードストリームを構築する方法を説明する。

50

【 0 0 7 5 】

本例では、J P E G 2 0 0 0 ファイルの各マーカー・セグメント用の X M L 要素を作成するため、以下の規則を適用することによって選択が行われる。

- ・要素はマーカーの 3 文字コードの名前をとって命名される。
- ・マーカーの長さ、マーカーの値、並びに、可能であればマーカーに関連付けられた他のパラメータが要素の属性として定義される。
- ・パラメータ若しくはパラメータグループが繰り返されるとき、属性の代わりにサブ要素が作成される。なぜならば、X M L 言語は、同一スタートタグ内の同一名に 2 個以上の属性が与えられること（本例では、例えば、マーカー・セグメント S I Z が画像の各成分に対して繰り返される 3 個のパラメータ S s i z、X R s i z 及び Y R s i z のグループによって終端されること）を許可しないからである。
- ・ビットストリームのパケットは要素のコンテンツの形式で文書に組み込まれる。

10

【 0 0 7 6 】

しかし、ビットストリームは、全くのバイナリ・データにより構成されるので、（ある種の明瞭な文字だけを許容する）X M L 文書にビットストリームをそのまま組み入れることは不可能である。

【 0 0 7 7 】

第 1 の実施例では、ビットストリームは文字に変換される。この目的のため、有利的には、「ベース 6 4」という名称で知られている符号化方法が使用される。「ベース 6 4」は、I E T F によって発行された文書 RFC2045 の第 6.8 節に記載されている。この方法は、ビットストリームを、6 ビットのグループに分割し、6 ビットの各グループに、アルファベット文字からの文字を関連付ける。この方法は、データを 3 3 % ずつ拡張するという欠点がある。

20

【 0 0 7 8 】

第 2 の実施例では、ビットストリームを文字に変換し、X M L 文書に挿入するのではなく、ポインタが X M L 文書に導入され、このポインタは、基本ファイルの中でバイナリ・データを格納している領域を指定する。本実施例では、X M L 文書は、基本ファイルに依存する。

【 0 0 7 9 】

非限定的な例として、例 1 には、成分によるプログレッシブ性を説明する図 5 に表された J P E G 2 0 0 0 コードストリームを記述する X M L 文書が掲載されている。本例の場合、パケットの内容は、1 ラインだけを占有するように切り捨てられている。

30

【 0 0 8 0 】

次に、画像の改良バージョンを記述する変換された X M L 文書を作成するため、X M L 文書に適用される変換の二つの例を説明する。変換には、一般に、基本 X M L 文書のある種の要素を抑制すること、並びに、コードストリーム集合の一貫性が保たれるように、ある種の属性とある種の要素のうちの少なくとも一方の値を変更することが含まれる。

【 0 0 8 1 】

変換の第 1 例は、成分プログレッシブ性を用いて符号化された画像に適用可能な変換であり、カラー画像から階調画像まで通用する。このような変換は、クロミナンス成分に対応するビットストリームのパケットを抑制（隠蔽）する。この目的のため、この変換は、

40

- ・クロミナンスパケットに対応する要素 S O P を抑制するステップと、
 - ・クロミナンス成分に関係した属性 S s i z、X R s i z 及び Y R s i z を格納する要素を抑制するステップと、
 - ・成分の個数を示す要素 S I Z の属性 C s i z を変更するステップと（古い値 3 新しい値 1 ）、
 - ・マーカー・セグメント S I Z のサイズを示す要素 S I Z の属性 M a r k S e g L e n を変更するステップと（古い値 $3 \times 8 + 3 = 27$ 新しい値 $3 \times 8 + 3 = 27$ ）と、
- を含む。

【 0 0 8 2 】

50

このような変換は、有利的には、X S Lスタイル・シートに定義されている。非限定的な例として、例 2 には、例 1 に掲載された X M L 文書に適用される X S L スタイル・シートが掲載されている。

【 0 0 8 3 】

変換の第 2 例は、解像度によってプログレッシブ符号化された画像に適用される変換であり、大きいサイズの画像から小さいサイズの画像までの変化を許容する。このような変換は、獲得されるべき解像度の数を定義する入力パラメータ N_{resout} を使用する。この変換は、 N_{resout} よりも高い解像度に対応するビットストリームのパケットを抑制する。この目的のため、この変換は、

- ・ N_{res} 、 N_{comp} および N_{layers} がそれぞれプログレッシブ基本ファイルで使用される解像度の数、成分の数及びレイヤの数を表すとき、ビットストリームの $(N_{res} - N_{resout}) \cdot N_{comp} \cdot N_{layers}$ の最後のパケットに対応する要素 S O P を抑制するステップと、
 - ・ 画像のサイズに関する要素 S I Z の属性を変更するステップ (属性 X_{siz} 、 $X_{T_{siz}}$ 、 Y_{siz} 及び $Y_{T_{siz}}$ に含まれる値を $2^{N_{res} - N_{resout}}$ で除算) と、
 - ・ 分割レベルの数に関連した要素 C O D の属性を変更するステップ ($n_{DecompLevel} = N_{resout}$) と、
 - ・ 要素 Q C D の属性 S_{qcd} によって示された定量化スタイルに応じて、最後の要素 S P qcd を抑制するステップと、
 - ・ 属性 $markSegLen$ を要素 Q C D と適合させるステップと、
- を含む。

【 0 0 8 4 】

上述の変換の 2 例は限定的な例ではない。本発明は、他のタイプの変換にも適用される。例えば、レイヤによってプログレッシブ符号化された画像に適用され、種々の信号対雑音比を得ることができる変換を定義してもよい。

【 0 0 8 5 】

図 6 は、本発明による J P E G 2 0 0 0 フォーマットのバイナリ・ファイルの一例の説明図である。図 6 によると、J P E G 2 0 0 0 フォーマットのバイナリ・ファイル F F は、本発明における基本プログレッシブ・ファイルを構成するコードストリーム C S を格納したメイン・ボックス B X 1 を含む。J P E G 2 0 0 0 フォーマットのバイナリ・ファイル F F は、D O C という名称の X M L 文書を格納した補助的なボックス B X 2 を更に含む。文書 D O C は、コードストリーム C S を記述する。補助的なボックス B X 2 は、更に、文書 D O C を適用対象とし、T R という名前が付けられた変換 X S L を格納している。

【 0 0 8 6 】

【表 1】

例1:XML 文書(1ページ)

```

<?xml version="1.0" ?>
<Codestream>
  <MainHeader>
    <SOC markSegLen="-1" markerCodeStr="ff4f">
      </SOC>
    <SIZ Csize="3" RsizeStr="JPEG 2000 ~ Part I" Xsize="0" XTOsize="0" XTsize="515"
      Xsize="515" Ysize="0" YTOsize="0" YTsize="512" Ysize="512" markSegLen="47"
      markerCodeStr="ff51">
        <Comp_siz Ssize="7" Xsize="1" Ysize="1">
          </Comp_siz>
        <Comp_siz Ssize="7" Xsize="1" Ysize="1">
          </Comp_siz>
        <Comp_siz Ssize="7" Xsize="1" Ysize="1">
          </Comp_siz>
      </SIZ>
    <COD codeBlockHeightExp="4" codeBlockWidthExp="4" ephUsed="false"
      markSegLen="12" markerCodeStr="ff52" mct="1" nDecompLevel="5"
      numLayers="1" optByPass="false" optErTerm="false" optRegTerm="false"
      optResetMQ="false" optSegMarkers="false" optVertStrCausal="false"
      precinctPartitionIsUsed="false" progType="4" progTypeStr="Component"
      sopUsed="true" wavTrans="0" wavTransStr="9-7">
      </COD>
    <QCD Sqcd="2" markSegLen="35" markerCodeStr="ff5c" nGuardBits="2">
      <SPqcd spqcd_val="28440">
        </SPqcd>
      <SPqcd spqcd_val="28394">
        </SPqcd>
      <SPqcd spqcd_val="28394">
        </SPqcd>
      <SPqcd spqcd_val="28348">
        </SPqcd>
      <SPqcd spqcd_val="26368">
        </SPqcd>
      <SPqcd spqcd_val="26368">
        </SPqcd>
      <SPqcd spqcd_val="26338">
        </SPqcd>
      <SPqcd spqcd_val="24396">
        </SPqcd>
      <SPqcd spqcd_val="24396">
        </SPqcd>
      <SPqcd spqcd_val="24420">
        </SPqcd>
      <SPqcd spqcd_val="18435">
        </SPqcd>
      <SPqcd spqcd_val="18435">
        </SPqcd>
      <SPqcd spqcd_val="18501">
        </SPqcd>
      <SPqcd spqcd_val="20434">
        </SPqcd>
      <SPqcd spqcd_val="20434">
        </SPqcd>
      <SPqcd spqcd_val="20321">
        </SPqcd>
      </QCD>
    </MainHeader>
  </Codestream>

```

10

20

30

40

【 0 0 8 7 】

【 表 2 】

例1:XML 文書(2ページ)

```

<TilePartHeader>
  <SOT Isot="0" Psot="32722" TNsot="1" TPsot="0" markSegLen="10"
    markerCodeStr="ff90">
  </SOT>
  <SOD markSegLen="-1" markerCodeStr="ff93">
  </SOD>
  <SOP Lsop="0" Nsop="0" markSegLen="4" markerCodeStr="ff91">
    zokxfjAav/0eNh0KCgQ=
  </SOP>
  <SOP Lsop="0" Nsop="1" markSegLen="4" markerCodeStr="ff91">
    vih3XvSqAxdN7qp7Aubt+h1A5Iai7DNKpOQYs1JGcF+kax9m
  </SOP>
  <SOP Lsop="0" Nsop="2" markSegLen="4" markerCodeStr="ff91">
    ion5YDUL4HqwaIzDAg==
  </SOP>
  <SOP Lsop="0" Nsop="3" markSegLen="4" markerCodeStr="ff91">
    v5HtMkNwzWEIhKL7bEcpU5ZrplIfzr18kzkNLcm1Cw==
  </SOP>
  <SOP Lsop="0" Nsop="4" markSegLen="4" markerCodeStr="ff91">
    o3a71Cv0m1V/xPkZz6Ks7HXbJYUyLNnyi8JhJ0OeYgvi
  </SOP>
  <SOP Lsop="0" Nsop="5" markSegLen="4" markerCodeStr="ff91">
    4opHeHv9L+qiu/UplNkyVlPIWoc8umfk14Z6eDIzKNGtEvbKCMaknWAcYXjmf
  </SOP>
  <SOP Lsop="0" Nsop="6" markSegLen="4" markerCodeStr="ff91">
    EnEF62yGwcRtY2ehQYEpuQ==
  </SOP>
  <SOP Lsop="0" Nsop="7" markSegLen="4" markerCodeStr="ff91">
    ldWatY7LLQ==
  </SOP>
  <SOP Lsop="0" Nsop="8" markSegLen="4" markerCodeStr="ff91">
    b3qu8Ev6TqqoK4IItm19Clfij08VY2qqzMsWd7eSOQhyp5H1StP1wM0sB6i2b/j3JMo80vg=
  </SOP>
  <SOP Lsop="0" Nsop="9" markSegLen="4" markerCodeStr="ff91">
    83mg5+Jv95wRIRM61swVYBHw7eAeb2BNzmBsC1XrTjvzGFL5rY1BCyAn8CS
  </SOP>
  <SOP Lsop="0" Nsop="10" markSegLen="4" markerCodeStr="ff91">
    7psE6fJW2/G3ohpvNat+ovEFscTFj++9sdVip0XwnPn25oWq2yZYNh+DWr3T7H+V
  </SOP>
  <SOP Lsop="0" Nsop="11" markSegLen="4" markerCodeStr="ff91">
    XaILyKsuXvRI+C4p
  </SOP>
  <SOP Lsop="0" Nsop="12" markSegLen="4" markerCodeStr="ff91">
    W6qLbsuVSfjQRnGVfO27J97dmNDaUYjlttBepCvUXQu71TWIWBzRRROmPgpPBg==
  </SOP>
  <SOP Lsop="0" Nsop="13" markSegLen="4" markerCodeStr="ff91">
    3s+ECWT/ZjkqNaHDcQ==
  </SOP>
  <SOP Lsop="0" Nsop="14" markSegLen="4" markerCodeStr="ff91">
    +eQE2Ek04vydGK++tx9kKpj2L6TeiR2nc4crNQ4zHDurQg==
  </SOP>
  <SOP Lsop="0" Nsop="15" markSegLen="4" markerCodeStr="ff91">
    lcQa40/p/tH7FOjY0fjYefzcUtdrynh7RRVhBftsTu0F6Y5qXhD1D4G2uBd/a7
    uw49QaaPKtfJb8XiaVpKVqBW9LwoqQ==
  </SOP>
  <SOP Lsop="0" Nsop="16" markSegLen="4" markerCodeStr="ff91">
    ZjJalwvKx0SiIQUOsc/Rkj04wqeX6qg3miqkFQ==
  </SOP>
  <SOP Lsop="0" Nsop="17" markSegLen="4" markerCodeStr="ff91">
    gA==
  </SOP>
</TilePartHeader>
<EOC markSegLen="-1" markerCodeStr="ffd9">
</EOC>
</Codestream>

```

10

20

30

40

【 0 0 8 8 】

【 表 3 】

例 2: XSL スタイル・シート(1ページ)

```
<?xml version="1.0"?>
```

```
<!--
```

```
Remove color components :
```

```
For this, this stylesheet performs the following transformation:
```

- check that SOP markers are used
- check progression type
- check that multiple component transform is used
- check that nCompOut <= Csiz
- replace the Csiz attribute value by the wanted value (nCompOut)
- delete Comp_siz elements accordingly
- update markSegLen value accordingly
- delete SOP elements accordingly

10

```
Input parameter : nCompOut, default value = 1
```

```
-->
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:import href="tplCheckSOPUsed.xsl"/>
  <xsl:output method="xml" indent="yes"/>
```

```
<!-- Parameter: number of output components - Default value = 1 -->
<xsl:param name="nCompOut">1</xsl:param>
```

```
<!-- Match all: default template -->
<xsl:template name="tplAll" match="@*|node()">
  <xsl:copy>
    <xsl:apply-templates select="@*|node()"/>
  </xsl:copy>
</xsl:template>
```

20

```
<!-- Match COD/@sopUsed - Overrides tplAll -->
<xsl:template name="tplSopUsed" match="//COD/@sopUsed" priority="1">
  <!-- Check /COD/sopUsed value -->
  <xsl:call-template name="tplCheckSOPUsed"/>
</xsl:template>
```

```
<!-- Match COD - Overrides tplAll -->
<xsl:template name="tplCOD" match="COD" priority="1">
```

```
  <!-- Check progType value -->
  <xsl:if test="@progType!=4">
    <xsl:message terminate="yes">
      Error: progression type is <xsl:value-of select="@progType"/>
      Should be 4 (progression by color component)
    </xsl:message>
  </xsl:if>
```

30

```
  <!-- Check mct value -->
  <xsl:if test="@mct=0 and //SIZ/@Csiz=3">
    <xsl:message>
      Warning: no multi-component transf. was applied to input image.
      If input was RGB, output will be the Red component.
    </xsl:message>
  </xsl:if>
```

```
  <!-- Copy COD -->
  <xsl:copy>
    <xsl:apply-templates select="@*"/>
    <!-- Set mct value to 0 -->
    <xsl:attribute name="mct">0</xsl:attribute>
  </xsl:copy>
</xsl:template>
```

40

【 0 0 8 9 】

【 表 4 】

例 2: XSL スタイル・シート(2ページ)

```

<!-- Match SIZ - Overrides tplAll -->
<xsl:template name="tplSIZ" match="SIZ" priority="1">

    <!-- Check nCompOut value : should be <= Csiz -->
    <xsl:if test="($nCompOut > @Csiz) or ($nCompOut < 1)">
        <xsl:message terminate="yes">
            Error: number of output color components should be >= 1
            and < <xsl:value-of select="@Csiz"/>. Exit...
        </xsl:message>
    </xsl:if>

    <xsl:copy>
        <!-- Copy attributes -->
        <xsl:apply-templates select="@*" />

        <!-- Update Csiz value -->
        <xsl:attribute name="Csiz">
            <xsl:value-of select="$nCompOut" />
        </xsl:attribute>

        <!-- Update markSegLen value accordingly -->
        <xsl:attribute name="markSegLen">
            <xsl:value-of select="38 + 3 * $nCompOut" />
        </xsl:attribute>

        <!-- Copy only relevant Comp_siz elements -->
        <xsl:apply-templates select="Comp_siz[position() <= $nCompOut]" />

    </xsl:copy>
</xsl:template>

<!-- Match TilePartHeader - Overrides tplAll -->
<xsl:template name="tplTilePartHeader" match="TilePartHeader" priority="1">

    <!-- Calculate number of output packets -->
    <xsl:variable name="nSOPOut">
        <xsl:value-of select="
            " (//COD/@nDecompLevel + 1) * //COD/@numLayers * $nCompOut" />
    </xsl:variable>

    <xsl:copy>
        <!-- Copy attributes -->
        <xsl:apply-templates select="@*|SOT|SOD" />

        <!-- Remove SOP packets accordingly -->
        <xsl:apply-templates select="SOP[position() <= $nSOPOut]" />
        <xsl:message>
            Initial number of SOP : <xsl:value-of
select="count(SOP)" />
            Output number of SOP : <xsl:value-of
select="$nSOPOut" />
        </xsl:message>
    </xsl:copy>
</xsl:template>

</xsl:stylesheet>

```

【 0 0 9 0 】

【 表 5 】

例2:XSL スタイル・シート(3ページ)

```

<?xml version="1.0"?>

<!--
  This stylesheet performs the following transformation:
  - check than "sopUsed" attribute of "COD" element is set to yes,
    otherwise, exit
-->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <!-- Template tplCheckSOPUsed -->
  <xsl:template name="tplCheckSOPUsed">

    <xsl:variable name="sopUsed">
      <xsl:value-of select="//COD/@sopUsed"/>
    </xsl:variable>

    <xsl:if test="$sopUsed!='true'">
      <xsl:message terminate="yes">
        Error: SOP markers are not used in this codetsream.
        Cannot process. Exit...
      </xsl:message>
    </xsl:if>
    <xsl:copy/>

  </xsl:template>

</xsl:stylesheet>

```

10

20

【図面の簡単な説明】

【図 1】 本発明によるファイル転送方法の種々のステップを説明するブロック図である。

【図 2】 本発明による伝送システムの一例の構成図である。

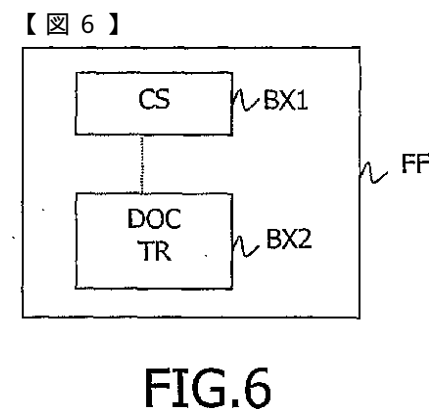
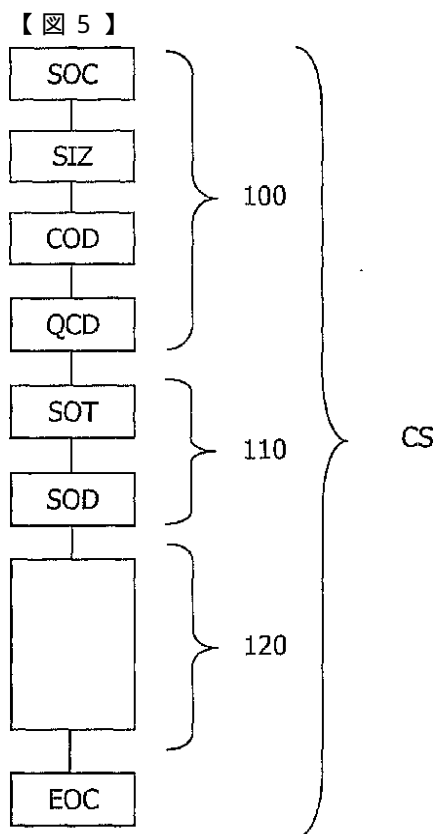
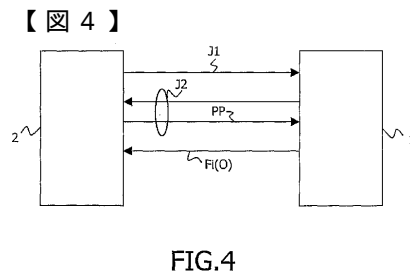
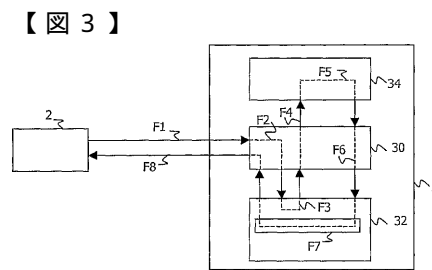
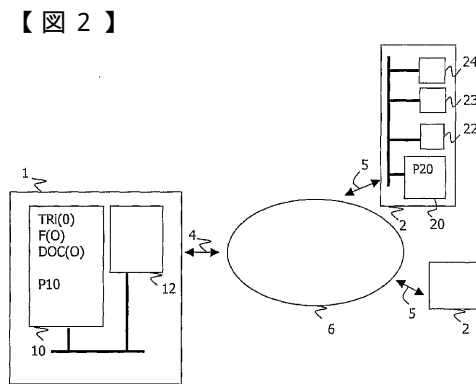
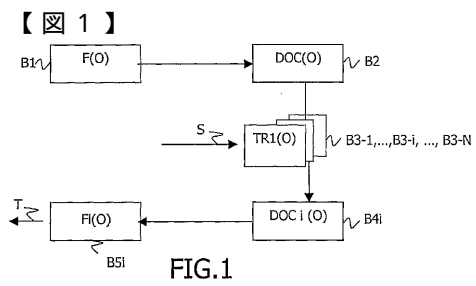
【図 3】 本発明による伝送システムのサーバのソフトウェア構造の一例の説明図である。

30

【図 4】 図 3 に示されているようなクライアント・サーバ型アーキテクチャを利用する本発明によるファイル転送方法のアプリケーションの一例の説明図である。

【図 5】 J P E G 2 0 0 0 コードストリームの概略的な説明図である。

【図 6】 本発明によるバイナリ・ファイルの一例の概略的な説明図である。



フロントページの続き

(51)Int.Cl. F I
G 0 6 F 13/00 5 5 0 L

(72)発明者 デヴィエール, シルヴァン
オランダ国, 5 6 5 6 アーアー アインドーフエン, プロフ・ホルストラーン 6

合議体

審判長 小曳 満昭

審判官 久保 正典

審判官 長島 孝志

(56)参考文献 特開 2 0 0 0 - 2 2 2 2 7 6 (J P , A)
特開 2 0 0 0 - 2 2 2 2 7 5 (J P , A)
特開 2 0 0 0 - 2 0 5 0 7 (J P , A)
特開平 1 1 - 6 6 0 5 1 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
G06F17/20-17/26
G06F13/00