(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4072285号
(P4072285)

(45)発行日　平成20年4月9日(2008.4.9)　(24)登録日　平成20年1月25日(2008.1.25)

(51)Int.Cl.		F I		
G06F 12/00	(2006.01)	G06F	12/00	520E
G06F 17/30	(2006.01)	G06F	17/30	230Z
H04N 5/92	(2006.01)	H04N	5/92	H

請求項の数 17　(全 14 頁)

(21)出願番号　特願平11-105768
(22)出願日　平成11年4月13日(1999.4.13)
(65)公開番号　特開2000-298606(P2000-298606A)
(43)公開日　平成12年10月24日(2000.10.24)
審査請求日　平成16年6月3日(2004.6.3)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100093908
弁理士　松本　研一
(74)代理人　100101306
弁理士　丸山　幸雄

最終頁に続く

(54)【発明の名称】データ処理方法及び装置及び記憶媒体

(57)【特許請求の範囲】

【請求項１】

バイナリデータにメタデータを登録するデータ処理方法であって、

メタデータの付与対象のバイナリデータを読み込む第１読込工程と、

前記バイナリデータに付与すべきメタデータを読み込む第２読込工程と、

前記第２読込工程で読み込まれたメタデータに基づいて確認情報を生成する生成工程と、

前記第１読込工程で読み込まれたバイナリデータの後に、前記生成手段で生成された確認情報と前記第２読込工程で読み込まれたメタデータを接続する接続工程と、

前記接続工程によって得られたデータの全体を一つのファイルとして出力する出力工程とを備えることを特徴とするデータ処理方法。

【請求項２】

前記生成工程は、前記第２読込工程で読み込まれたメタデータについて複数種類の確認情報を生成することを特徴とする請求項１に記載のデータ処理方法。

【請求項３】

前記生成工程で使用される確認情報の種別を表わす情報を前記第２工程で読み込まれたメタデータに追記する追記工程を更に備える　ことを特徴とする請求項１に記載のデータ処理方法。

【請求項４】

前記第２読込工程で読み込まれたメタデータが、所定のデータ記述言語における適正な

形式で記述されているか否かを判定する判定工程を更に備え、
前記生成工程と接続工程は、前記判定工程で適正な形式で記述されていると判定された場合に実行されることを特徴とする請求項１に記載のデータ処理方法。
【請求項５】
前記判定工程は、前記メタデータが前記所定のデータ記述言語としての正当性を満足するか否かを含めて判定することを特徴とする請求項４に記載のデータ処理方法。
【請求項６】
メタデータが登録されたバイナリデータにおいてメタデータを判別する方法であって、
データを読み込む読込工程と、
前記読込工程で読み込まれたデータを末尾より検査し、所定のデータ記述言語の所定形式で記述されたデータが存在するか否かを判定する第１判定工程と、
前記第１判定工程で所定形式のデータが存在すると判定された場合に、該所定形式で記述されたデータに基づいて確認情報を生成する生成工程と、
前記生成工程で生成された確認情報と、前記読込工程で読み込まれたデータ中の前記所定形式で記述されたデータに対して所定の位置に格納された確認情報とを比較する比較工程と、
前記比較工程の比較結果に基づいて該所定形式で記述されたデータをメタデータであると判別する判別工程とを備えることを特徴とするデータ処理方法。
【請求項７】
前記確認情報が格納される所定の位置は、前記所定形式で記述されたデータの直前であることを特徴とする請求項６に記載のデータ処理方法
【請求項８】
前記判別工程においてメタデータが判別された場合、判別されたメタデータを抽出して出力する出力工程を更に備えることを特徴とする請求項６に記載のデータ処理方法。
【請求項９】
前記判別工程は、
前記所定のデータ記述言語に規定された末尾文字列が前記データの末尾に存在するか否かをチェックするチェック工程と、
該末尾文字列が存在する場合に前記所定のデータ記述言語に規定された先頭文字列を該データの先頭へ向かって検索する検索工程と、
前記検索工程で前記先頭文字列が検索された場合、該先頭文字列と前記末尾文字列との間のデータが、前記所定のデータ記述言語における適正な形式を有するか否かを検査する検査工程とを備えることを特徴とする請求項６に記載のデータ処理方法。
【請求項１０】
バイナリデータにメタデータを登録するデータ処理装置であって、
メタデータの付与対象のバイナリデータを読み込む第１読込手段と、
前記バイナリデータに付与すべきメタデータを読み込む第２読込手段と、
前記第２読込手段で読み込まれたメタデータに基づいて確認情報を生成する生成手段と、
前記第１読込手段で読み込まれたバイナリデータの後に、前記生成手段で生成された確認情報と前記第２読込手段で読み込まれたメタデータを接続する接続手段と、
前記接続手段によって得られたデータの全体を一つのファイルとして出力する出力手段とを備えることを特徴とするデータ処理装置。
【請求項１１】
前記第２読込手段で読み込まれたメタデータが、所定のデータ記述言語における適正な形式で記述されているか否かを判定する判定手段を更に備え、
前記生成手段と接続手段は、前記判定手段で適正な形式で記述されていると判定された場合に動作することを特徴とする請求項１０に記載のデータ処理装置。
【請求項１２】
メタデータが登録されたバイナリデータにおいてメタデータを判別する装置であって、

データを読み込む読込手段と、
前記読込手段で読み込まれたデータを末尾より検査し、所定のデータ記述言語の所定形式で記述されたデータが存在するか否かを判定する第１判定手段と、
前記第１判定手段で所定形式のデータが存在すると判定された場合に、該所定形式で記述されたデータに基づいて確認情報を生成する生成手段と、
前記生成手段で生成された確認情報と、前記読込手段で読み込まれたデータ中の前記所定形式で記述されたデータに対して所定の位置に格納された確認情報とを比較する比較手段と、
前記比較手段の比較結果に基づいて該所定形式で記述されたデータをメタデータであると判別する判別手段とを備えることを特徴とするデータ処理装置。
【請求項１３】
前記確認情報が格納される所定の位置は、前記所定形式で記述されたデータの直前であることを特徴とする請求項１２に記載のデータ処理装置。
【請求項１４】
前記判別手段においてメタデータが判別された場合、判別されたメタデータを抽出して出力する出力手段を更に備えることを特徴とする請求項１２に記載のデータ処理装置。
【請求項１５】
前記判別手段は、
前記所定のデータ記述言語に規定された末尾文字列が前記データの末尾に存在するか否かをチェックするチェック手段と、
該末尾文字列が存在する場合に前記所定のデータ記述言語に規定された先頭文字列を該データの先頭へ向かって検索する検索手段と、
前記検索手段で前記先頭文字列が検索された場合、該先頭文字列と前記末尾文字列との間のデータが、前記所定のデータ記述言語における適正な形式を有するか否かを検査する検査手段とを備えることを特徴とする請求項１２に記載のデータ処理装置。
【請求項１６】
バイナリデータにメタデータを登録する処理をコンピュータに実行させるためのプログラムを格納したコンピュータ読み取り可能な記憶媒体であって、
メタデータの付与対象のバイナリデータを読み込む第１読込工程と、
前記バイナリデータに付与すべきメタデータを読み込む第２読込工程と、
前記第２読込工程で読み込まれたメタデータに基づいて確認情報を生成する生成工程と、
前記第１読込工程で読み込まれたバイナリデータの後に、前記生成手段で生成された確認情報と前記第２読込工程で読み込まれたメタデータを接続する接続工程と、
前記接続工程によって得られたデータの全体を一つのファイルとして出力する出力工程と、をコンピュータに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。
【請求項１７】
メタデータが登録されたバイナリデータにおいてメタデータを判別する処理をコンピュータに実行させるためのプログラムを格納したコンピュータ読み取り可能な記憶媒体であって、
データを読み込む読込工程と、
前記読込工程で読み込まれたデータを末尾より検査し、所定のデータ記述言語の所定形式で記述されたデータが存在するか否かを判定する第１判定工程と、
前記第１判定工程で所定形式のデータが存在すると判定された場合に、該所定形式で記述されたデータに基づいて確認情報を生成する生成工程と、
前記生成工程で生成された確認情報と、前記読込工程で読み込まれたデータ中の前記所定形式で記述されたデータに対して所定の位置に格納された確認情報とを比較する比較工程と、
前記比較工程の比較結果に基づいて該所定形式で記述されたデータをメタデータである

と判別する判別工程と、をコンピュータに実行させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明はバイナリデータとメタデータを扱うデータ処理方法及び装置及び記憶媒体に関する。

【０００２】

メタデータ（meta-data）とは、「データに関するデータ」であり、画像データや音声データ等のバイナリデータを説明するデータとして用いられている。しかし、バイナリデータとこれに対応するメタデータが別々のファイルで存在した場合、ファイルの移動やコピーの際に、ユーザはバイナリデータとメタデータとを同時に管理しなければならず、非常にわずらわしいことになる。

【０００３】

そこで一般に、バイナリデータとメタデータの管理を容易にするために、バイナリデータとメタデータを記述する様々な方法が提案されてきた。この種の従来技術は、新しいバイナリフォーマットを規定する方法と、データベースで管理する方法の２つに分けることができる。

【０００４】

まず、新しいバイナリフォーマットを規定する方法の一例をあげると、画像フォーマットではTiff、Exif、Flashpixなどがある。図７は、バイナリデータにメタデータを埋め込んだフォーマットの概観を示す図である。バイナリデータとしては、例えば画像データが挙げられる。図７に示されるように、画像のヘッダ部分にメタデータを記述する枠組みを設け、そこにユーザがメタデータを記述するというのが一般的な方法である。このようにメタデータを記述することにより、データの検索・分類が容易になる。また、バイナリデータ内にメタデータを含むようになるので、１つのファイルで管理でき、ファイルの管理は比較的容易になる。

【０００５】

次に、バイナリデータとメタデータをデータベースで管理する方法を説明する。図８はバイナリデータとメタデータをデータベースで管理する方法を概念的に示した図である。図８に示されるような、別々のファイルで存在するバイナリデータとメタデータをデータベース等を用いて管理するという方法も広く行われているものである。この場合は既存のバイナリデータが、既存のアプリケーションでそのまま使えるという利点がある。

【０００６】

【発明が解決しようとする課題】

しかしながら、上述したようなメタデータを記述する新フォーマットを規定する方法とデータベースを用いてメタデータを管理する方法のそれぞれに問題がある。

【０００７】

まず、メタデータを記述する新フォーマットを規定した場合には、既存のバイナリデータを当該新フォーマットに変換し、なおかつその新フォーマット内にメタデータを記述しなければならない。更に、その新フォーマット内のメタデータを用いて検索するためには、当該新フォーマット対応のアプリケーションが必要となる。すなわち、メタデータを記述したり利用したりするために、非常に多くのステップと専用の環境が必要になるという問題がある。また、このような新フォーマットのバイナリデータを処理する（例えば画像データであれば画像の再生）ためには、当該フォーマットに対応したアプリケーションが必要であり、既存のアプリケーションでは対応できなくなる。

【０００８】

そのうえ、メタデータの記述方法も新フォーマットにおいて独自に決められたものであり、新フォーマット内のメタデータを利用するアプリケーションを作成するためには、新規にメタデータの検索ルーチンをつくらなければならないという問題もある。さらに、新し

い枠組みのメタデータを記述するにはフォーマットの規定を変更しなければならないという問題点もあった。

【０００９】

一方、データベースを用いてバイナリデータとメタデータを同時に管理する場合、データベースソフトが無ければメタデータの登録も利用もできないという問題があった。また、登録したメタデータを表示するためにも専用のソフトウェアが必要である。更に、バイナリデータをデータベース外に持っていくと、メタデータは付加されず、メタデータのないバイナリデータとなってしまうという問題点もあった。

【００１０】

本発明はメタデータの記述・検索に関する上記の問題点に鑑みてなされたものであり、既存のアプリケーションに影響を与えずに、バイナリデータにメタデータを登録可能とすることを目的とする。

【００１１】

また、本発明の他の目的は、メタデータが登録されたバイナリデータを、既存のアプリケーションで処理することが可能な形態で提供可能とすることにある。

【００１２】

また、本発明の他の目的は、メタデータの記述に一般的なデータ記述言語を用いることにより、データ記述言語用の既存のツールを利用することを可能とし、対応アプリケーションの開発を容易にすることにある。

【００１３】

さらに、本発明の他の目的は、メタデータが記述されたバイナリデータからメタデータを抽出し、例えば検索、参照、変更等の処理に供することを可能とすることにある。

【００１４】

また、本発明の他の目的は、チェックサムの確認情報をメタデータとともに登録しておき、この確認情報を用いてメタデータの判別を行うことにより、より正確なメタデータの判別を可能とすることにある。

【００１５】

【課題を解決するための手段】

上記の目的を達成するための本発明のデータ処理方法はたとえば以下の工程を備える。すなわち、

バイナリデータにメタデータを登録するデータ処理方法であって、

メタデータの付与対象のバイナリデータを読み込む第１読込工程と、

前記バイナリデータに付与すべきメタデータを読み込む第２読込工程と、

前記第２読込工程で読み込まれたメタデータに基づいて確認情報を生成する生成工程と、

前記第１読込工程で読み込まれたバイナリデータの後に、前記生成手段で生成された確認情報と前記第２読込工程で読み込まれたメタデータを接続する接続工程と、

前記接続工程によって得られたデータの全体を一つのファイルとして出力する出力工程とを備える。

【００１６】

また、上記の目的を達成するための本発明の他の態様によれば、以下の工程を備えたデータ処理方法が提供される。すなわち、

メタデータが登録されたバイナリデータにおいてメタデータを判別する方法であって、

データを読み込む読込工程と、

前記読込工程で読み込まれたデータを末尾より検査し、所定のデータ記述言語の所定形式で記述されたデータが存在するか否かを判定する第１判定工程と、

前記第１判定工程で所定形式のデータが存在すると判定された場合に、該所定形式で記述されたデータに基づいて確認情報を生成する生成工程と、

前記生成工程で生成された確認情報と、前記読込工程で読み込まれたデータ中の前記所定形式で記述されたデータに対して所定の位置に格納された確認情報とを比較する比較工程と、

前記比較工程の比較結果に基づいて該所定形式で記述されたデータをメタデータであると判別する判別工程とを備える。

【００１７】

また、本発明の他の態様によれば、上記の方法を実現するデータ処理装置が提供される。また、本発明の他の態様によれば、上記の方法をコンピュータによって実現するための制御プログラムが格納された記憶媒体が提供される。

【００１８】

【発明の実施の形態】

以下、添付の図面を参照して本発明の好適な実施形態を説明する。

【００１９】

＜第１の実施形態＞

図１は第１の実施形態によるデータ処理装置の構成を示すブロック図である。図１において、１００は読込部であり、スキャナ装置などを用いて画像を読み込む。１０１は入力部であり、ユーザからの指示やデータを入力するもので、キーボードやポインティング装置を含む。１０２は蓄積部であり、バイナリデータやメタデータを蓄積する。蓄積部１０２としては、ハードディスクを用いるのが一般的であろう。１０３は表示部であり、蓄積部１０２に蓄積されたバイナリデータを表示したり、読込部１００で読み込まれた画像データを表示する。表示部１０３としては、ＣＲＴや液晶表示装置が一般的である。

【００２０】

１０４はＣＰＵであり、上述した各構成の処理のすべてに関わり、ＲＯＭ１０５とＲＡＭ１０６はその処理に必要なプログラム、データ、或いは作業領域をＣＰＵ１０４に提供する。なお、図２のフローチャートを参照して後述する本実施形態の処理手順を実現するための制御プログラムもＲＯＭ１０５に格納されているものとする。もちろん、蓄積部１０２にその制御プログラムを格納しておき、ＣＰＵ１０４による実行に応じてその制御プログラムがＲＡＭ１０６上へロードされるような構成であってもよい。

【００２１】

なお、第１の実施形態のデータ処理装置には上記以外にも、種々の構成要素が設けられているが、本発明の主眼ではないので、その説明については省略する。

【００２２】

つぎに、以上のように構成されたデータ処理装置において、メタデータをバイナリデータに登録する処理について説明する。図２は、第１の実施形態によるメタデータの登録処理を説明するフローチャートである。

【００２３】

図２において、まず、ステップＳ３０１で、ユーザによって指定されたバイナリデータをメモリ（ＲＡＭ１０６）上に読み込む。これは例えば所望のバイナリデータファイル名をキーボードから入力したり、ポインティング装置（例えばマウス）によって当該バイナリデータのアイコンを指示することによりなされる。次にステップＳ３０２において、ユーザによって指定された、メタデータが記述されているＸＭＬファイルをメモリ（ＲＡＭ１０６）上に読み込む。このＸＭＬファイルの指定も、キーボードからファイル名を入力したり、ポインティング装置（例えばマウス）で対応するアイコンを指示する等によって行われる。

【００２４】

次にステップＳ３０３で、メタデータを記述したＸＭＬファイルが適正形式のＸＭＬデータであるかを調べる。この適性形式の判定では、ＸＭＬファイルの記述フォーマットを満足しているか（例えば、タグの左右の括弧が正しく対をなしているか、タグ付けの形式が正しいか等）がチェックされる。なお、適性形式のＸＭＬデータであるか否かの判定は、正当なＸＭＬデータであるか否かを含めたチェックであってもよい。ここで、正当なＸＭＬデータか否かの判定は、例えば、ＸＭＬデータがＤＴＤ（Document Type Definition）等のスキーマに従って記述されているか等のチェックを行うことでなされる。

【００２５】

ステップＳ３０３において適正形式のＸＭＬデータでないと判定された場合にはステップＳ３０５に進む。ステップＳ３０５では、ＸＭＬデータにエラーがある旨を表示部１０３に表示し、本処理を終了する。

【００２６】

一方、ステップＳ３０３においてＸＭＬファイルが適正形式のＸＭＬデータであると判定された場合には、処理はステップＳ３０４に進む。ステップＳ３０４では、当該メタデータのチェックサムを算出する。チェックサムとしては、メタデータのサイズ（バイト数）、文字数、単語数、行数、タグで囲まれた項目の数など、メタデータから一意に決定できる数値を用いる。また、これらのうちの２つ以上を同時に用いるようにすれば、より確度の高いチェックサムを構成できる。また、第２の実施形態で後述するメタデータの判別処理のために、チェックサムとしてどの数値を用いたか（或いはチェックサムの算出法）をＸＭＬデータとして記述しておくようにしてもよい。

【００２７】

次に、ステップＳ３０６において、ステップＳ３０１で読み込まれたバイナリデータの後尾にメタデータとチェックサムを登録する。さらに、ステップＳ３０７でメタデータを登録したバイナリデータを出力し、処理を終了する。

【００２８】

図３は本実施形態によるバイナリデータへのメタデータの登録状態を説明する図である。図３に示されるように、バイナリデータの最後に、チェックサム及びＸＭＬデー他形式で記述されたメタデータが接続される。こうすることによって、他のアプリケーションには影響を与えずに、メタデータを登録することができる。具体的には、バイナリデータが標準的なＪＰＥＧ画像データであったとすると、末尾にメタデータを付け加えても市販の（メタデータを認識しない）画像アプリケーションで障害なく画像を読み込むことができる。すなわち、バイナリデータのヘッダー部分の情報はメタデータの接続前から変化しないので、例えばバイナリデータが画像データであった場合には、一般的なブラウザによって画像再生が行える（接続されたメタデータは無視される）。

【００２９】

さらに、メタデータはＸＭＬで記述されているため、このＸＭＬデータ部分を抽出しておくことにより、ＸＭＬデータを理解するツールがあれば、メタデータの追加・変更・参照が可能であり、非常に汎用性に優れている。なお、ＸＭＬデータ部分の抽出については第２の実施形態で詳しく説明する。

【００３０】

以上説明したように、第１の実施形態によれば、メタデータをＸＭＬで記述し、このＸＭＬデータをバイナリデータの最後に接続することにより、既存のバイナリデータにメタデータを登録することができる。

【００３１】

そして、第１の実施形態によれば、所定のデータ記述言語における適正形式で記述されたメタデータ（ＸＭＬ形式のデータとチェックサム）をバイナリデータの最後に接続することにより、既存のアプリケーションに影響を及ぼすことなく、既存のバイナリデータにメタデータを登録することが可能となる。すなわち、メタデータが登録されたバイナリデータを、既存のアプリケーションで処理することが可能な形態で提供することができる。また、メタデータとして既存のデータ記述言語を用いれば、メタデータの編集、参照等に際して、当該データ記述言語用の既存のツールをそのまま用いることができ、開発に関する手間も省くことができる。

【００３２】

なお、上記実施形態ではＸＭＬデータの正当性をチェックしたが、Well-formedであるかというチェックに置き換えてもよい。また、メタデータ形式はＸＭＬに限らず、ＳＧＭＬ，ＨＴＭＬなど、他の形式でもよい。

【００３３】

＜第２の実施形態＞

第１の実施形態においてバイナリデータにメタデータを登録する方法を説明した。第２の実施形態では、バイナリデータにメタデータが登録されているかどうかを判別し、登録されている場合にはそのメタデータを抽出する処理について説明する。なお、第２の実施形態におけるデータ処理装置の構成は第１の実施形態（図１）と同様であるのでここでは説明を省略する。

【００３４】

以下、指定されたファイルのデータに第１の実施形態で説明した如きメタデータが登録されているか否かの判定と、登録されたメタデータを抽出する動作について説明する。図４は第２の実施形態による登録されたメタデータの判別及び抽出手順を示すフローチャートである。なお、本実施形態では、抽出されたメタデータを表示部１０３に表示するが、出力の形態はこれに限らない。例えば、抽出したメタデータを検索処理に提供するように構成してもよいことは当業者には明らかであろう。

【００３５】

図４によれば、まず、ステップＳ５０１で、ユーザの指示により、メタデータが登録されているかを判別したいファイルを指定する。ステップＳ５０１における、処理対象となるファイル（処理対象データ）の指定は、キーボードから当該処理対象データのファイル名を入力したり、対応するアイコンをポインティング装置（マウス）で指示することにより行われる。

【００３６】

次にステップＳ５０２において、指定されたファイルのデータにＸＭＬで記述されたメタデータが登録されているかどうかを判別する。以下、ステップＳ５０２における判別処理の詳細について図５のフローチャートと、図６の概略図にしたがって説明する。図５は第２の実施形態によるメタデータの判別処理の詳細を説明するフローチャートである。また、図６はメタデータとしてＸＭＬデータが登録された処理対象データのデータ構成例を示す図である。

【００３７】

第１の実施形態で説明したように、メタデータとしてのＸＭＬデータが登録されているバイナリデータのデータ構成は図６のようになっている。したがって、メタデータの有無の判別は以下のように行われる。

【００３８】

図５に示されるように、まず、ステップＳ６０１で、ステップＳ５０１で指定されたファイルのデータ全体をメモリ（ＲＡＭ１０６）上に読み込む。なお、第１の実施形態のステップＳ３０６によって出力されたデータは一つのファイルとして管理されるので、一般的なファイル管理システムによってこの処理対象データの全体を読出すことが可能である。

【００３９】

次にステップＳ６０２において、ステップＳ６０１で読み込んだデータの最後に“＜/PhotoXML＞”という文字列があるか調べる。存在しなかった場合はステップＳ６０８に進み、当該処理対象データにメタデータは登録されていないものと結論づける。

【００４０】

一方、読み込んだデータの最後に、“＜/PhotoXML＞”という文字列が存在した場合はステップＳ６０３にすすむ。ステップＳ６０３では“＜/PhotoXML＞”という文字列の前に“＜PhotoXML＞”という文字列が存在するかどうかを調べ、さらにそれらの文字列で囲まれたデータが、ＸＭＬの適正形式で記述されているかを確認する。なお、このとき、ＸＭＬの正当なデータであるか否かの判定を含めて行うようにしてもよい。適性形式か否かの判定、正当なデータか否かの判定は、第１の実施形態（ステップＳ３０３）で説明したとおりである。

【００４１】

ステップＳ６０３において適正形式であることが確認された場合は、ステップＳ６０４にすすむ。ステップＳ６０４において、文字列＜PhotoXML＞の直前に格納されているチェックサムを読み込み、これをチェックサム１として保持する。続くステップＳ６０５におい

て、上記適正形式であることが確認されたメタデータのチェックサムを計算し、これをチェックサム２とする。ここで、チェックサムの計算方法は実施例１で説明したメタデータ接続時の処理と同じにする。例えばメタデータ接続時に「文字数」をチェックサムとしたときは、本処理においても「文字数」をチェックサムとして計算する。

【００４２】

次に、ステップＳ６０６においてチェックサム１とチェックサム２を比較し、両者が等しければステップＳ６０７へ進み、メタデータが登録されている結論づける。一方、チェックサムが等しくなければステップＳ６０８に進む。ステップＳ６０２で、当該バイナリデータの最後に文字列“＜/PhotoXML＞”が存在しない場合、ステップＳ６０３で文字列“＜PhotoXML＞”が存在しない場合、ステップＳ６０３で内部の記述が適正でないと判定された場合、或いはステップＳ６０７でチェックサムが一致しない場合は、処理はステップＳ６０５にすすみ、当該処理対象データにメタデータは登録されていないものと結論づける。

以上で、メタデータの判別を終了する。

【００４３】

次に、図４のフローチャートにもどる。上記の図５のフローチャートで示される処理によってメタデータが登録されていると結論づけられた場合には、処理はステップＳ５０３に進む。ステップＳ５０３では、文字列“＜PhotoXML＞”と“＜/PhotoXML＞”で囲まれた部分のＸＭＬデータに基づいて登録されているメタデータの内容を表示し、処理を終了する。一方、ステップＳ５０２でメタデータが登録されていないと判定された場合にはそのまま処理を終了する。

【００４４】

以上説明したように、第２の実施形態によれば、メタデータ付きのバイナリデータと通常のバイナリデータとの判別を、データの末尾にＸＭＬデータが適正形式で記述されているか否かによって判別することが可能となる。また、メタデータが判別された場合には、そのメタデータを表示することが可能となる。

【００４５】

すなわち、第２の実施形態によれば、メタデータが登録されたバイナリデータとメタデータが登録されていないバイナリデータとを判別するとともに、登録されたメタデータを抽出することが可能となる。従って、メタデータとして既存のデータ記述言語を用いれば、メタデータを用いた検索に際して、当該データ記述言語用の既存のツールをそのまま用いることができ、開発に関する手間も省くことができる。

【００４６】

また、上記実施形態によれば、メタデータの判別のためにチェックサム等のチェック機構を設けてあるので、より確実にメタデータを判別することができる。

【００４７】

なお、上記各実施形態では、メタデータとしてＸＭＬデータを用いたがこれに限られるものではない。例えば、ＳＧＭＬやＨＴＭＬ等のデータ記述言語であってもよい。もちろん、これらのデータの存在を検出するために用いられる文字列（実施形態では＜PhotoXML＞と＜/PhotoXML＞を用いている）や正当性の判定は使用されるデータ記述言語等によって変わるものであり、実施形態によって限定さる者ではない。また、バイナリデータとしては静止画像データ、動画像データ、音声データ等が挙げられる。

【００４８】

また、チェックサムは第１の実施形態で説明したように、種々のものを利用可能である。また、メタデータの本体内（上記実施形態ではＸＭＬデータ内）にチェックサムの態様（例えば、「文字数」等）を記述するようにしておいて、データ判別の際にはその記述に従ってチェックサムを計算するようにしてもよい。

【００４９】

また、上記第２の実施形態において、チェックサムを用いてメタデータの有無を判定したが、チェックサムによる判定の後に、ＸＭＬ自体のもつ整合性（ValidateもしくはWell-F

ormed）を確認することにより、より正確な判定を行うことができる。

【００５０】

なお、本発明は、複数の機器（例えばホストコンピュータ，インタフェイス機器，リーダ，プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機，ファクシミリ装置など）に適用してもよい。

【００５１】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。

【００５２】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【００５３】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク，ハードディスク，光ディスク，光磁気ディスク，ＣＤ－ＲＯＭ，ＣＤ－Ｒ，磁気テープ，不揮発性のメモリカード，ＲＯＭなどを用いることができる。

【００５４】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【００５５】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【００５６】

【発明の効果】

以上説明したように、本発明によれば、既存のアプリケーションに影響を与えずに、バイナリデータにメタデータを登録することが可能となる。

また、本発明によれば、メタデータが登録されたバイナリデータを、既存のアプリケーションで処理することが可能となる。

また、本発明によれば、メタデータの記述に一般的なデータ記述言語を用いることにより、データ記述言語用の既存のツールを利用することが可能となり、対応アプリケーションの開発が容易になる。

さらに、本発明によれば、メタデータが記述されたバイナリデータからメタデータを抽出し、例えば検索、参照、変更等の処理に供することが可能となる。

さらに、本発明によれば、チェックサムの確認情報をメタデータとともに登録しておき、この確認情報を用いてメタデータの判別を行うので、より正確なメタデータの判別が可能となる。

【図面の簡単な説明】

【図１】第１の実施形態によるデータ処理装置の構成を示すブロック図である。

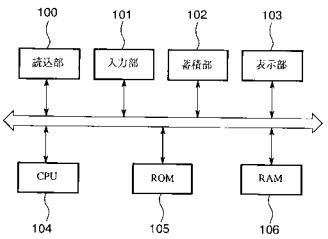
【図２】第１の実施形態によるメタデータの登録処理を説明するフローチャートである。

【図３】本実施形態によるバイナリデータへのメタデータの登録状態を説明する図である。

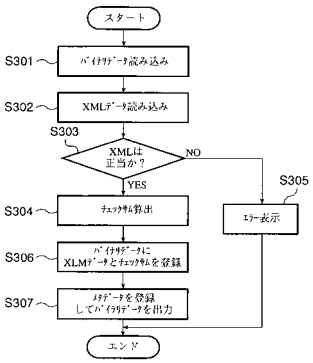
【図４】第２の実施形態による登録されたメタデータの判別及び抽出手順を示すフローチ

ャートである。
【図５】第２の実施形態によるメタデータの判別処理の詳細を説明するフローチャートである。
【図６】メタデータとしてＸＭＬデータが登録されたバイナリデータのデータ構成例を示す図である。
【図７】バイナリデータにメタデータを埋め込んだフォーマットの概観を示す図である。
【図８】バイナリデータとメタデータをデータベースで管理する方法を概念的に示した図である。

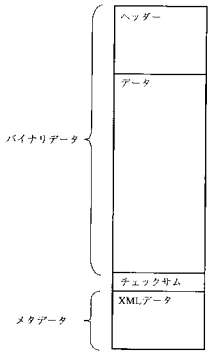
【図１】



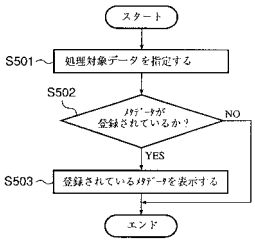
【図２】



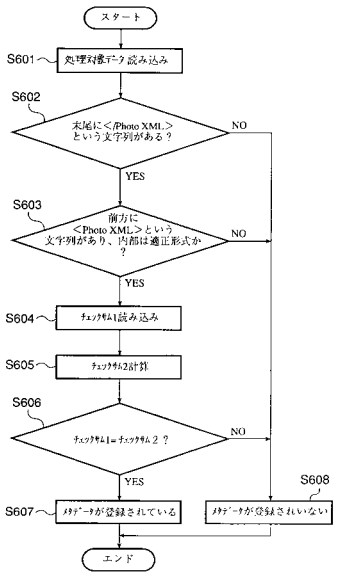
【図３】


ヘッダー
データ
バイナリデータ
チェックサム
XMLデータ
メタデータ

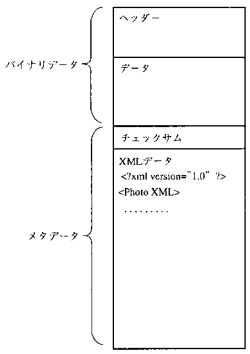
【図４】


スタート
S501
処理対象データを指定する
S502
メタデータが登録されているか？
NO
YES
S503
登録されているメタデータを表示する
エンド

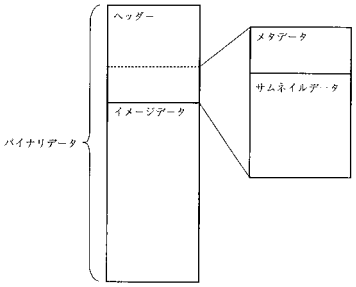
【図５】


スタート
S601
処理対象データ読み込み
S602
末尾に</Photo XML>という文字列がある？
NO
YES
S603
前方に<Photo XML>という文字列があり、内部は適正形式か？
NO
YES
S604
チェックサム1読み込み
S605
チェックサム2計算
S606
チェックサム1＝チェックサム2？
NO
YES
S607
メタデータが登録されている
S608
メタデータが登録されいない
エンド

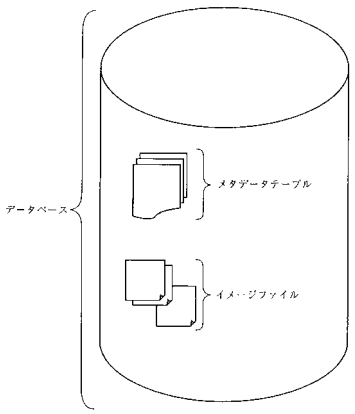
【図６】


ヘッダー
データ
バイナリデータ
チェックサム
XMLデータ
<?xml version="1.0" ?>
<Photo XML>
.....
メタデータ

【図７】


ヘッダー
メタデータ
サムネイルデータ
イメージデータ
バイナリデータ

【図８】


メタデータテーブル
イメージファイル
データベース

フロントページの続き

(72)発明者　山本　邦浩
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内
(72)発明者　草間　澄
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

審査官　高瀬　勤

(56)参考文献　特開平０７－２７４１０８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/00
G06F 17/30
H04N 5/92
JSTPlus(JDream2)