

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2013-131107

(P2013-131107A)

(43) 公開日 平成25年7月4日(2013.7.4)

(51) Int.Cl.

G06F 12/00 (2006.01)

F I

G06F 12/00 520P

G06F 12/00 546A

テーマコード (参考)

審査請求 未請求 請求項の数 8 O L (全 21 頁)

(21) 出願番号 特願2011-281077 (P2011-281077)
 (22) 出願日 平成23年12月22日 (2011.12.22)

(71) 出願人 390002761
 キヤノンマーケティングジャパン株式会社
 東京都港区港南2丁目16番6号
 (71) 出願人 312000206
 キヤノンMJアイティグループホールディングス株式会社
 東京都品川区東品川2丁目4番11号
 (71) 出願人 301015956
 キヤノンソフトウェア株式会社
 東京都品川区東品川二丁目4番11号
 (74) 代理人 100126103
 弁理士 伊藤 幹郎
 (72) 発明者 齊藤 竜一
 東京都品川区東品川2丁目4番11号 キヤノンソフトウェア株式会社

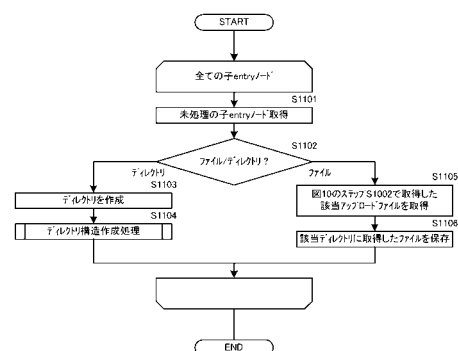
(54) 【発明の名称】 情報処理システム、情報処理装置、情報処理方法、及びコンピュータプログラム

(57) 【要約】 (修正有)

【課題】ウェブベースの文書管理サービスにおいて、ディレクトリの階層構造を指定した形で、ファイルの登録を行うことを可能とする。

【解決手段】クライアント装置に表示される画面への操作により、文書管理システムにアップロードするファイルの指定と、その階層構造を受け付ける。そして、アップロード指示を受け付けると、指定されたファイルと階層構造情報を含むHTTPリクエストを作成し、文書管理サービスを提供する文書管理サーバに対して送信する。HTTPリクエストを受信した文書管理サーバは、受信したHTTPリクエストを解析し、ディレクトリを構築するとともに、アップロードされたファイルを指定されたディレクトリに保存する。

【選択図】図11



【特許請求の範囲】**【請求項 1】**

ウェブサービスにより文書管理サービスを提供するサーバ装置と、当該文書管理サービスを利用するクライアント装置とを備えて構成される情報処理システムであって、

前記サーバ装置は、

前記クライアント装置から、文書登録要求を受け付けた後に、要求を行ったクライアント装置に対して登録画面を表示するための画面情報を送信する第 1 の送信手段と、

を備え、

前記クライアント装置は、

前記サーバ装置から送信される画面情報を受信する受信手段と、

10

前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示手段と、

前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、

前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、

前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、HTTPリクエストを生成する生成手段と、

前記生成手段で生成されたHTTPリクエストを、前記サーバ装置に送信する第 2 の送信手段と、

20

を備えることを特徴とする情報処理システム。

【請求項 2】

前記登録画面には、アップロードファイルの指定を受け付ける領域が設定されており、

前記受付手段は、前記領域に対するファイルのドラッグ&ドロップ指示を受け付けることで、アップロードファイルの指定を受け付けること

を特徴とする請求項 1 に記載の情報処理システム。

【請求項 3】

前記領域は、記憶装置に設定されているディレクトリに対応付けられており、

前記サーバ装置は、

前記HTTPリクエストの生成に用いられたアップロードファイルを、前記ディレクトリに保存すること保存手段

30

をさらに備えることを特徴とする請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

ウェブサービスにより文書管理サービスを提供するサーバ装置と通信可能に接続される情報処理装置であって、

前記サーバ装置に対して、ファイルのアップロード要求を行う要求手段と、

前記要求手段に応じて前記サーバ装置より送信される、アップロードファイルの指定に用いる登録画面を表示するための画面情報を受信する受信手段と、

前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示手段と、

40

前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、

前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、

前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、HTTPリクエストを生成する生成手段と、

前記生成手段で生成されたHTTPリクエストを、前記サーバ装置に送信する送信手段と、

を備えることを特徴とする情報処理装置。

【請求項 5】

50

ウェブサービスにより文書管理サービスを提供するサーバ装置と、当該文書管理サービスを利用するクライアント装置とを備えて構成される情報処理システムによって行われる情報処理方法であって、

前記サーバ装置の第1の送信手段が、前記クライアント装置から、文書登録要求を受け付けた後に、要求を行ったクライアント装置に対して登録画面を表示するための画面情報を送信する第1の送信工程と、

前記クライアント装置の受信手段が、前記サーバ装置から送信される画面情報を受信する受信工程と、

前記クライアント装置の表示手段が、前記受信工程で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示工程と、

前記クライアント装置の受付手段が、前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付工程と、

前記クライアント装置の作成手段が、前記受付工程で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成工程と、

前記クライアント装置の生成手段が、前記受付工程で指定を受け付けたアップロードファイル、及び、前記作成工程で作成した階層構造データを用いて、HTTPリクエストを生成する生成工程と、

前記クライアント装置の第2の送信手段が、前記生成工程で生成されたHTTPリクエストを、前記サーバ装置に送信する第2の送信工程と、

を備えることを特徴とする情報処理方法。

【請求項6】

ウェブサービスにより文書管理サービスを提供するサーバ装置と通信可能に接続される情報処理装置によって行われる情報処理方法であって、

前記サーバ装置に対して、ファイルのアップロード要求を行う要求工程と、

前記要求工程でのファイルのアップロード要求に応じて前記サーバ装置より送信される、アップロードファイルの指定に用いる登録画面を表示するための画面情報を受信する受信工程と、

前記受信工程で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示工程と、

前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付工程と、

前記受付工程で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成工程と、

前記受付工程で指定を受け付けたアップロードファイル、及び、前記作成工程で作成した階層構造データを用いて、HTTPリクエストを生成する生成工程と、

前記生成工程で生成されたHTTPリクエストを、前記サーバ装置に送信する送信工程と、

を備えることを特徴とする情報処理方法。

【請求項7】

ウェブサービスにより文書管理サービスを提供するサーバ装置と、当該文書管理サービスを利用するクライアント装置とを備えて構成される情報処理システムで実行可能なコンピュータプログラムであって、

前記サーバ装置を、

前記クライアント装置から、文書登録要求を受け付けた後に、要求を行ったクライアント装置に対して登録画面を表示するための画面情報を送信する第1の送信手段と、

として機能させ、

前記クライアント装置を、

前記サーバ装置から送信される画面情報を受信する受信手段と、

前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて

10

20

30

40

50

登録画面を表示する表示手段と、

前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、

前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、

前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、ＨＴＴＰリクエストを生成する生成手段と、

前記生成手段で生成されたＨＴＴＰリクエストを、前記サーバ装置に送信する第２の送信手段

として機能させることを特徴とするコンピュータプログラム。

10

【請求項 8】

ウェブサービスにより文書管理サービスを提供するサーバ装置と通信可能に接続される情報処理装置を、

前記サーバ装置に対して、ファイルのアップロード要求を行う要求手段と、

前記要求手段に応じて前記サーバ装置より送信される、アップロードファイルの指定に用いる登録画面を表示するための画面情報を受信する受信手段と、

前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示手段と、

前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、

20

前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、

前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、ＨＴＴＰリクエストを生成する生成手段と、

前記生成手段で生成されたＨＴＴＰリクエストを、前記サーバ装置に送信する送信手段として機能させることを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、文書管理システムに関し、特にウェブベースの文書管理システムにおけるファイルのアップロードに関する。

30

【背景技術】

【0002】

従来、複数のユーザが共有する文書データを管理するために、文書管理サーバを設置し、共有文書データをその文書管理サーバで管理することが行われている。例えば、文書管理サーバに複数のユーザがアクセス可能な共有ディレクトリを作成し、そのディレクトリに複数のユーザが共有すべき文書データを保存することにより、複数のユーザ間で文書データを共有することが可能となる。

【0003】

その際には、それぞれのユーザに対して、当該共有ディレクトリに保存されているファイルに対して、どのような処理を行うことが出来るかというアクセス権を設定することが可能である。例えば、文書データの新規登録、既に登録されている文書データの編集や削除を行うことが出来るか否かという設定が可能である。

40

【0004】

また、最近では、数多くのユーザで文書データを共有したり、また、複雑なアクセス権を設定したりするために、文書管理サービスや、コンテンツ管理サービスを提供するケースが増えてきている。例えば、特許文献１に示すように、ウェブベースで文書管理サービスを提供することが行われるようになってきている。

【先行技術文献】

【特許文献】

50

【 0 0 0 5 】

【特許文献 1】特開 2 0 0 8 - 4 6 7 7 6 号公報

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 6 】

通常、ユーザが使用するコンピュータで文書管理を行う際には、階層構造を持つディレクトリを作成し、そのディレクトリに文書を分類して管理している。ウェブベースでの文書管理サービスでも、このように階層構造ディレクトリを設定し、文書データを分類して管理することを可能にしていることが多い。

【 0 0 0 7 】

10

しかし、通常、この階層構造を有するディレクトリに新規に文書ファイルを登録する場合、ディレクトリを指定し、どのディレクトリに登録するファイルを設定するという作業を、それぞれのディレクトリごとに行わなくてはならず、文書データの登録が効率的ではなかった。

【 0 0 0 8 】

そこで、本発明は、ウェブベースの文書管理サービスにおいて、階層構造を有するディレクトリに効率的に文書データを登録する可能とする技術を提供することを目的とする。

【課題を解決するための手段】

【 0 0 0 9 】

20

上記した目的を達成するために、本発明の情報処理システムは、ウェブサービスにより文書管理サービスを提供するサーバ装置と、当該文書管理サービスを利用するクライアント装置とを備えて構成される情報処理システムであって、前記サーバ装置は、前記クライアント装置から、文書登録要求を受け付けた後に、要求を行ったクライアント装置に対して登録画面を表示するための画面情報を送信する第 1 の送信手段と、を備え、前記クライアント装置は、前記サーバ装置から送信される画面情報を受信する受信手段と、前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示手段と、前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、H T T P リクエストを生成する生成手段と、前記生成手段で生成された H T T P リクエストを、前記サーバ装置に送信する第 2 の送信手段と、を備えることを特徴とする。

30

【 0 0 1 0 】

上記した目的を達成するために、本発明の情報処理装置は、ウェブサービスにより文書管理サービスを提供するサーバ装置と通信可能に接続される情報処理装置であって、前記サーバ装置に対して、ファイルのアップロード要求を行う要求手段と、前記要求手段に応じて前記サーバ装置より送信される、アップロードファイルの指定に用いる登録画面を表示するための画面情報を受信する受信手段と、前記受信手段で受信した画面情報に基づき、ウェブブラウザアプリケーションを用いて登録画面を表示する表示手段と、前記登録画面に対する操作指示により、サーバ装置へのアップロードファイルの指定、及び、ディレクトリ階層構造の入力を受け付ける受付手段と、前記受付手段で入力を受け付けたアップロードファイル及びディレクトリ階層構造に従って、階層構造データを作成する作成手段と、前記受付手段で指定を受け付けたアップロードファイル、及び、前記作成手段で作成した階層構造データを用いて、H T T P リクエストを生成する生成手段と、前記生成手段で生成された H T T P リクエストを、前記サーバ装置に送信する送信手段と、を備えることを特徴とする。

40

【発明の効果】

【 0 0 1 1 】

本発明によれば、ウェブベースでの文書管理サービスにおいて、複数の異なるディレク

50

トリにファイルを登録する作業を効率的に行わせることが可能となる。

【図面の簡単な説明】

【0012】

【図1】本発明の実施の形態における情報処理システムのシステム構成の一例を示すシステム構成図である。

【図2】図1の文書管理サーバ101、クライアント装置102に適用可能な情報処理装置のハードウェア構成の一例を示す図である。

【図3】クライアント装置102によって行われるアップロードファイル設定処理の概要を示すフローチャートである。

【図4】図3のステップS304のディレクトリ/ファイルのドラッグ&ドロップ受付処理の詳細を示すフローチャートである。

【図5】図4のステップS410のツリー構造作成処理の詳細を示すフローチャートである。

【図6】図3のステップS306のディレクトリ追加処理の詳細を示すフローチャートである。

【図7】図3のステップS308の矩形エリアクリア処理の詳細を示すフローチャートである。

【図8】図3のステップS310の全矩形エリアクリア処理の詳細を示すフローチャートである。

【図9】図3のステップS312のファイルアップロード処理の詳細を示すフローチャートである。

【図10】文書管理サーバ101のCPU201によって行われるファイル受信処理を示すフローチャートである。

【図11】図10のステップS1004のディレクトリ構造作成処理の詳細を示すフローチャートである

【図12】矩形情報ファイル1200の構成の一例を示す図である。

【図13】ツリー構造ファイル1300の構成の一例を示す図である。

【図14】アップロードファイルリスト1400の構成の一例を示す図である。

【図15】ファイルアップロード画面1500の構成の一例を示す図である。

【図16】矩形領域1501にディレクトリアイコンがドラッグ&ドロップされた場合のファイルアップロード画面1500の表示制御の例を示す図である。

【発明を実施するための形態】

【0013】

以下、図面を参照して、本発明の実施の形態の一例について説明する。

【0014】

図1は、本発明の実施の形態における情報処理システムのシステム構成の一例を示す図である。図1に示す通り、本発明の情報処理システムは、文書管理サーバ101、複数台のクライアント装置102、LAN103を備えて構成されている。

【0015】

文書管理サーバ101は、クライアント装置102に対して、文書ファイルの管理を行うための文書管理サービスをウェブサービスとして提供するサーバ装置である。

【0016】

クライアント装置102は、文書管理サーバ101が提供する文書管理サービスを利用するユーザが使用する装置であり、ブラウザアプリケーションを用いて文書管理サーバ101にアクセスすることにより、文書管理サービスの提供を受ける。

【0017】

LAN103は、上記した各装置を通信可能に接続するためのネットワークであり、その接続形態は有線/無線を問わない。以上が、本発明の実施形態における情報処理システムのシステム構成の一例の説明である。

【0018】

10

20

30

40

50

次に、図 2 を参照して、図 1 の文書管理サーバ 101、及び、クライアント装置 102 に適用可能な情報処理装置のハードウェア構成の一例について説明する。

【0019】

図 2 において、201 は CPU で、システムバス 204 に接続される各デバイスやコントローラを統括的に制御する。また、ROM 203 あるいは外部メモリ 211 には、CPU 201 の制御プログラムである BIOS (Basic Input / Output System) やオペレーティングシステムプログラム (以下、OS) や、文書管理サーバ 101、クライアント装置 102 がそれぞれ実行する機能を実現するために必要な後述する各種プログラム等が記憶されている。

【0020】

202 は RAM で、CPU 201 の主メモリ、ワークエリア等として機能する。CPU 201 は、処理の実行に際して必要なプログラム等を ROM 203 あるいは外部メモリ 211 から RAM 202 にロードして、該ロードしたプログラムを実行することで各種動作を実現するものである。

【0021】

また、205 は入力コントローラで、キーボードやポインティングデバイス等の入力装置 209 からの入力を制御する。206 はビデオコントローラで、ディスプレイ装置 210 等の表示器への表示を制御する。これらは必要に応じて操作者が使用するものである。

【0022】

207 はメモリコントローラで、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、各種データ等を記憶するハードディスク (HD) や、フレキシブルディスク (FD)、或いは PCMCIA カードスロットにアダプタを介して接続されるコンパクトフラッシュ (登録商標) メモリ等の外部メモリ 211 へのアクセスを制御する。

【0023】

208 は通信 I/F (インタフェース) コントローラで、ネットワーク (例えば、図 1 に示した LAN 103) を介して外部機器と接続・通信するものであり、ネットワークでの通信制御処理を実行する。例えば、TCP/IP を用いた通信等が可能である。

【0024】

なお、CPU 201 は、例えば RAM 202 内の表示情報用領域へアウトラインフォントの展開 (ラスターライズ) 処理を実行することにより、ディスプレイ装置 210 上での表示を可能としている。また、CPU 201 は、ディスプレイ装置 210 上の不図示のマウスカーソル等でのユーザ指示を可能とする。

【0025】

本発明を実現するための後述するフローチャートに示す各ステップの処理は、コンピュータで読み取り実行可能なプログラムにより実行され、そのプログラムは外部メモリ 211 に記録されている。そして、必要に応じて RAM 202 にロードされることにより CPU 201 によって実行されるものである。さらに、上記プログラムの実行時に用いられる定義ファイル及び各種情報テーブル等も、外部メモリ 211 に格納されており、これらについての詳細な説明も後述する。

【0026】

次に、図 3 を参照して、図 1 のクライアント装置 102 の CPU 201 によって行われるアップロードファイル設定処理について説明する。本図に示す処理をクライアント装置 102 の CPU 201 に行わせるためのプログラムは、外部メモリ 211 等に記憶されており、本図に示す処理を実行するに際し、CPU 201 は当該プログラムを RAM 202 にロードし、ロードしたプログラムによる制御に従って本処理を実行する。

【0027】

まず、クライアント装置 102 は、文書管理サーバ 101 に対して、ファイルアップロード要求を行うことにより、その要求に応じて文書管理サーバ 101 より送信される画面情報に従って、ファイルアップロード画面 1500 (図 15) を表示する (ステップ S 3

10

20

30

40

50

01)。

【0028】

ここで、図15を参照して、ファイルアップロード画面1500の構成について説明する。図15は、クライアント装置102のディスプレイ装置210に表示されるファイルアップロード画面1500の構成の一例を示す図である。

【0029】

図15に示すように、ファイルアップロード画面1500は、矩形エリア1501、クリアボタン1502、ディレクトリ名入力欄1503、ディレクトリ追加ボタン1504、全クリアボタン1505、アップロードボタン1506、キャンセルボタン1507等を備えて構成されている。

10

【0030】

矩形エリア1501は、文書管理サーバ101にアップロードするファイルを指定するために用いられるエリアである。このエリアに、文書管理サーバ101にアップロードするファイルのアイコンをドラッグ&ドロップすることにより、アップロードするファイルの指定が可能である。

【0031】

また、この矩形エリア1501で、ディレクトリのツリー構造を構築可能であり、アップロードするファイルをこのツリー構造のディレクトリにドラッグ&ドロップすることにより、ファイルを、階層構造をもった形でアップロードすることが可能となる。

【0032】

尚、ファイルアイコンだけでなく、ディレクトリアイコンのドラッグ&ドロップを受け付けることも可能である。その例については、図16を参照して後述することにする。本発明では、文書管理サーバ101上のディレクトリに紐づいた複数の矩形エリア1501を表示し、それぞれの矩形エリアに対して、アップロードするファイルを、階層構造をもった形で設定することが可能である。これにより、異なるディレクトリに対するファイルのアップロードを一度の操作で行うことが可能となる。

20

【0033】

クリアボタン1502は、矩形エリア1501に構築されたディレクトリ構造(ツリー構造)や、アップロードファイルの設定をクリアするために用いられるボタンである。

【0034】

ディレクトリ名入力欄1503は、矩形エリア1501に新たに追加するディレクトリの名称の入力を受け付ける入力欄である。ディレクトリ追加ボタン1504は、矩形エリア1501に新規にディレクトリを追加するために用いられるボタンである。矩形エリア1501のいずれかのディレクトリが選択されている状態で、このディレクトリ追加ボタン1504が押下されると、選択状態のディレクトリの直下に、ディレクトリ名入力欄1503に入力されたディレクトリ名のディレクトリが作成される。

30

【0035】

全クリアボタン1505は、全ての矩形エリア1501に構築されたディレクトリ構造(ツリー構造)や、アップロードファイルの設定をクリアするために用いられるボタンである。

40

【0036】

アップロードボタン1506は、それぞれの矩形エリア1501に構築されたディレクトリ構造を保った形で、そのディレクトリに設定されたファイルを文書管理サーバ101にアップロードする指示を行うために用いられるボタンである。

【0037】

キャンセルボタン1507は、ファイルをアップロードせずに本図を用いたファイルアップロード処理を終了するために用いられるボタンである。以上が、図15のファイルアップロード画面1500の構成の説明である。

【0038】

尚、文書管理サーバ101のCPU201は、図12に示す矩形情報ファイル1200

50

を用いて、ファイルアップロード画面 1500 を表示させるための画像情報を作成する。この矩形情報ファイル 1200 は、例えばユーザごとに登録されており、ログイン処理により特定されたユーザの矩形情報ファイル 1200 を用いて、ファイルアップロード画面 1500 を作成する。図 12 の場合には、「矩形エリア 1」という名称、「矩形エリア 2」という名称、及び、「矩形エリア 3」という名称の 3 つの矩形エリアがファイルアップロード画面 1500 に作成されることになる（つまり、図 15 に示すイメージ）。

【0039】

また、それぞれの矩形エリア 1501 はそれぞれ、文書管理サーバ 101 上のディレクトリと対応付けられており（以下、対応ディレクトリという）、それぞれの矩形エリア 1501 に設定されたファイルは、対応ディレクトリに保存されることになる。また、矩形エリア 1501 にディレクトリ構造化構築されている場合には、対応ディレクトリの直下にそのディレクトリ構造が作成されることになる。

【0040】

図 3 の説明に戻る。クライアント装置 102 の CPU 201 は、ステップ S301 において、ファイルアップロード画面 1500 をディスプレイ装置 210 に表示後、そのファイルアップロード画面 1500 を介して、ユーザからの各種操作を受け付ける（ステップ S302）。

【0041】

そして、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作が、ディレクトリアイコンやファイルアイコンの矩形エリア 1501 へのドラッグ&ドロップ処理であると判定した場合には（ステップ S303 で YES）、処理をステップ S304 に進め、ディレクトリ/ファイルのドラッグ&ドロップ受付処理を行う。この処理の詳細については、図 4 を参照して後述する。

【0042】

また、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作が、ディレクトリ追加ボタン 1504 に対する押下指示であると判定した場合には（ステップ S305 で YES）、処理をステップ S306 に進め、ディレクトリ追加処理を行う。この処理の詳細については、図 6 を参照して後述する。

【0043】

また、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作が、クリアボタン 1502 に対する押下指示であると判定した場合には（ステップ S307 で YES）、処理をステップ S308 に進め、矩形エリアクリア処理を行う。この処理の詳細については、図 7 を参照して後述する。

【0044】

また、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作が全クリアボタン 1505 に対する押下指示であると判定した場合には（ステップ S309 で YES）、処理をステップ S310 に進め、全矩形エリアクリア処理を行う。この処理の詳細については、図 8 を参照して後述する。

【0045】

また、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作がファイルアップロードボタン 1506 に対する押下指示であると判定した場合には（ステップ S311 で YES）、処理をステップ S312 に進め、ファイルアップロード処理を行う。この処理の詳細については、図 9 を参照して説明する。

【0046】

また、クライアント装置 102 の CPU 201 は、ステップ S302 で受け付けた操作がキャンセルボタン 1507 に対する押下指示であると判定した場合には（ステップ S313 で YES）、本図に示す処理を終了する。なお、上記のいずれでもない操作指示を受け付けたと判定した場合には（この場合、ステップ S313 で YES と判定）、処理をステップ S314 に進め、指示されたその他の処理を実行することになる。例えば、ディレクトリ名入力欄 1503 へのディレクトリ名入力操作を受け付けた場合には、その入力さ

10

20

30

40

50

れた文字列を、ディレクトリ名入力欄 1 5 0 3 に表示する処理を行う。

【 0 0 4 7 】

以上が、クライアント装置 1 0 2 の CPU 2 0 1 によって行われるアップロードファイル設定処理の概要の説明である。

【 0 0 4 8 】

次に、図 4 を参照して、図 3 のステップ S 3 0 4 のディレクトリ / ファイルドラッグ & ドロップ処理の詳細について説明する。

【 0 0 4 9 】

まず、クライアント装置 1 0 2 の CPU 2 0 1 は、ディレクトリアイコンや、ファイルアイコンがドロップされた箇所（いずれの矩形エリア 1 5 0 1 のいずれのディレクトリ）を特定する（ステップ S 4 0 1）。この時、複数のファイルアイコン、複数のディレクトリアイコン、ファイルアイコンとディレクトリアイコンに対するドラッグ & ドロップ処理を受け付けることが可能である。

【 0 0 5 0 】

その後、クライアント装置 1 0 2 の CPU 2 0 1 は、ドラッグ & ドロップ処理を受け付けた全てのアイコンに対して、ステップ S 4 0 2 以降の処理を行う。

【 0 0 5 1 】

まず、ステップ S 4 0 2 において、ドラッグ & ドロップ処理を受け付けたアイコンに対応するディレクトリ、若しくはファイルを取得する。例えば、HTML5 のファイル API を用いて、その情報を取得することになる。そして、ステップ S 4 0 2 で取得したものが、ディレクトリであるか、それともファイルであるかを判定する（ステップ S 4 0 3）。

【 0 0 5 2 】

ステップ S 4 0 3 の判定処理で、ステップ S 4 0 2 で取得したものがディレクトリであると判定した場合には、処理をステップ S 4 0 4 に進め、ステップ S 4 0 2 で取得したディレクトリに保存されている（サブディレクトリも含む）ファイルの数を取得する。尚、一度に所定数（制限値）以上のファイルをアップロードすると文書管理サーバ 1 0 1 の負荷が高くなるので、本発明では、一度にアップロード可能なファイル数に制限を設けている。尚、ファイル数ではなく、ファイルサイズ等で制限を設けるようにしても構わない。

【 0 0 5 3 】

その後、ステップ S 4 0 5 の判定処理で、ディレクトリに保存されているファイルと、既にアップロード対象として設定されているファイルの合計が所定数（制限値）以上である（YES）と判定した場合には、処理をステップ S 4 0 7 に進めエラー表示を行う。一方、所定数（制限値）未満である（NO）と判定した場合には、矩形エリア 1 5 0 1 中のステップ S 4 0 1 で特定したディレクトリの直下にディレクトリアイコンを追加表示し（ステップ S 4 0 6）、処理をステップ S 4 1 0 に進める。

【 0 0 5 4 】

ここで、図 1 6 を参照して、図 4 の S 4 0 6 で行われる処理の概要について説明する。

【 0 0 5 5 】

図 1 6 の（１）に示すように、デスクトップ上の「文書ディレクトリ」という名称のディレクトリアイコン 1 6 0 1 を、矩形エリア 1 5 0 1 中の「ルート」という名称のディレクトリアイコン 1 6 0 2 に対してドラッグ & ドロップすると、図 1 6 の（２）に示すように、矩形エリア 1 5 0 1 中のディレクトリアイコン 1 6 0 2 の直下に、新たに「文書ディレクトリ」という名称のディレクトリアイコン 1 6 0 3 が表示されることになる。

【 0 0 5 6 】

これにより、デスクトップ上のディレクトリアイコン 1 6 0 1 に保存されているファイル（サブディレクトリに保存されているファイルも含む）をアップロード対象のファイルとして設定することが可能である。また、図 1 6 の（２）の状態、図 1 6 では不図示のアップロードボタン 1 5 0 6 に対する押下指示を受け付けると、この矩形エリア 1 5 0 1 に対応する文書管理サーバ 1 0 1 上のディレクトリの直下に、「ルート」という名称のデ

10

20

30

40

50

ィレクトリが、「ルート」ディレクトリの直下に「文書ディレクトリ」という名称のディレクトリが作成され、デスクトップ上の「文書ディレクトリ」に保存されているファイルが、文書管理サーバ 101 上に作成された「文書ディレクトリ」にアップロードされることになる。

【0057】

図4の説明に戻る。ステップS403の判定処理で、ステップS402で取得したものがファイルであるとクライアント装置102のCPUが判定した場合には、処理をステップS408に進め、ステップS402で取得したファイルのMIMEタイプ(ファイルタイプ)を取得する。そして、取得したMIMEタイプに合わせたアイコンを、ステップS401で特定したディレクトリの直下に追加表示する(ステップS409)。その後、処理をステップS410に進める。

10

【0058】

ステップS406、ステップS409の処理が終了後、クライアント装置102のCPU201は処理をステップS410に進め、ツリー構造作成処理を行う。この処理の詳細については、図5を参照して説明する。

【0059】

そして、上記ステップS402からステップS410の処理を、ドラッグ&ドロップ処理を受け付けた全てのファイルアイコン、ディレクトリアイコンについて行う。以上が、図3のステップS304のディレクトリ/ファイルのドラッグ&ドロップ受付処理の詳細の説明である。

20

【0060】

次に、図13を参照して、矩形エリア1501に作成されるディレクトリ構造を管理するツリー構造ファイル1300の構成について説明する。ツリー構造ファイル1300はXML形式のファイルである。

【0061】

ツリー構造ファイル1300には、最上位のノードとして、「entries」ノードが設定されている。そして、その下に、アップロードするファイルやディレクトリ構造を示す「entry」ノードが作成されることになる。

【0062】

「entries」ノードの直下の「entry」ノードは矩形エリア1501を示すノードである。そして、その下には、矩形エリア1501に構築されたディレクトリ構造やアップロード対象となるファイルを示す「entry」ノードが作成されることになる。

30

【0063】

尚、ファイルを示すノードのtypeには「file」が、ディレクトリを示すノードのtypeには「folder」が登録されるので、その「entry」ノードがファイルを示すものであるか、それともディレクトリを示すものであるかを識別することが出来る。以上が、ツリー構造ファイル1300の説明である。

【0064】

次に、図5を参照して、図4のステップS410のツリー構造作成処理の詳細について説明する。この処理では、クライアント装置102のCPU201は、矩形エリア1501に対するディレクトリアイコンやファイルアイコンのドラッグ&ドロップ処理により、矩形エリア1501内のディレクトリ構造に変更が加えられた場合に、当該変更に合わせて、図13のツリー構造ファイル1300や図14のアップロードファイルリストを更新することになる。

40

【0065】

まず、ステップS501において、CPU201は、entryノードを直下に追加するノード、つまりは、ディレクトリアイコン/ファイルアイコンのドラッグ&ドロップを受け付けたディレクトリに対応するentryノードを特定する。そして、追加対象のディレクトリまたはファイルを取得する(ステップS502)。

50

【0066】

そして、ステップS502で取得したファイルまたはディレクトリに対応するentryノードを新たにツリー構造ファイル1300に追加する(ステップS503)。その後、ステップS502で取得したものが、ファイルであるかそれともディレクトリであるかを判定する(ステップS504)。

【0067】

ステップS504の判定処理で、ステップS502で取得したものがディレクトリであると判定した場合には、クライアント装置102のCPU201は処理をステップS505に進め、ステップS503で新たに追加したentryノードのnameにドラッグ&ドロップ処理がされたディレクトリのディレクトリ名を設定する。また、typeには「folder」を設定する(ステップS506)。その後、ドラッグ&ドロップ処理がされたディレクトリに含まれるサブディレクトリやファイルの登録をするために、ツリー構造作成処理を再帰的に行うことになる(ステップS507)。

【0068】

一方、ステップS504の判定処理で、ステップS502で取得したものがファイルであると判定した場合には、クライアント装置102のCPU201は処理をステップS508に進め、ステップS503で新たに追加したentryノードのnameにドラッグ&ドロップ処理がされたファイルのファイル名を設定する。また、Contentにファイルタイプを、typeに「file」をそれぞれ設定することになる(ステップS509、S510)。

【0069】

その後、クライアント装置102のCPU201は、処理をステップS511に進め、ステップS502で取得したファイルのパス情報を、当該ファイルを管理するentryノードの「id」と対応付けて、図14に示すアップロードファイルリスト1400に登録する。以上が、図4のステップS410のツリー構造作成処理の詳細な説明である。

【0070】

次に、図14を参照して、アップロードファイルリストのデータ構成の一例について説明する。図14に示す通り、アップロードファイルリスト1400は、データ項目として、ノードID1401、ファイルパス1402等を備えて構成されている。

【0071】

ノード1401は、ツリー構造ファイル1300中のファイルを管理するentryノードのノードIDに対応するかを示す情報が登録されるデータ項目である。ファイルパス1402は、entryノードで特定されるアップロード対象ファイルのファイルパス情報が登録されるデータ項目である。以上が、図14のアップロードファイルリスト1400のデータ構成の説明である。

【0072】

次に、図6を参照して、図3のステップS306のディレクトリ追加処理について説明する。この処理は、ファイルアップロード画面1500のディレクトリ追加ボタン1504に対する押下指示を受け付けた際に、クライアント装置102のCPU201が実行する処理である。

【0073】

まず、クライアント装置102のCPU201は、矩形エリア1500中で選択状態になっているディレクトリを取得する(ステップS601)。尚、選択状態のディレクトリがない場合には、ステップS601ではディレクトリの取得は行われない。

【0074】

そして、その後、ディレクトリ名入力欄1503に入力されたディレクトリ名を取得する(ステップS602)。その後、選択状態のディレクトリの直下に新たにディレクトリアイコンを追加表示する(ステップS603)。この時、ステップS601でディレクトリが取得されなかった場合には、最上位階層に新たにディレクトリアイコンを追加表示することになる。また、このディレクトリの名称として、ステップS602で取得したディ

10

20

30

40

50

レクトリ名を表示する。

【0075】

その後、アップロードファイルリスト1400中の、ステップS601で取得したディレクトリに対応するentryノードの直下に新規にentryノードを作成し(ステップS604)、そのnameにステップS602で取得したディレクトリ名を設定する(ステップS605)。また、typeには「folder」を設定する(ステップS606)。

【0076】

尚、ステップS601でディレクトリを取得しなかった場合には、矩形エリア1501に対応するentryの直下に新規にentryノードを作成することになる。以上が、図3のステップS306のディレクトリ追加処理の説明である。

10

【0077】

次に、図7を参照して、図3のステップS308の矩形エリアクリア処理の詳細について説明する。この処理は、ファイルアップロード画面1500のクリアボタン1502に対する押下指示を受け付けた際に、クライアント装置102のCPU201が実行する処理である。

【0078】

まず、クライアント装置102のCPU201は、アップロードファイルリスト1400中にある、クリア指示を受け付けた矩形エリア1501に対応するentryノードを特定する(ステップS701)。そして、そのentryノードの配下のノード(<entry...>から、対応する</entry>までに含まれるノード)を全て削除する(ステップS702)。

20

【0079】

その後、クリア指示を受け付けた矩形エリア1501中のディレクトリツリー構造を示すアイコンの表示を消去する(ステップS703)。そして、ステップS702で削除したノードのノードIDに対応するアップロードファイルリスト1400中のレコードを削除する(ステップS704)。以上が、図3のステップS308の領域クリア処理の詳細な説明である。

【0080】

次に、図8を参照して、図3のステップS310の全領域クリア処理の詳細について説明する。この処理は、ファイルアップロード画面1500の全クリアボタン1505に対する押下指示を受け付けた際に、クライアント装置102のCPU201が実行する処理である。

30

【0081】

まず、ステップS801において、クライアント装置102のCPU201は、ツリー構造ファイル1300中のentriesノードの配下のノードをすべて削除する。そして、その後、entriesノードの直下に、各矩形エリア1501に対応するentryノードを追加する(ステップS802)。

【0082】

尚、ステップS801の処理で、entriesノードの配下のノードをすべて削除後、ステップS802で、各矩形エリア1501に対応するentryノードを追加するようにせず、それぞれの矩形エリア1501に対応するentryノードの配下のノードを削除するようにしてももちろん構わない。

40

【0083】

その後、全ての矩形エリア1501に中のディレクトリツリー構造を示すアイコンの表示を消去する(ステップS803)。そして、図14のアップロードファイルリスト1400に含まれるレコードをすべて削除する(ステップS804)。以上が図3のステップS310の全矩形エリアクリア処理の詳細な説明である。

【0084】

クライアント装置102は、図15のファイルアップロード画面1500に対する操作

50

指示に応じて、上記した処理を行うことで、文書管理サーバ101にアップロードするファイルとその階層構造情報を作成することになる。その後、ファイルアップロード画面1500中のアップロードボタン1506に対する押下指示を受け付けると、文書管理サーバ101に対してアップロード対象のファイルとその階層構造情報を送信するアップロード処理を行うことになる。

【0085】

次に、ファイルアップロード画面1500中のアップロードボタン1506に対する押下指示を受け付けた場合に、クライアント装置102のCPU201によって行われるアップロード処理(図3のステップS312)の詳細について、図9を参照して説明することにする。

【0086】

クライアント装置102のCPU201は、アップロードファイルリスト1400に登録されている各レコードのファイルパス1402に登録されているファイルパス情報に従って、アップロード対象の外部メモリ211からファイルを取得する(ステップS901)。

【0087】

そして、取得したファイルをもとに、Multipartデータ形式のデータを作成する(ステップS902)。また、図13のツリー構造ファイル1300をもとに、Multipart形式のデータを作成する(ステップS903)。

【0088】

そして、ステップS902、ステップS903で作成したMultipartデータを含むHTTPリクエストを作成し(ステップS904)、作成したHTTPリクエストを文書管理サーバ101に対して送信する。これにより、クライアント装置102は、ブラウザアプリケーションを介して(つまりはHTTPを用いて)、文書管理サーバ101にアップロードするファイルとその階層構造を送信することが可能となる。

【0089】

次に、図10を参照して、図9のステップS905でクライアント装置102から送信されたHTTPリクエストを受信した文書管理サーバ101が行う、ファイル受信処理について説明する。本図に示す処理を文書管理サーバ101のCPU201に実行させるためのプログラムは外部メモリ211等に記憶されており、本処理を実行する際にCPU201は当該プログラムをRAM202にロードし、ロードしたプログラムによる制御に従って本処理を実行する。

【0090】

文書管理サーバ101のCPU201は、クライアント装置102から送信されたHTTPリクエストを受信すると、文書管理サーバ101は、HTTPリクエストに含まれるツリー構造ファイルを取得する(ステップS1001)、また、アップロードされてきたファイルの情報も合わせて取得する(ステップS1002)。

【0091】

その後、ツリー構造ファイルのentriesノードを取得し、そのentriesノード直下のentryノードに対するディレクトリ構造作成処理を行う(ステップS1004:図11のステップS1101で取得される「子ノード」がentriesノード直下のentryノードになる)。この処理の詳細については、図11を参照して説明することにする。そして、以上が、文書管理サーバ101のCPU201によって行われるファイル受信処理の説明である。

【0092】

次に、図11を参照して、図10のステップS1004のディレクトリ構成作成処理の詳細について説明する。

【0093】

CPU201は、注目ノード(初期はentriesノード)の直下の(子ノードである)entryノードに対してステップS1101から1106の処理を行うことで、図

10

20

30

40

50

15の各矩形エリア1501に対応付けられた文書管理サーバ101上のディレクトリに、クライアント装置102で受け付けた階層構造でファイルを保存する処理を行う。

【0094】

まず、ステップS1101では、CPU201は、注目ノードの子ノードであるentryノードのうち、ステップS1102以降の処理を行っていないentryノードを取得する。そして、そのノードがディレクトリを示すノードであるか、それともファイルを示すノードであるかを判定する(ステップS1102)。

【0095】

ステップS1102の判定処理でディレクトリであると判定した場合には、CPU201は処理をステップS1103に進め、注目ノードに対応するディレクトリにステップS1101で取得したノードに対応するディレクトリを作成する。その際のディレクトリ名称は、当該ノードのnameに設定されている名称とする。尚、既にその名称のディレクトリが作成されている場合には、このステップでディレクトリの作成は行わない。

【0096】

そして、その後、当該ノードの直下のentryノードに基づいて、ステップS1103で作成されたディレクトリの下層のディレクトリ構造を作成するために、ステップS1101で取得したノードを注目ノードとして、再帰的にディレクトリ構成作成処理を行う(ステップS1104)。

【0097】

一方、ステップS1102の判定処理でファイルであると判定した場合には、CPU201は、処理をステップS1105に進め、図10のステップS1002で取得したアップロードファイルのうち、当該entryノードに対応する(例えばファイル名で特定)ファイルを取得し(ステップS1105)、そのファイルを注目ノードに対応するディレクトリに保存する(ステップS1106)。その際には、そのファイル名称として、ステップS1101で取得したentryノードのnameに設定された名称が設定される。

【0098】

尚、既にアップロードファイルを登録すべきディレクトリに、その名称のファイルが既に登録されている場合には、ファイルを上書きする/ファイルの名称を変えて保存する、のいずれの方法を採用しても良い。いずれの方法にするかはあらかじめ設定されており、その設定情報に従って、文書管理装置101のCPU201は、ファイルを上書きするか、それともファイルの名称を変えて保存するかを判断することになる。

【0099】

そして、以上の処理を注目ノードの子ノードについて全て行うまで、文書管理サーバ101のCPU201は本図に示す処理を繰り返し行うことになる。以上が図10のステップS1004のディレクトリ構造作成処理の詳細な説明である。

【0100】

本発明では、文書管理サーバ101のCPU201が、クライアント装置102から受信したHTTPリクエストに含まれる、ツリー構造ファイルとアップロードファイル情報を用いて、ディレクトリ構造(階層構造)を構築し、構築したディレクトリ構造にアップロードファイルを保存することが出来る。これにより、ウェブベースの文書管理装置においても、ファイルに階層構造を持たせた形で、ファイルのアップロードを行うことが可能となる。

【0101】

本発明は、例えば、システム、装置、方法、プログラム若しくは記憶媒体等としての実施形態も可能であり、具体的には、複数の機器から構成されるシステムに適用してもよいし、また、1つの機器からなる装置に適用してもよい。

【0102】

なお、本発明は、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を、システム或いは装置に直接、或いは遠隔から供給するものを含む。そして、そのシステム或いは装置のコンピュータが前記供給されたプログラムコードを読み出して実行することに

10

20

30

40

50

よっても達成される場合も本発明に含まれる。

【0103】

したがって、本発明の機能処理をコンピュータで実現するために、前記コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

【0104】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等の形態であってもよい。

【0105】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RWなどがある。また、磁気テープ、不揮発性のメモリカード、ROM、DVD(DVD-ROM、DVD-R)などもある。

【0106】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続する。そして、前記ホームページから本発明のコンピュータプログラムそのもの、若しくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。

【0107】

また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明に含まれるものである。

【0108】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してホームページから暗号化を解く鍵情報をダウンロードさせる。そして、ダウンロードした鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【0109】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される。その他、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部又は全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【0110】

さらに、記録媒体から読み出されたプログラムが、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き込まれる。その後、そのプログラムの指示に基づき、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又は全部を行い、その処理によっても前述した実施形態の機能が実現される。

【0111】

尚、前述した実施形態は、本発明を実施するにあたっての具体化の例を示したものに過ぎず、これらによって本発明の技術的範囲が限定的に解釈されてはならないものである。即ち、本発明はその技術思想、又はその主要な特徴から逸脱することなく、様々な形で実施することができる。

【符号の説明】

【0112】

101 文書管理サーバ

102 クライアント装置

10

20

30

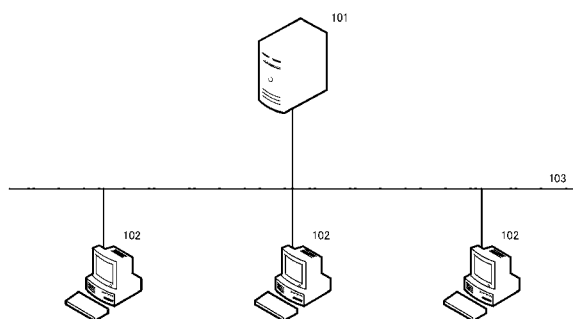
40

50

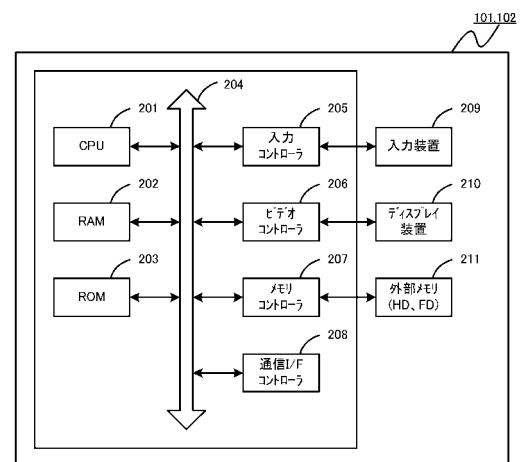
- 1 0 3 L A N
- 2 0 1 C P U
- 2 0 2 R A M
- 2 0 3 R O M
- 2 0 4 システムバス
- 2 0 5 入力コントローラ
- 2 0 6 ビデオコントローラ
- 2 0 7 メモリコントローラ
- 2 0 8 通信 I / F (インタフェース) コントローラ
- 2 0 9 入力装置
- 2 1 0 ディスプレイ装置
- 2 1 1 外部メモリ

10

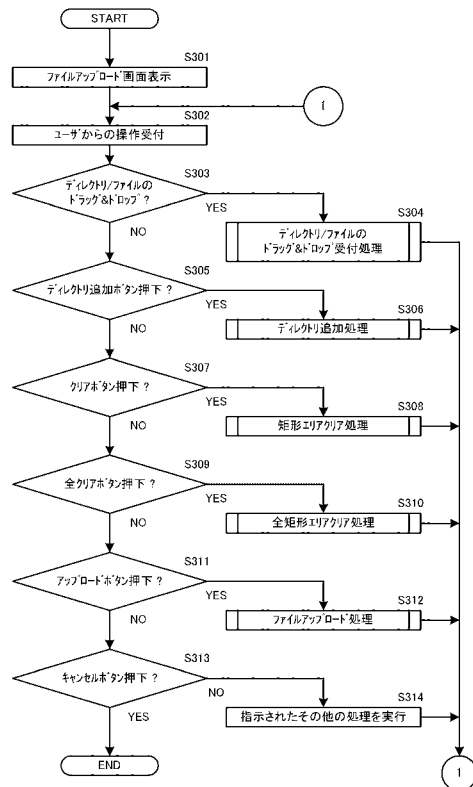
【 図 1 】



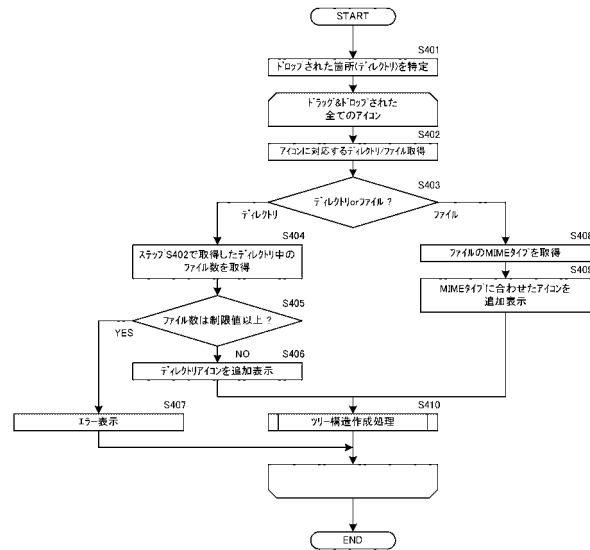
【 図 2 】



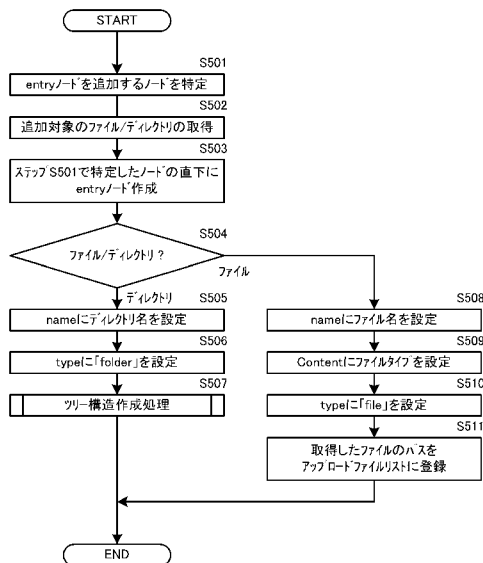
【図 3】



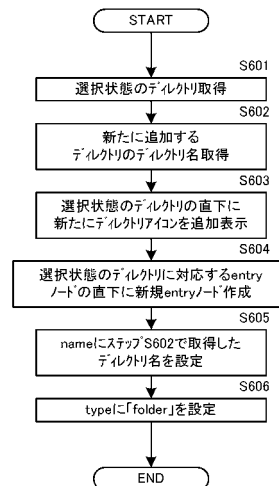
【図 4】



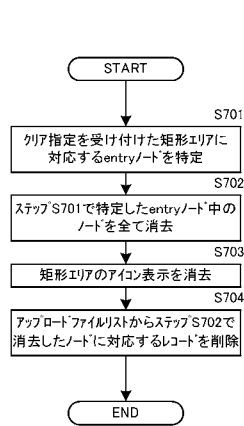
【図 5】



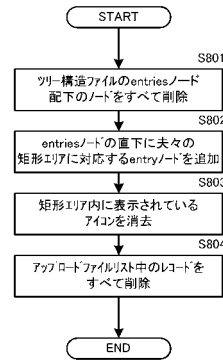
【図 6】



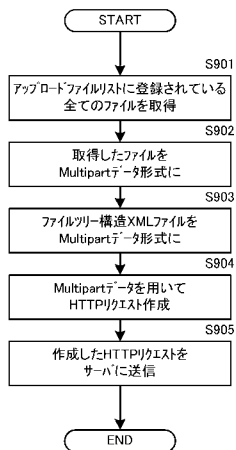
【 図 7 】



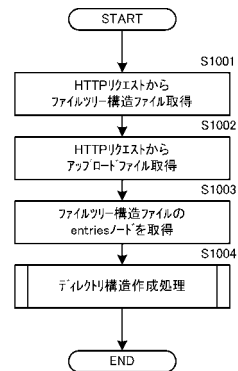
【 図 8 】



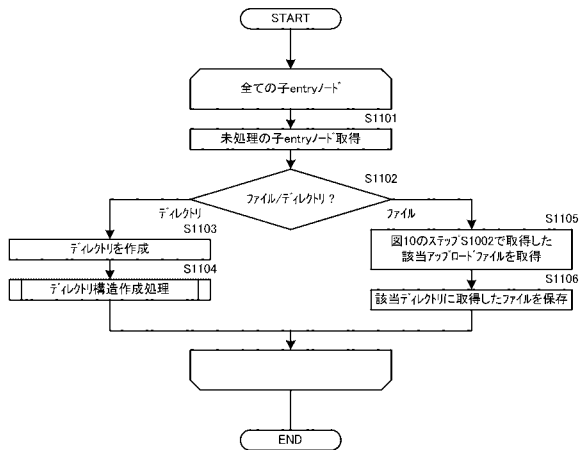
【 図 9 】



【 図 10 】



【図 1 1】



【図 1 2】

1200

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--
<square id="1" name="矩形エリア1" />
<square id="2" name="矩形エリア2" />
<square id="3" name="矩形エリア3" />
-->

```

【図 1 3】

1300

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--
<entry id="1" name="矩形エリア1" type="folder">
  <entry id="2" name="テキスト" type="folder">
    <entry id="4" name="test.txt" content="application/octet-stream" type="file" />
  </entry>
  <entry id="3" name="画像" type="folder">
    <entry id="5" name="Blue hills.jpg" content="image/gif" type="file" />
  </entry>
</entry>
<entry id="6" name="矩形エリア2" type="folder">
  <entry id="7" name="Water lilies.jpg" content="image/jpeg" type="file" />
</entry>
<entry id="8" name="矩形エリア3" type="folder">
  <entry id="9" name="HTML" type="folder">
    <entry id="10" name="test.html" content="application/octet-stream" type="file" />
  </entry>
</entry>
-->

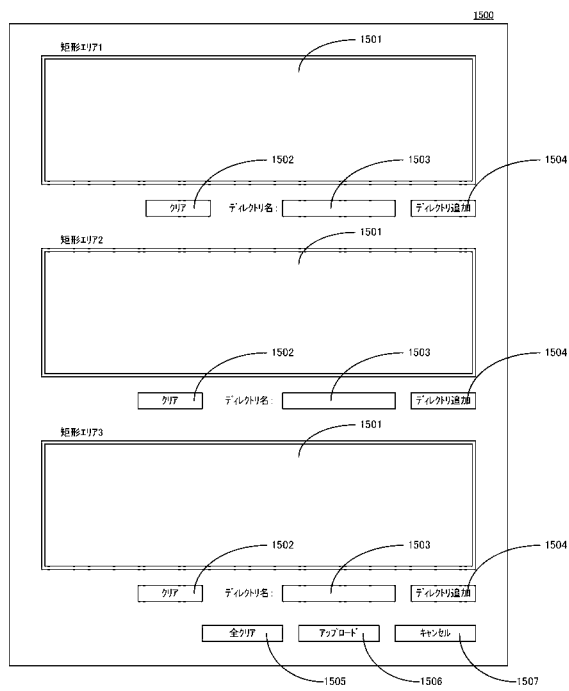
```

【図 1 4】

1400

ノードID	ファイルパス	
4	C:\fileFolder¥1220¥.....
5	D:\imageFolder¥.....
7	D:\imageFolder¥.....
10	D:\htmlFolder¥.....
⋮	⋮	⋮
⋮	⋮	⋮

【図 15】



【図 16】

