

(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2006-164288

(P2006-164288A)

(43) 公開日 平成18年6月22日(2006.6.22)

(51) Int.Cl.

G06F 12/00 (2006.01)

F I

G06F 12/00 520G

テーマコード (参考)

5B082

審査請求 未請求 請求項の数 17 O L (全 25 頁)

(21) 出願番号 特願2005-354120 (P2005-354120)
 (22) 出願日 平成17年12月7日 (2005.12.7)
 (31) 優先権主張番号 11/004, 933
 (32) 優先日 平成16年12月7日 (2004.12.7)
 (33) 優先権主張国 米国 (US)

(特許庁注：以下のものは登録商標)

1. Bluetooth

(71) 出願人 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (72) 発明者 アンドリュー ジー. バイビー
 アメリカ合衆国 98052 ワシントン
 州 レッドモンド ワン マイクロソフト
 ウェイ マイクロソフト コーポレーシ
 ョン内

最終頁に続く

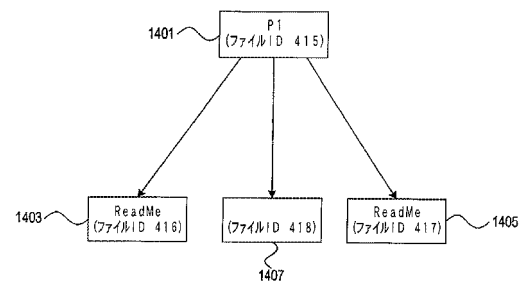
(54) 【発明の名称】 あいまいな名前を許容するコンピュータファイルシステム

(57) 【要約】

【課題】 アイテム名が一意であることを必要とせず、またアイテム名が存在することも必要としないファイルシステムを提供すること。

【解決手段】 アイテムがあいまいな名前を有している場合は、ファイルシステムはあいまいさを無くするプロシージャを実施して、完全に限定されたパスなど、概念的に一意の名前をクライアント（ユーザまたはアプリケーション）に提供する。このファイルシステムは、レガシーアプリケーションやレガシーシステムとの互換性を維持するためのユーザビリティ機能、例えば、アイテムに名前が無いときは合成アイテム名を生成する機能、2つのアイテムが同一の名前を有するときは小さな整数または英数字など、あいまい性除去用の文字を使用してあいまいさを無くする機能などを提供する。

【選択図】 図14



【特許請求の範囲】

【請求項 1】

電子ファイルシステムによって使用されるデータ構造を格納したコンピュータ可読媒体であって、該データ構造は、

前記ファイルシステムによって読み取り可能であって、複数のアイテムのためのファイル情報を格納し、

第 1 のアイテムのための一意の第 1 の識別子を格納する第 1 のデータフィールドと、

前記第 1 のアイテムのユーザが見ることの可能な名前である第 1 のファイル名を格納する第 2 のデータフィールドと、

第 2 のアイテムのための一意の第 2 の識別子を格納する第 3 のデータフィールドと、

前記第 2 のアイテムのユーザが見ることの可能な名前である前記第 1 のファイル名を格納する第 4 のデータフィールドと

を含むファイル情報テーブル、および

前記ファイルシステムによって読み取り可能であって、前記複数のアイテムの関係情報を格納し、

前記第 1 のアイテムの親である第 3 のアイテムの指標を格納する第 1 のデータフィールドと、

前記第 2 のアイテムの親である前記第 3 のアイテムの指標を格納する第 2 のデータフィールドと

を含む編成テーブル

を含むことを特徴とするコンピュータ可読媒体。

【請求項 2】

前記ファイル情報テーブルは、

前記第 1 のファイルのための第 1 のあいまい性除去識別子を格納する第 5 のデータフィールドと、

前記第 2 のファイルのための第 2 のあいまい性除去識別子を格納する第 6 のデータフィールドと

をさらに含み、前記第 1 および第 2 のあいまい性除去識別子は一意であることを特徴とする請求項 1 に記載のコンピュータ可読媒体。

【請求項 3】

各前記あいまい性除去識別子は一意の整数を含むことを特徴とする請求項 2 に記載のコンピュータ可読媒体。

【請求項 4】

前記ファイル情報テーブルと前記編成テーブルは別個のものであることを特徴とする請求項 1 に記載のコンピュータ可読媒体。

【請求項 5】

前記第 1 のファイル名は名前がないことを特徴とする請求項 1 に記載のコンピュータ可読媒体。

【請求項 6】

電子ファイルシステムにデータを格納する方法を実施するためのコンピュータ実行可能な命令を格納するコンピュータ可読媒体であって、該命令は、

(a) 前記ファイルシステムに格納された第 1 のアイテムのための第 1 の一意の識別子を前記ファイルシステムに格納するステップと、

(b) 前記第 1 のアイテムのユーザが見ることの可能な名前である第 1 のファイル名を前記ファイルシステムに格納するステップと、

(c) 前記ファイルシステムに格納された第 2 のアイテムのための第 2 の一意の識別子を前記ファイルシステムに格納するステップと、

(d) 前記第 2 のアイテムのユーザが見ることの可能な名前である前記第 1 のファイル名を前記ファイルシステムに格納するステップと、

(e) 前記第 1 のアイテムの親である第 3 のアイテムを識別する第 1 の保持リンクを

10

20

30

40

50

前記ファイルシステムに格納するステップと、

(f) 前記第2のアイテムの親である前記第3のアイテムを識別する第2の保持リンクを前記ファイルシステムに格納するステップと
を含むことを特徴とするコンピュータ可読媒体。

【請求項7】

前記命令は、

(a) 前記第1のアイテムのための第1のあいまい性除去識別子を格納するステップと、

(b) 前記第2のアイテムのための第2のあいまい性除去識別子を格納するステップと

をさらに含むことを特徴とする請求項6に記載のコンピュータ可読媒体。

【請求項8】

各前記あいまい性除去識別子は一意の整数を含むことを特徴とする請求項7に記載のコンピュータ可読媒体。

【請求項9】

前記命令は、

(a) 前記第1のあいまい性除去識別子に基づいて、前記第1のアイテムの完全に限
定されたパスを生成するステップ

をさらに含むことを特徴とする請求項7に記載のコンピュータ可読媒体。

【請求項10】

前記命令は、

(a) 前記第1のあいまい性除去識別子に基づいて、前記第1のアイテムへのルート
パスを生成するステップ

をさらに含むことを特徴とする請求項7に記載のコンピュータ可読媒体。

【請求項11】

電子ファイルシステムにデータを格納するためのコンピュータ支援による方法であって、

(a) 前記ファイルシステムに格納された第1のアイテムのための第1の一意の識別
子を前記ファイルシステムに格納するステップと、

(b) 前記第1のアイテムのユーザが見ることの可能な名前である第1のファイル名
を前記ファイルシステムに格納するステップと、

(c) 前記ファイルシステムに格納された第2のアイテムのための第2の一意の識別
子を前記ファイルシステムに格納するステップと、

(d) 前記第2のアイテムのユーザが見ることの可能な名前である前記第1のファ
イル名を前記ファイルシステムに格納するステップと、

(e) 前記第1のアイテムの親である第3のアイテムを識別する第1の保持リンクを
前記ファイルシステムに格納するステップと、

(f) 前記第2のアイテムの親である前記第3のアイテムを識別する第2の保持リン
クを前記ファイルシステムに格納するステップと

を含むことを特徴とするコンピュータ支援による方法。

【請求項12】

(a) 前記第1のアイテムのための第1のあいまい性除去識別子を格納するステップ
と、

(b) 前記第2のアイテムのための第2のあいまい性除去識別子を格納するステップ
と

をさらに含むことを特徴とする請求項11に記載のコンピュータ支援による方法。

【請求項13】

各前記あいまい性除去識別子は一意の整数を含むことを特徴とする請求項12に記載の
コンピュータ支援による方法。

【請求項14】

10

20

30

40

50

(a) 前記第1のあいまい性除去識別子に基づいて前記第1のアイテムの完全に限定されたパスを生成するステップ

をさらに含むことを特徴とする請求項12に記載のコンピュータ支援による方法。

【請求項15】

(a) 前記第1のあいまい性除去識別子に基づいて前記第1のアイテムへのルートパスを生成するステップ

をさらに含むことを特徴とする請求項12に記載のコンピュータ支援の方法。

【請求項16】

複数のユーザが見ることの可能なファイルを格納するための電子ファイルシステムを動作させるように構成されたコンピュータ装置であって、前記ユーザが見ることの可能なファイルの少なくとも2つは共通のあいまいなファイル名を有することを特徴とするコンピュータ装置。 10

【請求項17】

前記共通のあいまいなファイル名は名前のないファイル名を含むことを特徴とする請求項16に記載のコンピュータ装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般に、コンピュータシステム用のファイルシステムに関する。より詳細には、本発明は共通の親を持つ複数のアイテムがあいまいなファイル名をもつこと、例えば同一の名前を持つこと、あるいは名前を全く持たないことなどを許容するファイルシステムを提供することに関する。 20

【背景技術】

【0002】

コンピュータはハードディスクドライブ（以下、ハードドライブ）などの記憶装置にファイルを格納する。ハードドライブは、空（empty）のファイルキャビネット（書類棚）と同様に、単にデータを格納する場所だけを提供する。空のファイルキャビネットがファイルのためのどのような所定のファイリングシステムを強いることもないように（すなわち、ファイルキャビネットの使用者が自分自身で、例えばファイルをアルファベット順に並べることなどによって、ファイリングシステムまたは組織構造（organization structure）を生成しなければならないように）、ハードドライブもまた初期設定では単なる空の記憶空間である。したがって、ハードドライブ上のデータにアクセスする方法は、ディスク上のデータの物理位置を指定する（例えば、ファイルが格納されているハードドライブのシリンダ、ヘッド、およびセクタを指定する）か、あるいはディスク上のデータの論理位置（例えば、21, 671番目のブロック）を指定することに限られる。ハードディスクドライブがコンピュータにインストールされた後は、コンピュータは、ファイルシステムを使用して、ハードディスクドライブ上に格納されたファイルを、容易にアクセス可能な方法で把握することができる。 30

【0003】

既知のファイルシステムでは、オペレーティングシステムと、ファイルシステムのユーザとが、ファイルシステム中のファイルをどのように編成することができるのかが極度に制限されている。すなわち、既知のファイルシステムでは一般に、ユーザはファイルとディレクトリからなるツリーの形（木構造）でアイテムを編成する必要があり、その場合、ディレクトリは、実際には、ファイルシステムによって識別される特別な種類のファイルとなる。既知のファイルシステムではまた、各アイテムはユーザから見て唯一の名前を持つ必要がある。すなわち、各アイテムはその親（parent）の下で一意的な名前を持つことが必要になる。言い換えると、共通の親を持つ2つのアイテムも同一の名前を持つことはできないことになる（ただし、共通の祖先（ancestor）を持つ2つのアイテムは同一の名前を持つことができる）。一意的な名前が必要になるのは、名前が、オペレーティングシステム、ソフトウェアアプリケーション、およびユーザが、ファイルシ 40 50

テムに格納されたファイルまたはアイテムを検出し識別するための手段になっていたからである。しかし、ユーザがアクセスするファイルの数が益々増大してきているので、このファイル名の要件は問題が多く、耐え難い負担になってきている。

【0004】

デジタルカメラなどの装置が、ファイル名の問題を悪化させてきている。なぜならば、このような装置は、デジタル写真など大量の新しいファイルを生成し、そのファイルの各々が自動的に決定されるファイル名を持つものであるからである。一般に、このような装置では、異なるフォルダ間でファイル名が再利用され、また数値的に増分するファイル名が使用される。このため、ユーザはこれらの装置からパソコン（パーソナルコンピュータ）など他の装置にファイルをインポート（取り込み）する際に注意が必要になる。元のフォルダ構造が維持されない場合には、ユーザは名前が衝突し（すなわち、2つ以上のファイルが同じ場所で同じ名前を持ち）、ファイルのリネーム（ファイル名を変えること）が必要になる可能性について心配しなければならないこととなる。理屈の上では単純な仕事であるはずのファイルのインポートが、実際には、時間のかかるもどかしい仕事になる。

10

【0005】

図2は、現在のファイルシステムの代表的な組織構造201の簡単な例を示す。図2に示すように、既知のファイルシステムでは、ツリーを用いて、（丸い角の枠で囲んで示した）ディレクトリと、（四角い矩形枠で囲んで示した）ファイルとが構成される。ツリー構造では、アイテムの位置と編成とが融合され、各アイテムはただ1つのディレクトリ中に存在しなければならない。ユーザは、このような形で自分自身のファイルを編成する必要があり、したがってユーザはアイテムの新しいコピーを作成することなしに複数の編成にアイテムを配置することはできない。

20

【0006】

代表的なファイルシステムにおいては、相互に関連する2つのテーブルまたはデータベースを使用してファイルが編成される。第1のテーブルまたはデータベースは、ハードドライブなどの記憶装置上でファイルが格納されている物理位置を識別するためのルックアップテーブルである。第2のテーブルは、ファイルの組織構造を定義する。本明細書では、これらのテーブルを一般的に、それぞれロケーションテーブル（LOC）および編成テーブル（ORG；organizational table）と呼ぶ。編成テーブルは、保持リンク（holding links）、すなわち、あるアイテムが他のアイテムの親であることに関する情報を格納する。ファイルシステムによっては、ロケーションテーブルと編成テーブルを1つのテーブルまたは構造に組み合わせることができるが、ファイルシステムが適切に動作するためには、やはり両方のテーブルの要素が存在することが必要になる。例えば、マイクロソフトコーポレーションから販売されているNTブランドのファイルシステム、すなわちNTFSでは、マスタファイルテーブルがロケーションテーブルと編成テーブルの両方の機能を果たす。同様に、Unix（登録商標）ファイルシステム、すなわちUFSではiノードのテーブルが使用されるが、このテーブルもファイルのロケーションテーブルと編成テーブルの両方の機能を果たす。ディレクトリは特別な種類のファイルとして格納される。すなわち、ディレクトリ「ファイル」は、そのディレクトリ内のファイル名とそのファイルに対応するiノードのリストを格納する。

30

40

【0007】

これらおよび他の既知のファイルシステムでは、ファイルが編成テーブル中の保持リンクによって定義された少なくとも1つの位置に置かれている限り、ファイルシステムはそのファイルを物理記憶装置上に格納しておく。すなわち、ファイルを指示する少なくとも1つの保持リンクが存在することになる。言い換えると、あるファイルに対する保持リンクが編成テーブルから削除され、そのファイルを指示する保持リンクが存在しなくなった場合には、ファイルシステムは（そのファイルが記憶装置上に物理的に上書きされたかどうかに関係なく）ロケーションテーブル中のそのファイルのエントリを除去する。次いで、記憶装置はその記憶空間を新しいデータを書き込むために使用することができる。例えば、ユーザが図2に示したファイル、C:\PROGRAMS\MICROSOFT\O

50

FFICE\WORD\FILE.DOCを「削除」しようとした場合には、ファイルシステムは最初に編成テーブルからそのファイルのエントリを除去する。そのファイルがロケーションテーブル内に他のエントリを有していない場合に、すなわち、そのファイルが他のどの場所にも格納されていない場合には、ファイルシステムはロケーションテーブル内のそのファイルのエントリを除去し、他のデータのために空間を開放する。一般に、ファイルシステムは、所与のアイテムが有する（「保持リンク」と呼ばれる）祖先の数の参照カウントを維持する。あるアイテムについての最後の保持リンクが除去されたときには、ロケーションテーブル中のそのファイルの参照を除去することによって、そのアイテムは「削除」される。しかし、参照カウントはエンドユーザの観点からは望ましくない。なぜならば、非循環有向グラフ（directed acyclic graph）とともに使用されたときに、そのシステムが非決定性（non-deterministic）に見えるからである。 10

【発明の開示】

【発明が解決しようとする課題】

【0008】

上記の制限や制約のために、ディレクトリおよびファイルからなるツリー状の階層構造以外のデータ構造でファイルを編成することは困難になる。ユーザは、所与のアイテムが削除されることを心配せずに、様々な組織データ構造においてアイテムを編成し編成解除できることを望む。また、同一フォルダ内の各アイテムは名前を持つ必要があるが、重複した名前を持つことはできない。したがって、格納されたアイテムが名前を持つことを必要としないファイルシステムが提供できれば、当技術分野における進歩・前進になるはずである。共通の親を持つ2つのアイテムが、同一の名前を持つことができるファイルシステムを提供できれば、当技術分野における別の進歩・前進になるはずである。 20

【課題を解決するための手段】

【0009】

以下では、本発明のいくつかの態様についての基本的な理解が得られるように本発明の簡単な要約を提示する。この要約は本発明を幅広く概観するものではない。また、本発明の主要要素または重要要素を識別すること、あるいは本発明の範囲を記述することを意図したものでもない。後で記述されるより詳細な説明の準備として、本発明のいくつかの概念を簡単な形で提示することだけを目的としたものである。 30

【0010】

上記の従来技術における1つまたは複数の制限を克服するため、および／または本明細書を読み理解したときに明らかになるはずの他の制限を克服するために、本発明は一般に、コンピュータ可読媒体に格納されたデータアイテムおよびコンピュータ実行可能命令として実施されるファイルシステムを対象とするものである。データアイテムは一般に、これらに限られるものではないが、ファイル、フォルダ、データ、音楽など、ファイルシステムに格納できる任意のデータを意味する。ファイルシステムは、ファイル情報テーブルおよび編成テーブルを含むデータ構造を使用することができる。ファイル情報テーブルは、第1のアイテムのための一意の第1の識別子を格納する第1のデータフィールド、第1のアイテムのユーザが見ることの可能な名前である第1のファイル名を格納する第2のデータフィールド、第2のアイテムのための一意の第2の識別子を格納する第3のデータフィールド、および第2のアイテムのユーザが見ることの可能な名前である第1のファイル名を格納する第4のデータフィールドを保管する。編成テーブルは、第1のアイテムの親である第3のアイテムの指示を格納する第1のデータフィールド、および第2のアイテムの親である第3のアイテムの指示を格納する第2のデータフィールドを保管する。 40

【0011】

本発明の他の例示的態様によれば、ファイルシステムにアイテムを格納するためのコンピュータで実行される方法が提供される。この方法によると、ファイルシステムクライアント（コンピュータ）は同一の親を有する2つのアイテムに対してあいまいなファイル名を格納することができるようになる。一般にこの方法は、ファイルシステムに格納された 50

第1のアイテムの第1の一意の識別子をファイルシステムに格納すること、第1のアイテムのユーザが見ることの可能な名前である第1のファイル名をファイルシステムに格納すること、ファイルシステムに格納された第2のアイテムの第2の一意の識別子をファイルシステムに格納すること、第2のアイテムのユーザが見ることの可能な名前である第1のファイル名をファイルシステムに格納すること、第1のアイテムの親である第3のアイテムを識別する第1の保持リンクをファイルシステムに格納すること、および第2のアイテムの親である第3のアイテムを識別する第2の保持リンクをファイルシステムに格納することを含む。

【0012】

本発明および本発明の効果・利点についてのより完全な理解は、以下の説明を添付の図面を考慮しながら参照することによって得ることができる。図面中の同一の参照番号は同一の機能を示す。 10

【発明を実施するための最良の形態】

【0013】

様々な実施形態についての以下の説明では、添付の図面を参照する。添付の図面は本発明の一部をなすものであり、本発明を実施することができる様々な実施形態を例示したものである。その他の実施形態を使用することができること、ならびに本発明の範囲を逸脱することなく構造および機能に関する修正をなし得ることを理解されたい。

【0014】

例示的コンピュータ環境

図1は、本発明を実装することができる適切なコンピュータシステム環境100の一例を示す。このコンピュータシステム環境100は、適切なコンピュータ環境の一例に過ぎず、本発明の使用または機能の範囲に関して、いかなる制限を加えることも意図したものではない。コンピュータ環境100は、例示的動作環境100に示された各コンポーネントの任意の1つまたは組合せに関して、いかなる依存性または要件も有していないものと理解されたい。 20

【0015】

本発明は、他の多くの汎用または専用コンピュータシステム環境または構成とともに使用することができる。本発明とともに使用するのに適した周知のコンピュータシステム、環境、および／または構成の例には、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドもしくはラップトップ装置、マルチプロセッサシステム、マイクロコントローラベースシステム、セットトップボックス、プログラマブル家電、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ（大型コンピュータ）、上記のシステムまたは装置のいくつかを含む分散コンピュータ環境などが含まれるが、これらに限られるものではない。 30

【0016】

本発明は、コンピュータによって実行される、プログラムモジュールなどコンピュータ実行可能な命令の一般的なコンテキストで説明することができる。一般に、プログラムモジュールは、特定のタスクを実行するか、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。本発明は、通信ネットワークを介してリンクされたリモート（遠隔）処理装置によって各タスクが実行される分散コンピュータ環境においても実施することができる。分散コンピュータ環境では、各プログラムモジュールはローカル（局所）およびリモート（遠隔）のどちらのコンピュータ記憶媒体、例えばメモリ記憶装置にも配置することができる。 40

【0017】

図1を参照するように、本発明を実行する典型的なシステムには、コンピュータ110のようなタイプの汎用コンピュータ装置が含まれる。コンピュータ110の各コンポーネントには処理ユニット120、システムメモリ130、システムバス121などが含まれるが、これらに限られるものではない。システムバスは、システムメモリなど様々なシステムコンポーネントを処理ユニット120に連結する。システムバス121には、メモリ 50

バスもしくはメモリコントローラ、周辺機器バス、および様々なバスアーキテクチャのいずれかを使用するローカルバスなど、いくつかの種類のバス構造のうちのいずれかを使用することができる。例として、このようなアーキテクチャにはISA (Industry Standard Architecture) バス、MCA (Micro Channel Architecture) バス、EISA (Enhanced ISA) バス、VESA (Video Electronics Standards Association) ローカルバス、メザニン (Mezzanine) バスとも呼ばれるPCI (Peripheral Component Interconnect) バスなどが含まれるが、これらに限られるものではない。

【0018】

コンピュータ110は一般に、様々なコンピュータ可読媒体を含む。コンピュータ可読媒体は、コンピュータ110によってアクセス可能な任意の使用可能媒体であればよく、揮発性および不揮発性媒体、取り外し可能および取り外し不能媒体の両方を含んでいる。一例として、コンピュータ可読媒体にはコンピュータ記憶媒体および通信媒体を含めることができるが、これらに限られるものではない。コンピュータ記憶媒体は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、その他のデータなどの情報を記憶するためのいずれかの方法または技術で実装された、揮発性および不揮発性、取り外し可能および取り外し不能の両方の媒体を含む。コンピュータ記憶媒体には、これらに限られるものではないが、RAM、ROM、EEPROM、フラッシュメモリなどのメモリ技術製品、CD-ROM、DVD (digital versatile disks; デジタル多用途ディスク) などの光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置などの磁気記憶装置、あるいは所望の情報を格納するために使用でき、かつコンピュータ110によってアクセス可能な他の任意の媒体、または媒体の組合せが含まれる。通信媒体は一般に、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、その他のデータなどを、搬送波や他の伝送機構などの変調データ信号の形で実施するとともに、いずれかの情報送達媒体を含んでいる。「変調データ信号」という用語は、1つまたは複数の信号特性セットを含むか、または情報を信号中に符号化するような形で変更された信号を意味する。例として、通信媒体には、有線ネットワーク、直接有線接続などの有線媒体、ならびに音響、RF、赤外、その他のワイヤレス無線媒体などの無線媒体が含まれるが、これらに限られるものではない。

【0019】

システムメモリ130は、ROM (読み出し専用メモリ) 131やRAM (ランダムアクセスメモリ) 132など、揮発性および／または不揮発性メモリの形のコンピュータ記憶媒体を含む。基本入／出力システム (BIOS) 133は、例えば起動時などに、コンピュータ110内の各要素間の情報転送を支援する基本ルーチンを含み、通常はROM 131に格納されている。RAM 132は一般に、処理ユニット120によって即時にアクセス可能かつ／または現在実行中の、データおよび／またはプログラムモジュールを含む。一例として、図1にはオペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136、プログラムデータ137、およびファイルシステム138が示すが、これらに限られるものではない。

【0020】

コンピュータ110は、他の取り外し可能／取り外し不能、揮発性／不揮発性のコンピュータ記憶媒体を含むこともできる。例示の意味で図1には、取り外し不能な不揮発性の磁気媒体に読み書きするためのハードディスクドライブ141、取り外し可能な不揮発性磁気ディスク152に読み書きするための磁気ディスクドライブ151、ならびにCD-ROMや他の光媒体など取り外し可能で不揮発性の光ディスク156に読み書きするための光ディスクドライブ155が示す。例示の動作環境で使用できる他の取り外し可能／取り外し不能、揮発性／不揮発性のコンピュータ記憶媒体には、磁気テープカセット、フラッシュメモリカード、DVD、デジタルビデオテープ、半導体RAM、半導体ROMなどがあるが、これに限られるものではない。ハードディスクドライブ141は一般に、イン

10

20

30

40

50

タフェース140など取り外し不能メモリアンタフェースを介してシステムバス121に接続されるが、磁気ディスクドライブ151および光ディスクドライブ155は一般に、インタフェース150など取り外し可能メモリアンタフェースを介してシステムバス121に接続される。

【0021】

上で論じ、図1に示したドライブ（駆動装置）および関連するコンピュータ記憶媒体は、コンピュータ110のためのコンピュータ可読の命令、データ構造、プログラムモジュール、その他のデータなどに対する記憶を提供する。図1では、例えばハードディスクドライブ141は、オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、プログラムデータ147、およびファイルシステム148を格納するように図示している。これらの各コンポーネントは、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136、プログラムデータ137、およびファイルシステム138と同一であっても、異なっていることもよいことに留意されたい。オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、プログラムデータ147、およびファイルシステム148は、例えばそれらが異なるコピーであることを示すために、図1では異なる番号が付与されている。ユーザは入力装置、例えばキーボード162や、マウス、トラックボール、タッチパッドなどを総称するポインティング装置161を介して、コンピュータ110に命令および情報を入力することができる。他の入力装置として、リモコン（遠隔制御器）163、マイクロホン、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナなどを含むことができる（これらすべてを図示している訳ではない）。これらおよび他の入力装置は、システムバスに結合されたユーザ入力インタフェース160を介して、処理ユニット120に接続されることが多いが、パラレルポート、ゲームポート、またはUSB（universal serial bus）など、他のインタフェースやバス構造を用いて接続することもできる。モニタ191または他の種類の表示装置（例えば、TV）もまた、ビデオインタフェース190などのインタフェースを介してシステムバス121に接続される。コンピュータは、モニタの他に、例えば出力周辺機器インタフェース195を介して接続されるスピーカ197、プリンタ196など他の周辺出力装置を含むこともできる。

【0022】

いくつかの態様では、手書き入力をデジタル的に取り込むためにペンデジタイザ165および付属のペンまたはスタイラス166が備えられている。ペンデジタイザ165とユーザ入力インタフェース160の間は直接接続されているように図示しているが、実際には、ペンデジタイザ165は無線などの任意の技術によって、処理ユニット120に直接結合すること、パラレルポートもしくは他のインタフェースに結合すること、およびシステムバス121に結合することができる。ペン166はまた、それに結合されたカメラ、およびカメラによって取り込まれた画像情報をバス121を介して対話するインタフェースに無線で送信するためのトランシーバを備えることもできる。さらにペンには、加速度計、磁力計、ジャイロスコープなど他のセンシングシステムをカメラに追加し、またはカメラの代わりに備えることで、電子インクのストロークを決定することもできる。

【0023】

コンピュータ110は、リモートコンピュータ（遠隔地コンピュータ）180など1つまたは複数のリモートコンピュータへの論理接続を使用してネットワーク環境で動作させることができる。リモートコンピュータ180は、パーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピア(peer)装置、または他の共通ネットワークノードなどであってもよい。また、図1にはリモートコンピュータ180に対してメモリ記憶装置181だけを示すが、リモートコンピュータ180は一般に、コンピュータ110に関連して上で説明した要素の多く、またはすべてを含むことができる。図1に示した論理接続には、ローカルエリアネットワーク（LAN）171および広域ネットワーク（WAN）180が含まれているが、他のネットワークを含めることもできる。このようなネットワー

10

20

30

40

50

ク環境は、オフィス（事務所）、企業規模のコンピュータネットワーク、イントラネット（企業内ネットワーク）、インターネットなどにおいて一般的なものである。さらに、このシステムは、有線および／または無線の機能を含むこともできる。例えば、ネットワークインタフェース170は、ブルートゥース（Bluetooth）、SWLAN、および／またはIEEE802.11クラスの組合せ機能を含むことができる。これらのプロトコルとともに、またはこれらのプロトコルの代わりに、他の無線通信プロトコルが使用できることを理解されたい。

【0024】

LANネットワーク環境で使用されるとき、コンピュータ110は、ネットワークインタフェースまたはアダプタ170を介してLAN171に接続される。WANネットワーク環境で使用されるとき、コンピュータ110は一般に、インターネットなどのWAN173を介して通信を確立するためのモデム172などの手段を含む。モデム172は、内蔵のことも外付けのこともあるが、ユーザ入力インタフェース160または他の適切な機構によってシステムバス121に接続することができる。ネットワーク環境では、コンピュータ110に関連して示した各プログラムモジュールまたはその一部は、リモートのメモリ記憶装置に格納することができる。例えば、図1では、リモートのアプリケーションプログラム185が、メモリ装置181に常駐するように示しているが、これに限られる訳ではない。図1に示すネットワーク接続は例示的なものであり、コンピュータ間の通信リンクを確立するための他の手段を使用してもよいことが理解されるであろう。

【0025】

図1に図示したネットワーク接続が例示的なものであり、コンピュータ間の通信リンクを確立するための他の技術を使用できることが理解されよう。TCP/IP、イーサネット（登録商標）、FTP、HTTPなどの様々な周知のプロトコルのうちのいずれかの存在を想定している。このシステムは、ユーザがウェブベースのサーバからウェブページを取り出すことができるように、クライアントサーバ構成で動作することができる。様々な従来のウェブブラウザのいずれかを使用してウェブページ上のデータを表示し、操作することができる。

【0026】

本発明の1つまたは複数の態様は、コンピュータ110など1台または複数台のコンピュータ、または他の装置によって実行される1つまたは複数のプログラムモジュールなどのコンピュータ実行可能な命令の形で実施することができる。一般に、プログラムモジュールは、コンピュータまたは他の装置のプロセッサによって実行されるときに、特定のタスクを実施するか、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。コンピュータ実行可能な命令は、ハードディスク、光ディスク、取り外し可能記憶媒体、半導体メモリ、RAMなどのコンピュータ可読媒体に格納することができる。当業者ならば理解できるはずであるが、プログラムモジュールの機能は、様々な実施形態での必要に応じて組み合わせること、または分散させることができる。さらに、各機能は、その全体または一部を、集積回路やFPGA（field programmable gate arrays；フィールド・プログラマブル・ゲート・アレー）などのファームウェアまたはハードウェア均等物の形で実施

【0027】

本発明の例示的な実施形態の例

本発明の例示的態様に従って、本明細書には、アイテムのライフタイムとファイルシステムの組織構造内のアイテムの配置とを融合させないことによって、例えば特性ベースの照会やリストなどを用いることによって、アイテムが複数の編成に参加すること、あるいはどの編成にも参加しないことを許容し、それら編成が1つまたは複数の非循環有向グラフ（DAG）を形成するというファイルシステムが記載されている。DAGはアイテムが複数の親を持つことができる編成である。DAGにおいては、アイテムがその祖先の1つの親になり1つのサイクルを形成することを避けることができる。

【0028】

本明細書で説明するファイルシステムは、例えばファイルシステム138および／またはファイルシステム148（図1）としてコンピュータ可読媒体に格納されたコンピュータ実行可能命令として実施することができる。さらに、図1の他のコンピュータ可読媒体またはメモリもまた、本明細書で説明するファイルシステムと同一の、またはそれとは異なるファイルシステムを含むことができる。例えば、媒体152および156もまたファイルシステムを含むことができる。説明のために、本明細書ではファイルシステムをファイルシステム148に関連して述べることにする。

【0029】

図3および図4を参照すると、ファイルシステム148は、主に2つのテーブル、すなわちロケーションテーブル（LOC）301と編成テーブル（ORG）401に依存している。LOCテーブルとも呼ばれるロケーションテーブル301は、ファイルシステムに格納された各アイテムの物理および／または論理・記憶位置に関する情報、例えば、ファイル識別子（ID）303、ファイル名305、記憶ボリューム307、ロケーション（記憶場所）309、およびファイルサイズ311などを格納する。ファイルID303は各アイテムを参照するために使用される基本キーまたは照会先（reference）であり、ファイルシステム148に格納された各アイテムに対して一意である。ファイルIDは、各ファイルIDがファイルシステム148に格納された単一のアイテムを一意に識別する限りは、数字、アルファベットテキスト、シンボル、またはそれらの任意の組合せとすることができる。

【0030】

ファイル名305は、ファイルシステム148のユーザによって付けられた、アイテムの短い記述的ファイル名とすることができる。ファイル名305は各ファイルに対するユーザの一番の概念的参照であり、ファイル名に基づいてユーザがファイルを容易に認識できるように記述的な形でユーザが指定できることが好ましい。ファイル名305では、例えば、8.3フォーマット、256文字の最大長、またはその他の最大長さなど、長さを制限することができる。ファイル名が互いに重複しているかどうかに関する制約、あるいはファイル名が必要であるかどうか、または割り当てられているかどうかに関する制約さえ無いことが好ましい。すなわち、各ファイルが一意のファイルID303を有する限りは、同一のディレクトリ、フォルダ、またはリスト内にあっても、複数のファイルをそれぞれ「README.TXT」または「Read this first」などと名付けることができる。この点は、各ファイルがその親のコンテキスト内で一意の名前を有する必要がある従来のファイルシステムと異なる。

【0031】

ボリューム307、ロケーション309、サイズ311を使用して、物理または論理記憶位置からアイテムを取り出すことができる。ボリューム307により、物理または論理記憶装置、例えば物理または論理ハードディスクドライブ、光ディスク記憶装置、記憶媒体カード、半導体記憶素子などを参照することができる。ロケーション309により、識別された物理または論理ボリューム上のアイテムの実際の開始位置を参照することができる。また、サイズ311により、識別されたボリューム上で、識別されたロケーションから始まるアイテムが格納されたスペースの量についての定量的測度を参照することができる。各識別された記憶位置から各格納されたアイテムを取り出すために含まれた情報が使用できる限りは、ロケーションテーブル301に追加の、または代替の情報を含めることができる。

【0032】

ORGテーブルとも呼ばれる編成テーブル401は、ファイルシステム148に格納されたアイテム間の階層関係を格納する。先に要約して述べたように、ORGテーブル401は1つまたは複数の非循環有向グラフ（DAG）を定義することができる。非循環有向グラフは、ファイルシステム148内のアイテムを概念的に配列するための1つまたは複数のツリーを含むことができる。ORGテーブル401は、各関連ごとに親403と子4

10

20

30

40

50

05を含むことができる。図5は、LOCテーブル301の一部によって定義され、またORGテーブル401によって定義されたDAG501の一例を示す。DAG501はツリーではない。なぜならば、アイテムI3については「To Do List」と「Items For My Trip」の両方が親になっており（両方に「属する(in)」とも呼ばれる）、アイテムI4については「Items For My Trip」と「Key Slide Decks」の両方が親になっているからである。ファイルシステム148に格納された各アイテムは、組織構造がDAGである限りは、親を持たないこと、1つの親を持つこと、または2つ以上の親を持つことができる。

【0033】

図6は、ファイルシステム148で 사용할 ことができるソフトウェアモジュールのブロック図を示す。ファイルシステム148は、ファイルシステム148全体の動作を管理するファイルシステムマネージャソフトウェアモジュール601を含むことができる。マネージャ601は、本明細書に記載のLOCテーブル301およびORGテーブル401を使用して、（例えば、循環が許されないなど）DAGに設定されたあらゆる制約に関するルールを実施しながら、データストア611に格納された各アイテムの追加、更新、削除、および読み出し／照会（問合せ）を実施することができる。マネージャ601は、オペレーティングシステムなどの上位プログラムがファイルシステム148と対話できるようにする1つまたは複数のAPI（アプリケーションプログラムインタフェース）を公開することができる。APIには、Add__Item API 603、Update__Item API 605、Delete__Item API 607、およびQuery__Item API 609を含めることができる。

【0034】

Add__Item API 603は、新しいアイテムをファイルシステム148に格納するときに、オペレーティングシステムや他の上位プログラムまたはアプリケーションによってコールされることができる。Add__Item API 603は、新しいアイテムの現在の一時記憶場所へのポインタおよびファイル名、ならびにオプションにより対象とする記憶ボリュームおよび／または親IDを入力として受け取ることができる。Add__Item APIは、新しく追加されたアイテムのファイルIDをオプションにより返すことができる。

【0035】

Update__Item API 605は、ファイルシステム148に既に格納済みの既存のアイテムを更新するためにコールすることができる。また、入力として、ファイルIDならびに1つまたは複数のオプションで含められた情報、すなわちアイテムの更新されたバージョンへのポインタ、ORGテーブル401に含めるための更新された関係（関連）、および／または更新されたファイル名の中から選択された情報を受け取ることができる。

【0036】

Delete__Item API 607は、削除するファイルIDをそのDelete__Item APIに渡すことによって、ファイルシステム148からアイテムを除去するためにコールすることができる。アイテムがファイルシステム148から削除されるのは、Delete__Item API 607が明白にコールされたときだけである。

【0037】

Query__Item API 609は、OSまたはアプリケーションによって使用するためにファイルシステムからアイテムを取り出す（読み出す、検索する）必要があるときにコールすることができる。Query__Itemは、入力としてファイルIDを受け取り、要求されたアイテムを返す。上記の各APIはAPIの種類のうちの代表的なものであり、本発明の例示的实施形態に含めることが好ましい。ファイルシステム148には、システムの必要性または設計に基づいて追加の、または代替のAPIを含めることもできる。

【0038】

10

20

30

40

50

ファイルシステム 148 は各アイテムが 0 を含む任意の数の親を持つことを許容するので、単一のボリュームまたはファイルシステム内に複数の独立したデータ構造（すなわち、複数の DAG）が存在し得る。この点が既知のファイルシステムと異なっており、既知のファイルシステムでは、一般にクライアント（ユーザまたはアプリケーション）が各ボリュームあたり単一のツリーにアイテムを格納するように制限されている。したがって、本明細書で説明するようなファイルシステムに各アイテムが格納される概念空間は、ファイル領域（FR）と呼ばれる。ファイル領域は、アイテムのライフタイムを管理するために使用される編成領域（organizational area）の上位ユーザ概念（high-level user concept）を意味する。アイテムがファイル領域にとどまる限りは、そのアイテムが ORG テーブル 401 に定義されたいずれかの親子関係に従うかどうかとは関係なく、そのアイテムが削除されることはない。

【0039】

ファイル領域は、各アイテムが概念的に関連づけられているかどうかに関係なく、概念的にはアイテムを置くことができるボックスと考えることができる。アイテムがファイル領域に置かれている場合は、そのアイテムは、ユーザがそのアイテムを削除するまでファイル領域内にとどまる。アイテムがファイル領域内に存在している間は、アイテムがファイル領域内に存在するかどうかに影響を与えることなく、またアイテムのライフタイムに影響を与えることもなく、アイテムはファイル領域内の 1 つまたは複数の DAG のいずれかで構成することができる。すなわち、ファイルは、そのファイルを削除することをユーザがファイルシステムに断言的に指示するまでは削除されることはない。アイテムのライフタイムを保証するために、データストアが単一のファイル領域として管理される限りは、関係するすべてのデータストアが同一のコンピュータ、ネットワークなどの中で互いに物理的に接近しているかどうかに関係なく、ファイル領域はどのように定義された記憶領域でも包含することができる。例えば、ファイル領域は、コンピュータのすべての内部ハードドライブ、コンピュータの単一のハードドライブの一部、単体またはローカルの記憶装置と組み合わされたネットワーク上の記憶装置、または他の任意に定義された記憶空間を包含することができる。

【0040】

図 7 および図 8 は、3 つの独立した DAG 501、703、705 を含むファイル領域 701 を示す。DAG 501、703、および 705 の間で要求される関係は、それらが同一のファイル領域内にあることだけであり、それらの間の概念的関係は要求されない。図 8 は、図 7 に示した DAG を記述する ORG テーブル 801 を示す。図 7 は、ファイル領域 701 の概念的な図を示し、図 8 はファイル領域 701 で使用されている保持リンクを示す。ファイル領域 701 は、ユーザによってオープンすることができ、オープンされた時点で、ファイルシステム 148 はユーザにファイルシステムによって定義されたファイル領域内のすべての「トップレベル（最高位）」アイテムまたは「フローティング（浮動）」アイテムを提示することができる。本明細書の使用では、あるアイテムが、ファイル領域に対応する ORG テーブル内の他のアイテムのどの保持リンクからもターゲットにされていない場合に、そのアイテムを「トップレベル」または「フローティング」と定義する。したがって、図 7 では、アイテム i1、i7、To Do List、および Items For My Trip がトップレベルアイテムまたはフローティングアイテムと見なされる。エンドユーザにとって、リスト i7 は組織データ構造の形になっている。複数の親の概念に基づけば、アイテムは、例えば i3 のように複数のリスト（目録）に属することができる。リストは、順序通りまたは順不同で他のアイテムを保持するアイテムであり、リストは、ユーザがこれまでのファイルシステムにおいて取り扱いに慣れたしんでいるフォルダ／ディレクトリに置き換えることができる。リストはまた、他のリストに属するか、または他のリストを含むことができる。

【0041】

ファイルシステム 148 は、ユーザまたはアプリケーション（まとめてクライアントと呼ぶ）が、アイテムの基になる組織構造からアイテムを除去する（remove）ことを

望む場合と、クライアントがファイルシステムからアイテムを完全に削除する（delete）ことを望む場合を区別する。図9および図10は、ユーザまたはアプリケーションがKey Slide Decksリストからアイテムi5を除去した後のファイル領域701を示す。すなわち、ファイルシステムマネージャ601は、Update_Item API 605の要求に基づいて、アイテムi5（ファイルID 000006）の親であるKey Slide Decks（ファイルID 000007）を識別する、ORGテーブル1001中の保持リンクを除去する。アイテムi5はもはや複数アイテム組織構造の一部ではない。なぜならば、ORGテーブル401からアイテムi5に関連するすべての保持リンクが除去されたからである。しかし、アイテムi5はまだファイル領域701に残っているため、アイテムi5はファイルシステム148またはLOCテーブル301からは削除されていない。言い換えると、クライアントがアイテムi5を削除することを要求したのではなく、アイテムi5を除去（remove）するよう要求したので、アイテムi5はデータストア611およびLOCテーブル301から削除されなかったことになる。クライアントが積極的に、断言的にアイテムi5を選択し、Delete_Item API 607による削除動作を要求したときだけ、マネージャ601はデータストア611およびLOCテーブル301からアイテムi5（ファイルID 000006）を抹消する。

【0042】

図11および図12を参照すると、ファイルシステム148を使用して、従来から知られているファイルシステム、例えばディレクトリツリーで使用されている組織構造に基づいて、ユーザによく知られた組織構造を模擬（模倣）することができる。LOCテーブル301（図3）を再度参照すると、ファイルID 000301～000312を有するアイテムは、ファイルシステム148に格納されたリストの保持リンクを表す。各順不同のリストは、ファイル領域1201（図12）およびORGテーブル1101（図11）の形で概念的に表されたディレクトリツリー1203中のディレクトリを表す。このディレクトリツリー1203と従来から知られているディレクトリツリーの主な違いは、サブディレクトリ「Shared」が順不同リストの2つの「User Data」のどちらからもアクセスできることであり、その結果ファイルシステム148が複数の親の機能をうまく利用できることである。このようにして、今までどおりユーザが使い慣れた組織構造を提供しながら、各ディレクトリをユーザ間で容易に共用することができる。

【0043】

本発明のこの例示的实施形態によれば、ファイルシステム148は、ファイルシステム148が実装されたコンピュータシステムまたはネットワークの各個別のユーザのための専用ワークスペースを提供し、またそのコンピュータシステムまたはネットワークのすべてのユーザが使用できる共通共用ワークスペースも提供する。どちらもがUser Dataの下にあるPrivateおよびSharedファイル領域を含むような名前空間が形成される。これにより、システムに対するユーザの照会をUser Dataによって形成されたアイテムドメインで検索することが可能になり、その結果、すべての一般的な照会に対して専用と共用の両方のアイテムが返されることになる。図12の名前空間を使用することにより、（新しい写真または音楽を入手する、文書を作成するなどのために）ユーザがファイルシステムの各アイテムと対話するとき、ユーザは概念的に「これは専用か共用か？」の質問を考えることだけが必要であって、そのアイテムを適切なワークスペース内に置くことができる。次いで、Sharedワークスペースのすべてのユーザは、Sharedワークスペースへの共通の可視性を有する。さらに、ワークスペースごとに共用のリサイクルごみ箱を設置して、あるユーザが他のユーザがまだ必要としているアイテムを削除した場合でも他のユーザがそのアイテムを取り出せるようにすることもできる。

【0044】

共通共用ワークスペースのシナリオはホームユーザに限定されるものではない。企業ドメイン上では、知識労働者はプロジェクトで協働するために同僚のための共用ワークスペース

10

20

30

40

50

ースを提供することができる。ワークスペースの柔軟性により、ワークスペースの各ユーザに、アイテムを構成し、ワークスペース内のアイテムを照会する機能が与えられ、これらのアイテムについての共通の全体の可視性が提供される。

【0045】

図13は、アイテムの基になる組織構造からアイテムを除去するメソッド、および／またはファイルシステム148からアイテムを削除するメソッドを示す。最初に、ステップ1301で、クライアントは特定のファイルIDを有するアイテムを選択する。次に、ステップ1303で、クライアントは、クライアントがファイルシステム148から完全にアイテムを削除したいのか、あるいはクライアントが単にアイテムの基になる組織構造からアイテムを除去したいのかを指示する。ステップ1303で、クライアントが「除去」を選択した場合は、ステップ1305でファイルシステムマネージャは、アイテムのファイルIDが子カラム（子の欄）に存在するあらゆるエントリをORGテーブルから抹消する。次いで、このメソッドは終了する。

【0046】

ステップ1303で、クライアントが「削除」を選択した場合には、ステップ1307でファイルシステムマネージャ601は、アイテムのファイルIDが親カラム（親の欄）に存在する各ORGテーブルエントリごとに、そのアイテムの子が子であり、そのアイテムの親が親であるエントリを、それらがORGテーブルに未だ存在していない場合には、ORGテーブルに追加する。以下に、使用可能な擬似コードを示す。

【0047】

```
For each ORG table entry X where Xparent = file_ID(ITEM)
  For each ORG table entry Y where Ychild = file_ID(ITEM)
    If ORG table does not have entry (Yparent, Xchild)
      Then Add ORG table entry (Yparent, Xchild)
```

【0048】

このようにしてORGテーブルを更新した後に、ステップ1309で、ファイルシステムマネージャは、そのアイテムのファイルIDを親または子として参照するあらゆるエントリをORGテーブルから削除する。最後に、ステップ1311で、ファイルシステムマネージャ601は、LOCテーブルからそのアイテムが格納された場所を示すエントリを削除する。ファイルシステムマネージャ601はさらに、安全削除動作を実施し、削除取り消し動作ができないようにガーベッジデータ（不要なデータ）を有するアイテムの位置に蓄積データを上書きする。

【0049】

別法として、ファイルシステムマネージャはステップ1307をスキップし、ステップ1309で単に、そのアイテムが親または子であるあらゆるエントリをORGテーブルから消去することもできる。この別法の副作用は、削除されるそのアイテムの各子が、ファイルシステム148の組織構造に引き続き存在する第2のアイテムがその子の親になっている場合を除き、ファイルシステム148の組織構造から消去されるようになることである。

【0050】

上記の方法は、選択されたアイテムおよびその選択されたアイテムによって指示されるあらゆるアイテムが基になる組織構造から除去されるように説明されている。しかし、当業者ならば、あるアイテムが組織構造から除去されるときにそのアイテムの子が組織構造から除去されない代替方法も使用できることを理解されたい。除去方法がどのように行なわれるかは、除去（removal）と削除（deletion）が2つの別個のプロセスであるという事実ほど重要ではない。削除プロセスは記憶装置全体からアイテムを削除するが、除去プロセスは、アイテムのライフタイムに影響を与えることなく、基になる組織構造からアイテムを除去する（すなわち、記憶装置またはファイルシステムからアイテムを削除する訳ではない）。

【0051】

従来のファイルシステムナビゲーションツール、例えば米国ワシントン州レッドモンドのマイクロソフトコーポレーションによるWindows（登録商標）EXPLORERブランドのシステムナビゲーションツールは、一般にファイルシステムの組織構造のファイルだけを示す。したがって、クライアントがファイルシステム148の組織構造からアイテムを除去した場合、すなわちアイテムがORGテーブルに存在しない場合には、そのアイテムはファイルシステムナビゲーション表示には表示されない。しかし、本発明の諸態様を使用すると、クライアントは、ファイルシステムマネージャ601にすべてのフローティングまたはトップレベルアイテム（すなわち、親がないアイテム）を照会して、現在の組織構造中に存在しない任意のアイテムを見つけ出すことができる。

【0052】

ファイルシステム148は、ORGテーブル401によって定義された組織構造に基づいて、ドメイン照会、セキュリティなど追加の機能を提供することができる。すなわち、組織構造の定義に加えて、ORGテーブル中の保持リンクを使用して、名前空間（ネームスペース）とも呼ばれる照会可能なアイテムドメインを形成することができる。クライアントは、そのアイテムが親である、少なくとも1つの保持リンクに関係する任意のアイテムを照会することができる。例えば、図7を参照すると、「Items For My Trip」についての照会では、アイテムi3、i4、i5、i8ならびにKey Slide Decksが返される。重複アイテムは、一度だけ返されることが好ましい。例えばi4については、Key Slide Decksも親になっているが、一度だけ返すことが好ましい。

【0053】

ファイルシステム148の組織構造を使用して、レガシー（legacy）アプリケーション（古い仕様や技術のアプリケーション）による使用のために名前空間を伝達することもできる。また、新しいユーザに、ファイルシステム148と対話するための概念的に認識可能なユーザインタフェースを提供することもできる。例えば、図12を参照すると、「U1」の下の子アイテム「User Data」は、クライアントから、\Users\U1\User Dataとして参照することができ、エンドユーザおよびレガシーアプリケーションに概念的によく知られたインタフェースを提供することができる。

【0054】

ORG401によって定義された組織構造を使用して、ファイルシステム148のセキュリティ情報を伝達することもできる。すなわち、セキュリティテーブル（図示せず）によりアイテムのセキュリティ情報を提供することもできる。デフォルトで、ユーザが所与のアイテムにアクセスすることができる場合には、そのユーザは、デフォルトで所与のアイテムが指示する他の全てのアイテムに、直接または間接的にアクセスすることができる。

【0055】

例えば、アイテムU1（図12）のセキュリティは、ユーザ名=Rossを有するユーザだけが、U1内のファイルにアクセスできるように指示することができる。U1が親になっているアイテムであるApp DataおよびUser Dataに異なるセキュリティ情報が与えられていない場合には、やはりRossだけがそれらのアイテムを閲覧することができる。同様のことが、（U1が親であるUser Dataが親になっている）PrivateとSharedについても当てはまる。アイテムU2のセキュリティは、ユーザ名=Jordanを有するユーザだけがU2内のファイルにアクセスできるように指示することができる。U2が親になっているアイテムであるApp DataおよびUser Dataに異なるセキュリティ情報が与えられていない場合には、やはりJordanだけがそれらのアイテムを閲覧することができる。同様のことが、（U2が親であるUser Dataが親になっている）PrivateとSharedについても当てはまる。アイテムSharedは矛盾したセキュリティ情報を有しているように見えるが、セキュリティ情報は加法的であることが好ましい。したがって、アイテムSharedはU1およびU2の両方からセキュリティ情報を受け取り、「RossとJo

10

20

30

40

50

r d a n だけ」を許可することになる。あるいは、U 1 が「T o m ではなく、R o s s だけ」に相当する許可を有していると想定する。このとき、アイテム S h a r e d は「T o m ではなく、R o s s と J o r d a n だけ」に相当するセキュリティ許可を受けることになる。

【0056】

アイテムが O R G から除去されたとき、セキュリティは除去されたアイテムへの伝達を停止する。例えば、図 9 に示すように、アイテム i 5 が「K e y S l i d e D e c k s」から除去された場合には、i 5 にアクセスする許可がユーザに与えられている場合を除き、もはやユーザは i 5 にアクセスできないことになる。しかし、ファイルシステム 1 4 8 は、デフォルトで、あるクライアントが特にあるアイテムへのアクセスを禁止されていない限り、そのクライアントはすべてのアイテムにアクセスすることができると指定することもできる。

10

【0057】

上述のようにリストもまた名前空間を形成するので、（本明細書に記載のファイル領域、D A G、またはリストとともに動作する機能を備えていないことがある）W i n d o w s（登録商標）X P マシンや他のレガシーマシンのユーザがファイルシステム 1 4 8 に接続した場合は、ユーザはそのリストが、ナビゲートすることができるフォルダに見えることに気付くかもしれない。同様に、ユーザが（レガシーオペレーティングシステム上、またはファイルシステム 1 4 8 によって使用可能なシステム上の）リストのコンテキストに存在する所与のアイテムをダブルクリックしたとき、ファイルシステムは要求元アプリケーションに対して現在のリストによって形成された名前空間を提供する。これ（名前空間）は、アプリケーションに必要なコンテキストを提供する。例えば、ユーザがリスト「T o d o L i s t」を介してアイテム i 2 をオープンしようと試みた場合、ファイルシステムは「T o d o L i s t」を介して形成されるパス、すなわち、\ T o D o L i s t \ i 2 をアプリケーションに返す。したがってユーザが、（例えば、アイテムの新しいコピーを生成するために）アプリケーション内のコマンド「F i l e | S a v e A s . . .」の使用を選択した場合は、アプリケーションはユーザに提示するための正しいコンテキストを得ることができることになる。

20

【0058】

あいまいな名前

上記のように、各アイテムは一意のアイテム名を持つ必要はない。なぜならば、各アイテムはひとえに（本質的に）ファイル I D によって識別されるからであり、ファイル I D はファイルシステム内の各アイテムを一意に識別するからである。したがって、複数のアイテムが同一のファイル名を持つことができる。さらに、アイテムは名前を持つ必要もない。以前のファイルシステムとは異なり、共通の親を持つ 2 つ以上のアイテムでさえ、同一の名前を持つことができる。また、どちらもが全く名前を持たないことも可能である（すなわち、ネームフィールドが空またはヌル）。あるアイテムが名前を持たないか、または同じ親を持つ他のアイテムと共通の名前を持つ場合、そのアイテムはあいまいなファイル名を持つと言われる。しかし、アイテムがあいまいな名前（a m b i g u o u s n a m e）を持つことを許容することにより、ファイルシステムの適切な動作に関する問題が生じることになる。このような問題には、あいまいな名前アイテムを参照することが含まれる。すなわち、クライアント（ユーザまたはアプリケーション）が、どのようにしてそのアイテムを参照するのか、およびあいまいな名前を持つ 2 つ以上のアイテムをどのようにして区別するのが含まれる。

30

40

【0059】

図 1 4 は、同じファイル名を持つ 2 つのアイテム 1 4 0 3、1 4 0 5、ならびにファイル名を持たないアイテム 1 4 0 7 を示す。したがって、これら 3 つのアイテム 1 4 0 3、1 4 0 5、1 4 0 7 の各々はあいまいなファイル名を持つことになる。ファイルシステム 1 4 8 でアイテムが生成されたとき、ファイルシステムは L O C テーブルでそのアイテムに一意のファイル I D を割り当てる。次いでこのファイル I D は、アイテムを参照するた

50

めにファイルシステムによって使用され、またオプションによりクライアントによっても使用される。ユーザクライアントはファイル名によってアイテムを参照する可能性が高く、一方アプリケーションクライアントはファイルIDによってアイテムを参照する可能性が高い。しかし、名前のないアイテムに対しては、ファイルIDがアイテムへの主要な参照になる。

【0060】

図14において、アイテム1403 (ReadMe) および1405 (ReadMe) は、どちらもアイテム1401 (P1) が直接の親となっている。アイテム1403およびアイテム1405は同一のファイル名を有しているが、それらは2つの異なるアイテムであり、かつ異なるデータを格納する。各アイテム1403および1405は、それらの異なるファイルID、00416および00417によって、それぞれ一意に識別することができる。アイテム1407にはファイル名が割り当てられておらず、したがってそのファイルID (00418) によって参照される。例えば、Save Asダイアログボックスによってアイテムを保管するとき、ファイルはファイル名なしで生成することができる。ユーザは、単にエンターキーを打つだけで、名前の入力なしにアイテムを保管することができる。

10

【0061】

重複したファイル名または全くファイル名が無いことを許容することによって、本質的にユーザとレガシーアプリケーションに対しては混乱が生じる可能性がある。リストはフォルダと下位互換性を持つことができるので、レガシーアプリケーションでは重複した名前によって、妥協できない名前の衝突が引き起こされる可能性がある。したがって、本発明の様々な態様により、ファイルシステム148は、他の方法ではあいまいになる所望のアイテムをクライアントによって識別できるようにするユーザビリティ機能 (便利で使い勝手の良い機能) を提供することができる。

20

【0062】

この例では、ファイルシステム148は、Win32アプリケーションなどのレガシーシステムとの下位交換性を提供し、一意のファイル名を提供する。Win32アプリケーションに対する下位交換性が必要なのは、Win32アプリケーションが、一意の名前空間が存在するとの想定の下に設計されているからである。したがって、Win32アプリケーションでは、アイテムをオープンするためにファイル名が必要になる。Win32アプリケーションとの下位交換性を提供するために、ファイルシステム148は、ファイルシステム148がレガシーアプリケーションに提供する一意のファイル名を自動的に生成することによって、ファイル名のあいまいさを無くする。ファイルシステム148はまた一意のファイル名へのパスも生成する。例えば、アイテム1403と1405に対するパスは、単に\ P1である。パスと名前を併せて、アイテムの完全に限定された名前 (qualified name) が構成される。したがって、アイテム1403の完全に限定されたあいまいな名前は、\ P1 \ ReadMeとなる。アイテム1405はアイテム1403とは異なるアイテムであるが、アイテム1405の完全に限定されたあいまいな名前もまた、\ P1 \ ReadMeとなる。両方のアイテムが同一の完全に限定されたパス (qualified path) を有することになるので、ファイルシステム148は、以下で説明するような方法でそれらのあいまいさを無くする。

30

40

【0063】

ファイル名それ自体があいまいなために、ファイルシステム148がファイル名のあいまいさを無くすることが望まれる状況には、少なくとも2つのシナリオが存在する。第1のシナリオは、共通の親の下で複数のアイテムが同一のファイル名 (例えば、上記の例のReadMeなど) を持つ場合である。第2のシナリオは、アイテムが全く名前を持たない場合である。どちらのシナリオでも、レガシーアプリケーションに提供するために、一意の完全に限定されたファイル名が生成される。本発明の例示的態様によれば、ファイルシステム148はアイテム名として単にファイルIDだけを使用することができる。例えば、ファイルシステム148は、アイテム1403を完全に限定された名前、\ P1 \ 0

50

0416で識別し、アイテム1405を完全に限定された名前、\P1\00417で識別し、アイテム1407を完全に限定された名前、\P1\00418で識別することができる。しかし、ユーザがアイテムの内容を認識するために、しばしば名前を利用するので、数値ネームの有用性は限られる。さらに、アプリケーションはMRU (Most Recently Used: 直前に使用された) リスト、例えばアプリケーションのファイルメニューに最近オープンしたアイテムの名前を表示することも多い。ユーザがMRU リスト中の数値ネームの表示を見て困惑しないとも限らないので、たとえ可能であり、本発明の範囲に含まれると見なされるとしても、この選択肢は好ましくない。

【0064】

その代わりとして、共通の親の下で複数のアイテムが重複した名前を有するシナリオでは、ファイルシステム148は重複したファイル名を基にして、そのファイル名に人間可読の(必須ではないが、好ましくは小さな値の)整数を挿入することによって、一意のファイル名を生成する。例えば、アイテム1403には、一意のファイル名「Readme1」を与え、アイテム1405には一意の名前「ReadMe2」を与えることができる。アイテムが全くファイル名を持たない場合には、一意のファイル名はファイルシステム148によって提供される識別用整数になる。したがって、アイテム1407には一意のファイル名「3」を与えることができる。ファイルシステムは、ファイル名のあいまいさを無くするために使用された整数を再利用しないことが好ましい。ファイルシステム148は、ファイルシステムは、特定のアイテムを参照するたびに、一貫して同一のあいまい性除去用の整数を使用するように、将来の参照のために一意のファイル名情報を保管する。ファイルシステム148は、単にファイルIDとともに使用された整数だけを保管し追跡することもできるし、一意の名前全体を保管することもできる。このようにして、レガシーアプリケーションは、アイテムを作成するかまたは最後に修正したクライアントによって最初に提供されたファイル名に非常に近い名前を表示し確認することになる。また、表示される名前空間は、レガシーアプリケーションを使用するとき、またはレガシーオペレーティングシステムからファイルシステム148にアクセスするときには、ユーザにとって見覚えのあるものになる。本発明の代替態様によれば、このファイルシステムは、あいまい性除去用の文字、記号、英数字、短い英数字列、または他の視覚的識別子を使用することができる。

【0065】

ファイルシステム148は、一意のファイル名を生成するたびに一意のあいまい性除去用の整数を使用するので、ファイルシステム148は、アイテムの完全に限定された名前に加えて、ファイル領域のルートから、各あいまい性が除去されたアイテムへの一意のパスを生成することができる。例えば、アイテム1403はそのアイテムの完全に限定され、あいまい性が除去された名前、\P1\ReadMe1、またはそのアイテムのあいまい性が除去されたルートパス、\ReadMe1によって参照することができる。同様に、アイテム1405は、\P1\ReadMe2、または\ReadMe2によって参照することができる。クライアントがアイテムを見つけ出すための十分なコンテキストを有していない状況(例えば、クライアントがリストをブラウズ(閲覧)するのではなく、照会によってアイテムを見つける状況)では、追加のパスを使用することによりアイテムへの単純なパスを提供する。追加のパスはまた、アイテムの不揮発性の名前を提供する。例えば、ユーザがP1を経由してReadMe1をオープンする場合、オープン動作を実施するアプリケーションは、そのMRUリストに保管された\P1\ReadMe1を使用する。しかしその後、ユーザがファイルシステム148の組織構造からアイテム1403を除去した場合には、上述のように、\P1\ReadMe1パスはもはや有効ではなく、オープン動作に失敗する。しかし、オープン動作の失敗を検出したときに、アプリケーションは、ファイル領域のルート(最上位のディレクトリ)からのパスを使用してファイルシステムにアイテムを照会することができ、それは成功することとなる。

【0066】

本発明の一態様によれば、すべてのアイテムは、そのアイテムがもともとあいまいな名

前を持っていたかどうかには関係なく、完全に制限されたパスの他に、ルートパスを持つことができる。例えば、編成テーブルはルートから各アイテムへの隠し保持リンク（hidden holding link）を格納することができる。このようにして、アプリケーションはルートから任意のアイテムを参照することができる。さらに、ルートから各アイテムに保持リンクを備えることにより、セキュリティ許可を伝達するために、組織構造を使用する実施形態において、基本レベルのセキュリティを提供する。また、すべてのアイテムに対するリンクのすべてを含む全体のセット（集合）を形成することによって、一意のあいまい性除去用の整数を容易に決定できるようになる。次いでファイルシステムは、例えば、各アイテムに、そのアイテムを参照するために使用できる一意の整数を割り当てることによって、ファイルシステムのルートからアドレス可能な全アイテムのあいまいさを無くすることができる。

10

【0067】

あるアイテムのあいまい性除去用の整数または他の視覚的識別子は、そのアイテムがその後あいまいでない名前にリネーム（改名）された場合でも不変であることが好ましい。したがって、アプリケーションは、引き続きそのアイテムに割り当てられたあいまい性除去用の整数を用いてアイテムを要求し、オープンすることができるが、このことは現在既知のファイルシステムでは不可能である。例えば、本明細書に記載のあいまい性除去用の整数の機能を認識しているアプリケーションファイルシステムのクライアントは、最近使用されたアイテムのあいまい性除去用の整数情報をそれ自体で追跡し、格納することができる。ユーザが最近使用したアイテムをオープンするように要求したとき、そのアプリケーションクライアントは、ファイルシステムの組織構造内のそのアイテムの位置に関わらず、特定のあいまい性除去用の整数に対応するアイテムだけをファイルシステムに要求することができる。

20

【0068】

本発明の一典型的態様によれば、ファイルシステムに格納されたアイテムが名前を持っていない場合には、ファイルシステム、オペレーティングシステム、および／またはアプリケーションは、アイテムの名前以外の情報を使用してユーザにファイルを提示することができる。例えば、デジタルカメラによって生成されたデジタル写真の場合には、一般にデジタルカメラは順次的な数値ファイル名を生成するので、ファイル名はほとんど使用されていない。本発明の諸態様を使用することにより、デジタルカメラがファイル名を提供する必要は全く無くなる。ファイルシステム、オペレーティングシステム、またはアプリケーションは、ユーザにファイルをサムネイル形式で提示し、ユーザはそこから所望のアイテムを選択してさらに使用することができる。

30

【0069】

本発明を、現時点で本発明を実施する好ましいモードである特定の具体例に即して説明してきたが、当業者ならば、上述のシステムおよび技術に関する多数の変形形態および代替形態が存在することを理解されるはずである。したがって、本発明の趣旨および範囲については、添付の特許請求の範囲に記載されているように広く解釈すべきである。

【図面の簡単な説明】

【0070】

40

【図1】本発明の一例示的实施形態によるメディアユーザインタフェースの実装に適する一般的な動作環境を示す図である。

【図2】既知のファイルシステムの一般的な組織構造を示す図である。

【図3】本発明の一例示的实施形態によるロケーションテーブル（LOC）を示す図である。

【図4】本発明の一例示的实施形態による編成テーブル（ORG）を示す図である。

【図5】図4に示された編成テーブルによる非循環有向グラフ（DAG）組織構造を示す図である。

【図6】本発明の一例示的实施形態によるファイルシステムのブロック図である。

【図7】本発明の一例示的实施形態によるファイル領域を示す図である。

50

【図 8】 図 7 の組織構造を定義する編成テーブルを示す図である。

【図 9】 本発明の一例示的实施形態によるファイル領域を示す図である。

【図 10】 図 9 の組織構造を定義する編成テーブルを示す図である。

【図 11】 本発明の一例示的实施形態による編成テーブルを示す図である。

【図 12】 図 11 に示された編成テーブルによるファイル領域を示す図である。

【図 13】 本発明の一例示的实施形態による、ファイルシステムに格納されたアイテムを管理する方法を示す流れ図である。

【図 14】 本発明の一例示的实施形態によるファイルシステムの一部を示す概念構成図である。

【符号の説明】

10

【0071】

- 120 処理ユニット
- 121 システムバス
- 130 システムメモリ
- 131 ROM
- 132 RAM
- 133 BIOS
- 134 オペレーティングシステム
- 135 アプリケーションプログラム
- 136 他のプログラムモジュール
- 137 プログラムデータ
- 138 ファイルシステム
- 140 取り外し不能揮発メモリインタフェース
- 144 オペレーティングシステム
- 145 アプリケーションプログラム
- 146 他のモジュール
- 147 プログラムデータ
- 148 ファイルシステム
- 150 取り外し可能揮発メモリインタフェース
- 160 ユーザ入力インタフェース
- 161 マウス
- 162 キーボード
- 165 デジタイザ
- 170 ネットワークインタフェース
- 171 ローカルエリアネットワーク
- 172 モデム
- 173 広域ネットワーク
- 180 リモートコンピュータ
- 185 アプリケーションプログラム
- 190 ビデオインタフェース
- 191 モニタ
- 195 出力周辺機器インタフェース
- 196 プリンタ
- 197 スピーカ
- 301 LOC テーブル
- 303 ファイル ID
- 305 ファイル名
- 307 ボリューム
- 309 ロケーション
- 311 サイズ

20

30

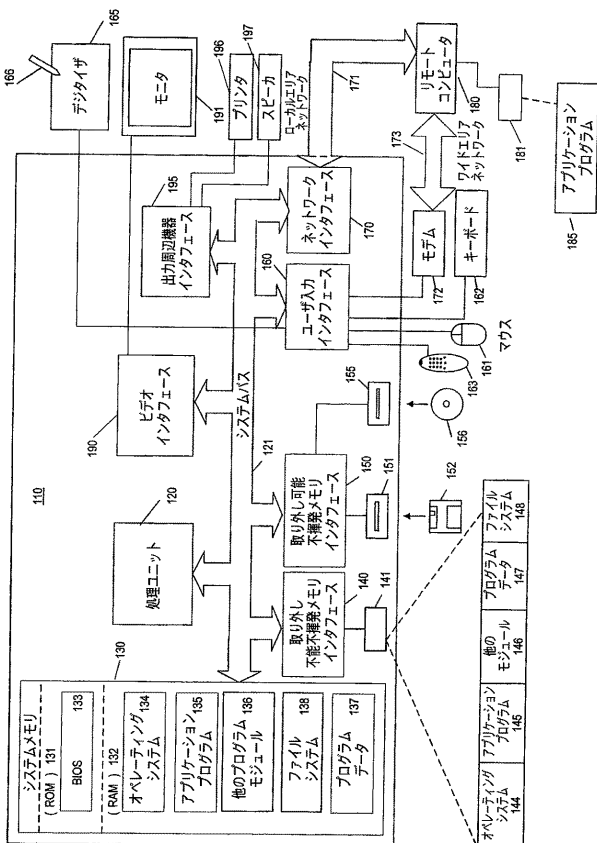
40

50

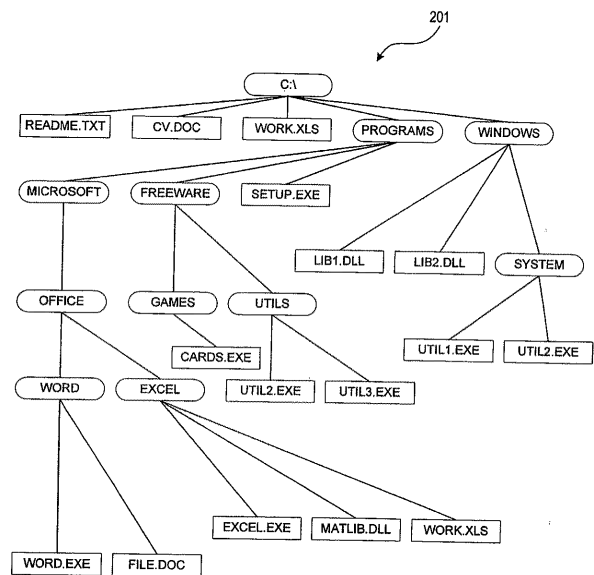
401 ORGテーブル（編成テーブル）
 403 親
 405 子
 601 ファイルシステムマネージャ
 603 ADD_Item API（アプリケーションプログラムインタフェース）
 605 UPDATE_Item API
 607 DELETE_Item API
 609 QUERY_Item API
 1401 P1（ファイルID 415）
 1403 ReadMe（ファイルID 416）
 1405 ReadMe（ファイルID 417）
 1407 （ファイルID 418）

10

【図1】



【図2】



従来技術

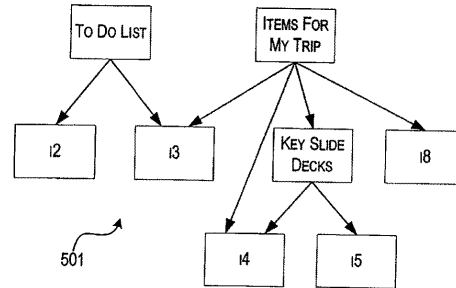
【図 3】

303 ファイルID	305 ファイル名	307 ボリューム	309 ロケーション	311 サイズ
000001	To Do List	C	19836	2
000002	I2	C	19838	20
000003	I3	C	19858	14
000004	Items For My Trip	C	02864	3
000005	I4	D	34884	32
000006	I5	C	07893	18
000007	Key Slide Decks	D	02389	2
000008	I8	D	00034	97
000009	I9	C	00625	23
...	...	...	...	...
00085	I1	C	08762	13
...	...	...	...	...
00135	I2	D	09238	27
...	...	...	...	...
00245	L1	C	04874	4
00246	I7	L	07629	45
...	...	...	...	...
00301	\	C	05672	1
00302	Users	C	05673	1
00303	Machine	C	05674	1
00304	U1	C	05675	1
00305	U2	C	05676	1
00306	App Data	C	05677	1
00307	User Data	C	05678	1
00308	App Data	C	05679	1
00309	User Data	C	05680	1
00310	Private	C	05681	1
00311	Private	C	05682	1
00312	Shared	C	05683	1
...	...	...	...	...
00415	P1	D	00324	1
00416	Readme.txt	D	00325	24
00417	Readme.txt	C	07124	36
00418		C	12976	100

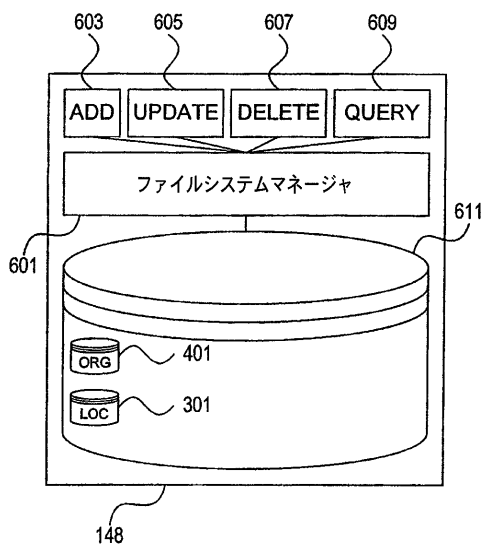
【図 4】

403 親	405 子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000007	000006

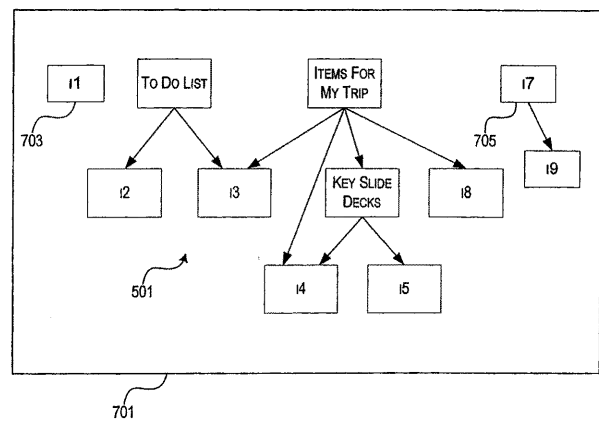
【図 5】



【図 6】



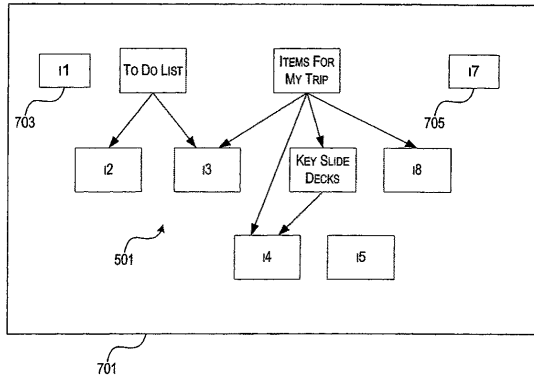
【図 7】



【図 8】

801 親	子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000007	000006
000246	000009

【図 9】



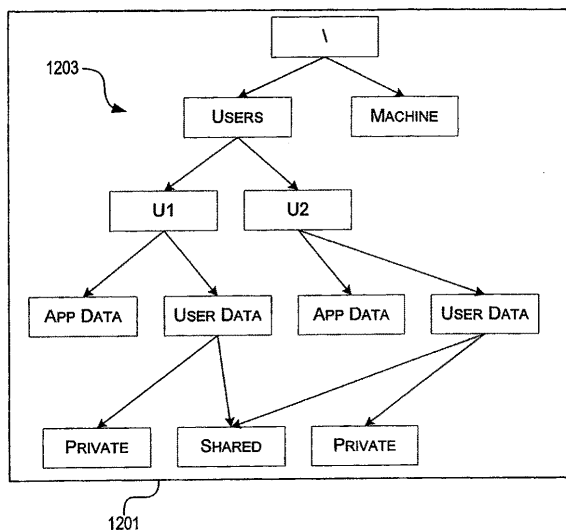
【図 11】

親	子
00301	00302
00301	00303
00302	00304
00302	00305
00304	00306
00304	00307
00305	00308
00305	00309
00307	00310
00307	00312
00309	00311
00309	00312

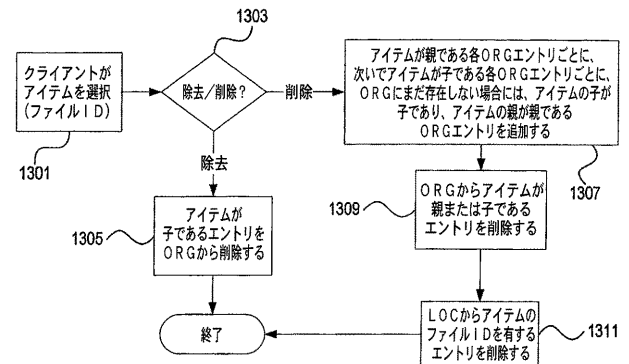
【図 10】

親	子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000246	000009

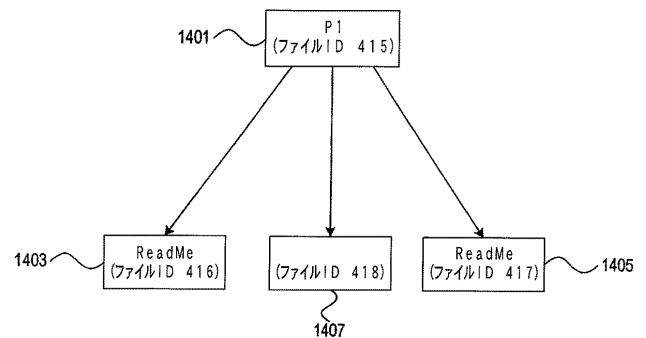
【図 12】



【図 13】



【図 14】



フロントページの続き

- (72)発明者 アニル ケー ノリ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 バラン セトゥ ラマン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 ティモシー ピー. マッキー
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 ウォルター アール. スミス
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内

Fターム(参考) 5B082 EA09