

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3993227号

(P3993227)

(45) 発行日 平成19年10月17日(2007.10.17)

(24) 登録日 平成19年8月3日(2007.8.3)

(51) Int. Cl.		F I		
<b>G06F 12/00</b>	<b>(2006.01)</b>	G06F 12/00	5 1 1	
<b>G06F 17/21</b>	<b>(2006.01)</b>	G06F 17/21		

請求項の数 1 (全 21 頁)

<p>(21) 出願番号 特願平6-515469  (86) (22) 出願日 平成5年12月27日(1993.12.27)  (65) 公表番号 特表平8-505723  (43) 公表日 平成8年6月18日(1996.6.18)  (86) 国際出願番号 PCT/US1993/012585  (87) 国際公開番号 W01994/015307  (87) 国際公開日 平成6年7月7日(1994.7.7)  審査請求日 平成12年11月27日(2000.11.27)  審判番号 不服2005-25157(P2005-25157/J1)  審判請求日 平成17年12月27日(2005.12.27)  (31) 優先権主張番号 07/997,688  (32) 優先日 平成4年12月28日(1992.12.28)  (33) 優先権主張国 米国(US)</p>	<p>(73) 特許権者 503260918  アップル インコーポレイテッド  アメリカ合衆国 95014 カリフォル  ニア州 クパチーノ インフィニット ル  ープ 1  (74) 代理人 100086531  弁理士 澤田 俊夫  (72) 発明者 アシュ、ディラン、ビー、  アメリカ合衆国、94087 カリフォル  ニア州、サニーバール、イースト エル  カミーノ ドライブ 870  (72) 発明者 クレジック、ニック、ジー、  アメリカ合衆国、95014 カリフォル  ニア州、カパーティノ、パトリオット ウ  エイ 21103</p>
--	---

最終頁に続く

(54) 【発明の名称】 ファイル翻訳システム

(57) 【特許請求の範囲】

【請求項1】

アプリケーションプログラムと変換プログラムとを実行可能に含むシステムであって、  
(a) 各アプリケーションプログラムが開くことができるファイルを識別する処理手段と、  
(b) 変換プログラムが変換できるファイルフォーマットを識別する処理手段と、  
(c) 選択された文書を作成した作成元のアプリケーションプログラムがコンピュータシステムによりアクセスできないことを判定する手段と、  
(d) 上記判定する手段が、上記作成元のアプリケーションプログラムがコンピュータシステムによりアクセスできないと判定したときに、当該選択された文書に対して開くことができることと上記(a)の処理手段の識別結果に基づいて判別されたアプリケーションプログラムであって、上記作成元のアプリケーションプログラム以外のアプリケーションプログラム、および、変換できると上記(b)の処理手段の識別結果に基づいて判別された変換プログラムのリストを作成する処理手段と、  
(e) 該識別されたアプリケーションプログラムと変換プログラムのリストをユーザに表示する処理手段と、  
(f) 上記選択された文書を、対応する変換プログラムで変換して当該文書を開く、上記作成元のアプリケーションプログラム以外の、アプリケーションプログラムに引き渡す処理手段と、  
含むことを特徴とするシステム。

10

20

## 【発明の詳細な説明】

### 発明の分野

本発明はコンピュータ用のファイリングシステムの改良に関する。より詳細には、異なるファイルシステム間における文書交換の簡素化に関する。

### 発明の背景

ワークステーションのソフトウェアのデベロッパにとって、ユーザーのインターフェイスにおける一貫性を維持しつつフレキシブルなソフトウェア環境を提供することがますます重要になっている。この種の操作環境を提供しようとする初期の試みについてはHernandezらに付与された米国特許第4,686,522号に記載がある。この特許はユーザーがカーソル位置に動的メニューを呼び出し、そのメニューからさまざまなファクションのどんなものでも呼び出すことができる図表と文章とを組合せた処理システムについて記載している。この種のユーザーとの自然な相互作用は、ユーザーのインターフェイスを改良し、アプリケーションを格段に直感的なものにする。

WORDのワードプロセッシングプログラム及びMacWriteのワードプロセッシングプログラムは現在のところ、文書変換の際に使用できる翻訳プログラムについて静止リストを提供している。しかし追加的に変換ルーチンが作られるのにつれて静止リストに動的に追加していきける能力はない。また、それがなければオペレーティングシステムに認識不能なファイルの動的オープニングを提供する解決法のあるシステムは未だ存在しない。今日の技術水準は、ファイルを選択した後、Macintosh[商標]コンピュータ、その他のコンピュータシステムにオープニングを開始するようファイルを挿入しても、そのオープニングシステムが、その特定ファイルの独自の性質を認識するよう前以てプログラムされていないと、あるいはそのファイル取扱い用のアプリケーションがインストールされていないと、処理できない旨を標準指示メッセージ[a standard prompt]がユーザーに表示するようなものすぎない。

### 発明の概要

そこで本発明の第1の目的は、ローカルエリアネットワークのような通信ラインにおいて遠距離及び近距離ともにコンピュータに接続されるディスクに記録された代替プログラムや翻訳ソフトウェアをサーチし、文書を開くとか変換するとかの二者択一的選択からユーザーが選択できるようにする文書変換能力を提供することにある。本発明は人間工学的な方法で1つ又は2つ以上の文書を新たなフォーマットに変換するオプションをユーザーに提供するものである。本発明はまた、ユーザーのファイルに対する視点に、Disk Operating System(DOS)、WindowsあるいはUNIXといった自システム以外のオペレーティングシステム(OS)で作られた文書を含むどんな種類の文書やプログラムについても新たな情報を供給するものでもある。

### 【図面の簡単な説明】

図1は本発明のパーソナルコンピュータシステムのブロックダイヤグラムである。

図2は先行技術のトランスレータを示す。

図3はファイルをオープンすることに失敗したことを表示している先行技術のディスプレイの1例である。

図4 Aは先行技術のファイル情報のディスプレイ例を示す。

図4 Bは本発明のディスプレイ例を示す。

図5 Aはオープンオペレーションの先行技術のMacintoshのディスプレイ例を示す。

図5 Bは本発明のOpenメニューの1例である。

図6はユーザーが本発明に従ってファイルをオープンしたとき動的に示されるディスプレイである。

図7は本発明に従い文書の翻訳の進行度合を示すインジケータである。

図8は本発明のFileメニューの1例である。

図9は本発明の文書翻訳ディスプレイの1例である。

図10は本発明に従い文書アイコンが引き出されプログラムアイコン上にドロップしている操作を示す。

10

20

30

40

50

図 1 1 A 及び図 1 1 B は本発明に係るフローチャートの詳細である。

図 1 2 A ~ 図 1 2 G は、本発明に係るフローチャートの詳細である。

図 1 3 は本発明のブロックダイヤグラムである。

#### 発明の詳細な説明

本発明は、好ましくはIBMのPS/2コンピュータ、またはAppleのMacintoshコンピュータのようなパーソナルコンピュータに既に備わっているオペレーティングシステムのコンテキスト中で実施されるのがよい。代表的なハードウェア環境を図 1 に示す。すなわち図 1 は、従来のマイクロプロセッサのような中央処理装置(CPU) 10、ならびにシステムバス12経由で相互接続されたその他多数の装置がある本発明のワークステーションの典型的なハードウェア構成を示している。図 1 のワークステーションには、ランダムアクセスメモリー(RAM) 14、リードオンリーメモリー(ROM) 16、磁気ディスク装置20のような端末機に接続されるI/Oアダプタ18、キーボード24に接続されるユーザーインターフェイスアダプタ22、マウス26、スピーカ28、マイクロホン32、及び/又は例えばバスに対するタッチスクリーン装置(図示せず)のようなその他のユーザーインターフェイスデバイス、ワークステーションをデータ処理網に接続する通信アダプタ34、バスをディスプレイ装置38に接続するディスプレイアダプタ36などがある。このワークステーションにはApple System/7[登録商標]のようなOSが既に装備されている。

10

図 2 はユーザーがプログラムを呼出し文書を使用可能なフォーマットに変換するため該当する翻訳ルーチンを積極的に選択しなければならない先行技術の翻訳プログラムの 1 例である。図 3 ではオープンファイルに失敗したときの代表的なMacintoshメッセージを示している。

20

本発明の好ましい実施例は、コンピュータが、それを作り出したプログラムにアクセスを有していないときでも文書をオープンできるシステムソフトウェアの拡張として実現されている。好ましくは本発明は文書をオープンしなくても文書を別のプログラムのフォーマットに変換する。

本発明は他のMacintoshのタスクと同じ様に容易に、このような文書をオープンし、別の文書フォーマットに変換する。それは文書をオープンすることができるプログラムを見つけだし、ユーザーの選択に供するためそれらのプログラムをリスト表示することができる機能をシステムに供給するものである。そうすればユーザーはそれ自身の内にある翻訳能力を使って文書を変換することができるプログラム、または特別の翻訳ソフトウェアを使うプログラムを選択することができる。

30

本発明は変換処理を準備し、変換を行う翻訳ソフトウェアを呼び出す。本発明は文書及びプログラムに関する情報を集め、最適のオプションを選択しリストアップさせるためにエキスパートシステムを使い、ユーザー選択を記録し、そしてユーザーに前に行った選択を思い出させる。本発明はまた、特定の具体的な文書にとって最も好ましいものを決定させるため、ユーザーがオプションから試行することを可能にする。こうした能力は文書に関する情報をより多く提供するもので、ひいてはユーザーをしてパーソナルコンピュータで利用可能なより広範な種類の文書、例えば、Other Equipment Manufactures(OEM)、Personal Computer(PC)のようなその他のコンピュータで作られシステムや文書に現時点

40

で実行中でないプログラムにより作成された文書などに当たってみるとい、より大きいフレキシビリティを与えるものである。図 4 A はファイル名 400 ならびにそれらのサイズ 410、種類 420 のディスプレイの 1 先行技術例である。図 4 B は本発明のディスプレイの 1 例である。440 の種類欄は先行技術のディスプレイより広い記述情報を有している。

ユーザーがディスクやフォルダの内容を名前、サイズ、種類あるいは日付につき見ることを選択すると、本発明でなければ見ることができない詳細な情報を見ることができる。図 5 A は先行技術のファイルビューの 1 例である。これと対照的に本発明では、アプリケーションプログラムで使われたときオープンすることができる文書全部のリストを示す。ユーザーがプログラムのFileメニューからOpenメニューを選択すると、そのプログラムでオ

50

ーブンすることができる文書より多くの文書がディスプレイされ得る。図5Bは本発明のOpenメニューを図示したものである。

典型的な翻訳拡張である翻訳拡張ファイルの例としては、アプリケーション拡張ファイル用のXTNDがある。これらのファイルは本発明をClarisのプログラム及びファイル翻訳ソフトウェアに結合させるように設計されている。RTF/スタイル拡張ファイルは、Microsoft WordとMacWrite II間のような一定のプログラム間で複写や切張りをするとき材料を翻訳するのに要求される追加ファイルである。

本発明は、文書をオープンし、文書間で切張りし、キーボード省略[keyboard shortcuts]を使うなどするとき、典型的なMacintoshのツール全てを採用するものである。また本発明はユーザーのために次のことを可能にする。すなわち、

(1) 文書をオープンし変換する際に使われるべきプログラム及び翻訳ソフトウェアを選択すること、

(2) 高速変換用の特別な文書コンバータにすること、

(3) 特定のユーザーの好みにマッチするように本発明を改変すること。

文書オープンの方法：

Apple Macintoshコンピュータに採用されている典型的なプロシージャは本発明の好ましい実施例に従い下記のステップに挙げてある。当業者であれば、特許請求された本発明の技術範囲から離れることなく、これら以外のステップでも代替できることを了解するであろう。

1. Macintosh Easy Open(ME0)ディスクウインドウ中のOpen Meアイコンを2度スイッチ[double-click]させる。Open MeはMacWriteバージョン5.0文書である。当業者ならOpen Meが文書の例示名称であって、それ以外の文書およびこれに関連する名称も同様に取扱うことができることを容易に理解するであろう。

2. リストを見直し、貴方の翻訳パフォーマンスを選択するためスイッチを入れる[click]。ユーザーは同時に1個の選択しか選択しない。ユーザーに見えるチョイスはその特定のシステムにインストールされているプログラムおよび翻訳ソフトウェアの種類によって異なる。よく起こることは、翻訳拡張を含むチョイスは最良の結果をもたらすが、翻訳は必要に応じて交換できることである。図6はユーザーが本発明に従ってファイルをオープンしたとき、動的に提示されるディスプレイである。600で本発明はどのプログラムがその文書を作ったのかを示す。この場合、“MacWrite”ラベル610はMacWrite IIがこの文書をそれ自身でオープンすることができること、あるいはラベル620でMacWrite IはXTND翻訳拡張ファイルと共にワークすることができることを示している。最後にラベル630で、選択ボックスはオプションが推薦されたチョイスだけしか示さないために存在していることを示している。

3. オープンボタンを入れよ[click]。進行度合インジケータは文書が翻訳されていく度合を示す。図7は本発明に従い文書が翻訳されていく進行度合インジケータを示している。ラベル700は使用中の翻訳拡張を明確に表している。ラベル710は現時点に行われている翻訳の進行度合を示している。ラベル720は進行中の翻訳を中止するための選択領域である。

4. 文書をオープンしたら、その文書が正しくフォーマットされているかチェックせよ。本発明は文書が新しいプログラムフォーマットに変換されていることを示すため文書を再命名する。

5. 変換されたコピーを保管するため、FileメニューからSave As\_\_を選択し、貴方のチョイスの場所にそのコピーを置け。図8は本発明のFileメニューを示す。ラベル800で、ユーザーがチョイスのファイル名を保管できるように上記Save Asのオプションが選択されている。Fileメニューがラベル810で選択されたことが示されている。もし翻訳の結果が満足のいくものでないときは、別のオプションがある。すなわち、

・もし貴方がこのチョイスを欲していないことが明白なら、新しい文書を保管することなく貴方はプログラムを中止することができ、別のチョイスをするためOpen Meアイコンを再び2度スイッチすることができる。そして、

10

20

30

40

50

・もし貴方が結果を比較してみたいなら、変換された文書を保管し、別のチョイスでオリジナル文書を再びオープンすることができる。本発明は各種文書のため貴方がつくった最も新しい優先度を記憶していて、次回に同じような種類の文書が選択されたとき、その最も新しい優先度がチョイスのリスト中にハイライト表示される。最適の結果を生み出すチョイスがないときは、本発明はオリジナルのフォーマット形成をしないで単にテキストとして文書をオープンし得るチョイスをディスプレイする。「推薦されたチョイスだけを示せ」というオプションはない。図9は本発明に従った文書翻訳ディスプレイを図示したものである。ラベル900は推薦されたチョイスだけがディスプレイされているなら選択することができるチェックボックスである。ラベル910は、もし貴方が前に同様な文書をオープンしようとしたのなら、本発明はリスト中の貴方の最も新しい優先度を選択することを示している。ライン920以下にリストされているチョイスは最適結果を生み出すとは限らない。推薦されたチョイスだけしか選べないのなら、本発明は全部のチョイスをディスプレイする。

文書アイコンを貴方のシステムに追加されたプログラムアイコンに引き当てる[drag to]ことができ、1または2以上の文書を自動的にオープンすることができる。それは文書アイコンを図10に示すようにその文書をオープンする能力のあるプログラムのアイコンに引き当てることによって行われる。もしそのプログラムが文書をオープンすることができるなら、そのアイコンがハイライト表示される。もしプログラムが貴方の引き出した文書をオープンすることができるなら、図6に示すのと同様に本発明は利用可能なチョイスのリストを提示する。このリストは特定のアプリケーションプログラムや利用可能な翻訳ソフトウェアを含むチョイスに限定される。プログラム同士を複写したり切張りすることは、情報が切張りされた文書中にある受容可能なフォーマット中に複写された材料の翻訳を行うよう本発明によって可能にされている。

文書コンバータ：

文書コンバータは文書を特定のプログラムのフォーマットにその文書をオープンすることなく変換するために設置される便利なシステムユーティリティである。ひとたび文書コンバータが特定のタイプの変換のために初期化されると、文書はそれらのアイコンをコンバータのアイコンに引き出すことによって容易かつ迅速に変換することができる。特定の文書コンバータのアイコンは各種文書のために設置することができる。

文書コンバータを設置するには次のステップに従う。

- 1．貴方のハードディスクからDocument Converterを選択し、FileメニューからDuplicateを選びなさい。
- 2．文書コンバータのコピーをオープンしなさい。
- 3．Setボタンを入れなさい。
- 4．Setボタンを入れなさい。

本発明は新しい文書コンバータ名を付与する。この名前はユーザーの判断で変更することができる。

文書の変換：

文書コンバータが正しく初期化されていれば使用は簡単である。文書を変換するには、Open Meアイコンを文書コンバータアイコンに引き出し、マウスボタンを解放する。両アイコンともハイライトされて変換が始まったことを知らせる。本発明は文書のコピーを作り、文書に(変換された)名前を付与し、オリジナルをそのまま残す。1グループの文書を新しいフォーマットに変換するには、アイコンを選択し、それらを初期化されている文書コンバータに引き出す。

本発明のオプションの初期化：

本発明のカスタム化のためのたくさんのオプションがある。

オプション

オン/オフ：

文書を翻訳する必要がないときは翻訳機能を切ることができる。しかし本発明のシステムソフトウェアがスタートアップ[startup]ディスクに備わっている限り、向上された情報

10

20

30

40

50

がFinder及びファイルのリストビューになお利用可能であり、文書コンバータもなお使用することができる。FinderについてはAddison-Wesley Publishing Co.から1985年～91年に発行された「Inside Macintosh」第1～6巻に詳述されている。

常時チョイスを表示する：

Always Show Choicesを選択すると、たとえ優先選択[preference]が選択されているとしても、特定の種類の文書をオープンするためのチョイスのリストを提示する。このオプションが解除[deselect]されると、本発明はかつて指示してあった優先選択を特に確認することなく再作動させる。

サーバーからチョイスを算入する：

本発明は、たとえディスクトップで利用可能にされているファイルサーバーをも含めて、ディスク上のあらゆる利用可能なプログラムをサーチするように設置されているのが通常である。ネットワークを介して遠隔のディスクにあるプログラムを使うには長時間が要求されることがある。ハードディスクにしかないプログラムのシステムサーチを、そのシステムに直接接続するため、この制御を解除するようスイッチする。

10

優先選択の削除：

特定の翻訳優先選択用に選ばれた本発明のチョイスのレコードを消すには、Delete Preferencesボタンを入れる。例えば新たな翻訳ソフトウェアが利用可能なときは、従前のチョイス全てを即座に削除し、そのシステムを各々の新しいチョイスに駆り立てることが便利であろう。

動的な翻訳リスト：

20

オペレーティングシステムがユーザーにファイルをオープンするための標準的ユーザーインターフェイスを提供しているコンピュータでは、本発明はインストールされている外部のトランスレータファイルに基づきアプリケーションが現実にオープンすることができるファイルフォーマットをオペレーティングシステムに供給する。本発明は後述するように、新しい[current]能力を発揮するためアプリケーションプログラムによりオンザフライに操作することができるファイルのリストを構築する。

ロジックの詳述

動的なアプリケーションオープン能力：

特定のプログラムにどんなフォーマットの文書を決定するかシステム及び方法は動的な方法で決定され、図11A及び図11Bに示すロジックでユーザーに相互作用で提供される。処理はロジックの概括的一覧が提示される機能ブロック1100から始まる。ロジックは特定のプログラムがオープンでき起動できるファイルフォーマットを見つけ出す。次に機能ブロック1110で、アプリケーションが常にオープンすることができるフォーマットで始めることによって処理が開始される。この情報は殆ど全てのアプリケーションプログラムの所定位置に配置され、好ましくはその後、最初のDynamic Application Open Capability(DAOC)が機能ブロック1120及び1130に示すように標準的なSystem 7オペレーションを介してアプリケーションリストを得ることによって開始される。

30

機能ブロック1130に関するロジックの詳述は図11Bに示されている。処理は、外部の全てのトランスレータファイルが同定される機能ブロック1131で開始する。すると機能ブロック1132に示された第1の外部のトランスレータファイルで開始して、第1トランスレータファイルが読み出すことのできる抽出ファイルフォーマットが機能ブロック1133で同定される。次に、機能ブロック1135で、前記ファイルトランスレータファイルに対応するファイルフォーマットがリストのソース部に付加される。次に何らかの追加的トランスレータファイルが利用可能かどうかを決定する決定ブロック1136でテストが行われる。もし利用可能なら、次のファイルが機能ブロック1134で選択され、トランスレータファイルが処理することができるファイルフォーマットが機能ブロック1133で決定される。もうファイルが残っていないなら、これら外部のトランスレータを使うアプリケーションがリストの宛て先部分に付加され、処理が図11Aの決定ブロック1140にターミナル1138から戻る。

40

プログラムがリストの宛先箇所に存在しているか否かを決定するため決定ブロック114

50

0でテストが行われる。プログラムが存在しているなら、機能ブロック1150で全てのフォーマットがオープンできるプログラムのリストに付加される。次に、ほかに処理すべきDAOCが残っていないかを判断するため処理が決定ブロック1170に移される。それ以上処理の必要がないのであれば、オープンできるファイルの完全なリストがターミナル1180でユーザーにリターンされる。もし処理にさらに別のファイルが利用可能なら、次のDAOCが機能ブロック1160で選択され、リストを次のDAOCで更新するため制御が機能ブロック1130に移される。

図12A~Gは本発明の翻訳ダイアログのロジックの詳細のフローチャートである。翻訳ダイアログは翻訳の可能性をコンピュータのユーザーに提供するユーザーインターフェイス部分である。可能性はユーザーが翻訳処理に関する知識をあまり要求せず、翻訳は通常 10  
のファイル処理オペレーションと同一方式で行われるという前提に基づいている。ユーザーが文書に二回スイッチを入れたことをシステムが感知すると、処理が機能ブロック1200で始まる。

機能ブロック1200で二回スイッチされた文書を作成したアプリケーションがシステムにインストールされているか否かにつき決定ブロック1202でテストされる。もしアプリケーションがシステム中に存在するなら、その文書が機能ブロック1212に示されるようにオープンされる。もしアプリケーションがシステム中に存在しないなら、本発明が文書をオープンすることができるか否かを決定するため別のテストが決定ブロック1204で行われる。もしオープンできないなら、その旨のメッセージが機能ブロック1214 20  
でユーザーに示される。しかし通常そうなのだが、もし文書が本発明を使ってオープンできるなら、機能ブロック1206に示されるように、また図12B~図12Gに詳細に示されるように、本発明の翻訳ダイアログが文書に適用される。

図12Bでは、本発明の翻訳ダイアログが終了する機能ブロック1206と連携するロジックの拡張が開始している。最初の処理は可能な翻訳経路が計算される機能ブロック1500で始まる。この機能ブロック1500と関係するロジックについては図12Cに詳細に示されている。最初の翻訳の組合せが機能ブロック1520で開始し、その他の全ての組合せは、機能ブロック1530をループの最初にして次の翻訳の組合せと始まることによつて、機能ブロック1510でシステム用に定義された全ての翻訳組合せに関する表が作られる。次に機能ブロック1540で、ユーザーがオープンしようとしている文書からある経路が翻訳できるものか否かが決定される。もしできるのなら、その翻訳組合せが機能 30  
ブロック1550に示すように表中の可能な翻訳に加えられる。その他の組合せが残っているか否かにつき機能ブロック1560でテストされ、もし何も残っていないのなら、処理は機能ブロック1530へ戻される。

図12Bの機能ブロック1600のロジックは図12Dに拡張されている。最初の翻訳経路から開始し、機能ブロック1620に示されるように次の翻訳経路を処理することによつて機能ブロック1610で処理が始まる。次に、選択した文書をオープンするように、そのシステムにアプリケーションがインストールされているか否かを決定するため機能ブロック1630でサーチを行う。もし決定ブロック1640中にアプリケーションがあるなら、機能ブロック1650でそのアプリケーションが有効なエントリーとして記録される。もしそのアプリケーションが決定ブロック1640中になければ、その他の経路が残 40  
っているか否かにつき決定するため決定ブロック1660でテストが行なわれる。もし追加の経路が残っているなら、次の翻訳経路が機能ブロック1620で選択される。もし追加の経路が残っていないなら、制御はターミナル1670から図12Bの機能ブロック1700に戻される。

的確な翻訳ダイアログが同定されたか否かを判断するため図12Bの機能ブロック1700でテストされる。もし何も検出されなければ、不履行という結果が機能ブロック2040にセットされ、制御は図12Aの決定ブロック1208に戻される。逆にもし的確な翻訳ダイアログが確認されたら、機能ブロック1800に進み、図12Eに詳細が示されているように、確定した忠実度に基づいてカテゴリごとに経路が分割される。忠実度はフォーマットごとのデータ保存量についても示される。例えばWORD文書がASCIIフォーマット 50

に変換されるときは、全てのフォーマット情報は喪失してしまう。しかしWORD文書がWordPerfectにトランスレータ経由で変換されるときは、そのフォーマット情報は保存される。このようにASCII翻訳は忠実度が低くWORDからWordPerfectへの翻訳は忠実度が高い。処理は最初の翻訳経路がセットされる機能ブロック1810で開始される。次に機能ブロック1820で次の翻訳経路が立てられ、宛先が忠実度の低いフォーマットであるか否かが決定ブロック1830でテストされる。その宛先が忠実度の低いフォーマットなら、経路は機能ブロック1840の非優先選択チョイスのカテゴリにセットされ、制御は決定ブロック1860に進む。宛先が忠実度の低いフォーマットでないなら、機能ブロック1850で、経路は優先選択チョイスのカテゴリに付加され、処理されるべき経路が別に残っていないかどうかを判断するため制御が決定ブロック1860に進む。もし別の付加される経路が残っているなら、制御はその次の経路を処理するため機能ブロック1820へ進む。もし付加される経路が残っていないなら、制御はターミナル1870へ進み、そこで処理は図12Bの機能ブロック1900に戻される。

10

優先選択機能が機能ブロック1900に設けられる。優先選択機能設置オペレーションに関するロジックの詳細については図12Fを参照のこと。処理は優先選択されたアプリケーションが設置される機能ブロック1910から始まる。次に機能ブロック1920で、優先選択された翻訳経路が設置される。そして最後に機能ブロック1930で優先選択機能が効力発効され、制御が図12Bの機能ブロック2000にターミナル1940から戻される。

図12Gに詳細に描かれているように、内容のリストは機能ブロック2000において作られる。処理は翻訳経路開始がセットされる機能ブロック2100で始まる。次に機能ブロック2110で次の翻訳経路が選択され、アプリケーション名が機能ブロック2120で得られる。するとその経路がその翻訳システムを使うことができるかどうかを判断するため決定ブロック2130でテストが行われる。もし使えるなら、その翻訳システム名が機能ブロック2140で得られ、制御は機能ブロック2150へと進む。もし使えないなら、制御は機能ブロック2150へと進み、翻訳ファイルのリストへエントリを付加する。次に決定ブロック2160で、そのエントリが優先選択機能か否かを判断するためテストが行われる。もしそうなら、その優先選択機能が機能ブロック2170で選択され、制御が決定ブロック2180へと進む。もしそうでないなら、別に処理されるべき経路が残っていないかどうかを判断するため決定ブロック2180へと制御が進む。もし別の付加された経路が残っているなら、制御はその経路を処理するため機能ブロック2110へ進む。もし残っていないなら、制御は2190経由で図12Bの機能ブロック2010へと戻る。

20

30

機能ブロック2010は図6に示したダイアログを示し、機能ブロック2020で示されるユーザーの決定に基づく結果をセットする。すると制御はターミナル2060を経由して図12Aの決定ブロック1208に戻る。ユーザーが優先選択された翻訳プログラムを無効にしたか[override]どうかを判断するため決定ブロック1208でテストが行われる。もしユーザーがその優先選択された翻訳プログラムを無効にしていないなら、ユーザーは機能ブロック1214でその旨を告げられる。もし別のチョイスが適当に選択されたときは、文書が機能ブロック1210で翻訳され、その文書が交換され、そして機能ブロック1212に示されるようにその文書はオープンされる。

40

それともなければ、もしユーザーが機能ブロック1220に示されるようにアイコンをアプリケーションに当てたなら[drag onto]、そのアプリケーションがアイコンの示す文書をオープンすることができるか否かにつき決定ブロック1222で判断するためテストが行われる。もしその文書をオープンできるなら、そのことを示すためにアイコンがハイライトされる。もし文書がそのアプリケーションではオープンできないなら、その文書が図11に示すロジックを採用してオープンすることができるか否かを判断するため決定ブロック1224でテストが行われる。もしその文書をオープンするのに本発明を使うことができるなら、そのアイコンをハイライトするため制御が機能ブロック1230へ進む。もしその文書をオープンするのに本発明を使うことができないなら、機能ブロック1226

50



に示すようにそのアイコンはハイライトされない。

機能ブロック1240に示すように、もしユーザーが別のアプリケーションにとって最適である文書のアイコンを出したときは、その文書をアプリケーションがオープンできるかどうかについて判断するため決定ブロック1242でテストが行われる。アプリケーションが一つも利用できないなら、上述したように本発明を使ってその文書进行处理するため決定ブロック1204に処理を進める。もしアプリケーションがその文書をオープンできるなら、機能ブロック1244に示すように文書はオープンされる。

機能ブロック1260に示すようにユーザーがFile MenuからOpenを選択したときは、機能ブロック1262に示すようにOpenダイアログリストに翻訳することができる文書が追加される。最後にもしユーザーがOpenダイアログにOpenスイッチを入れたら、アプリケーションがその選択された文書をオープンすることができるかどうか判断するため決定ブロック1252でテストが行われる。アプリケーションがオープンできないなら、本発明がその文書をオープンするように決定ブロック1204に制御を進める。

アプリケーションの詳細：

本発明にはプログラム上2つの視点がある。1つは本発明がFinder, Standard File及び将来的なアプリケーションのために提供する道具箱的なサービスである。これはApplication Capabilities Managerと呼ばれる。もう1つの視点は翻訳拡張の観点から要求されるサービスである。図13は本発明に従って作成されたブロックダイアグラムである。Finder 1300は標準的なオペレーティングシステムのApple System 7の一部で、文書をオープンすることを含む通常のファイル処理ルーチンである。アプリケーション1310はワードプロセッサその他の標準的ファイル1320に作用するアプリケーションである。このApplication Capabilities Manager 1330は、Finder 1300が文書翻訳を行うのに必要とするサービスを提供する。翻訳拡張1340は、アプリケーションファイルではないファイルのための翻訳動作のセットである。アプリケーション翻訳拡張1350はアプリケーションのための翻訳拡張を含んでいる。これらの部分については以下に詳述する。

貴方のアプリケーションがオープンできるファイルのタイプを告知する：

Appleのシステムソフトウェアバージョン7.0及びその後のバージョンでは、リソースIDがアプリケーションの束中にリストされている（即ち、タイプ「BNDL」のアプリケーションのリソース）タイプ「FREF」のリソースを検査することによって、どのタイプのファイルのアプリケーションをオープンすることができるかをFinderは決定する。Finderはこの情報を、どのファイルタイプが所定のアプリケーションにドロップラUNCH[drop-launch]され得るのかを判断するのに用いる。アプリケーションの束中にリストされた「FREF」リソースの全ファイルタイプは、それらが関連付けられたアイコンを有しているか否かとは無関係に、そのアプリケーションにドロップ可能と考えられる。

しかし場合によっては、アプリケーションがオープンされるべきでないファイルタイプの「FREF」リソースを含んでいることがある。例えばアプリケーションが辞書のような非文書ファイルを使い、ファイルをヘルプすることがある。これらファイルはアイコンを持つべきなのであり、したがって「FREF」リソースに相当するなのであるが、これらの内容はユーザーに表示され得ない。同様にアプリケーションは優先選択ファイルからデータを読み込むことがあるが、このデータはそのアプリケーションによって内部的に利用されるのであって、ユーザーにより文書としてオープンされない。アプリケーションがオープンできるファイルタイプのリストはアイコンを有しているタイプのリストとは違うものかもしれないので、Translation Managerは新しいリソースのタイプ「open」を定義する。このオープンリソースはどのファイルタイプのアプリケーションが文書としてオープンできるかを宣言する。タイプ「open」のサンプルリソースのRez入力フォームにつき下記に示す。

/\*TeachText用のリソースをオープンせよ\*/

リソース「open」(128)

{'ttxt',{'ttro','PICT','TEXT'}},

オープンリソースはファイルタイプのリストが続くアプリケーションシグナチャでできて

10

20

30

40

50

いる。これはある具体的なアプリケーションが、タイプがそのリスト中にあるファイルをオープンすることができることを示すものである。例えばTeachTextは、それ自身のプライベートなフォーマット、'ttro'、あるいはファイルタイプ'PICT'及び'TEXT'で作られた文書をオープンすることができる。もしMacintosh Easy Openを使用できるのなら、Finderはユーザーがこれらのタイプの文書をそのアプリケーションにドロップすることを可能にする。さらに、もし何らかの翻訳拡張がインストールされているなら、1 特定タイプに翻訳可能なあらゆる文書もそのアプリケーションにドロップすることができる。したがってもしタイプ'SURF'の文書をファイルタイプ'ttro'に翻訳することができる翻訳拡張があるなら、ユーザーはSurfWriter文書をTeachTextにドロップすることができる。もしTranslation Managerが多数のファイルタイプから翻訳のための宛先ファイルタイプとしてのファイルタイプを選択しなければならぬのだとすれば、そのリスト中で最初に出たファイルタイプを選ぶ。どの文書がファイルをオープンするダイアログボックス中にリストされるべきかを決定するために、オープンリソースルーチンStandardOpenDialogによって使われる。

10

アプリケーションがオープンできるファイルのタイプは動的に決定され（おそらくあるフォルダにおいてどのフィルタが利用可能なかを検査することによって）なければならないということは有り得ることである。フィルタは特定のアプリケーションのために利用可能な翻訳ルーチンを指摘する。もしそうなら、システムはオープンリソース中にそれらのファイルタイプを静的にリストすることができない。そのかわりに、オープンできるファイルタイプのリストをランタイムで作り出すために、簡単な翻訳拡張が書き込まなければならない。

20

翻訳拡張の書込み：

翻訳拡張はデータ認識能力及び翻訳能力を供給するために本発明が果たす1 構成要素である。翻訳拡張は1 構成要素であるので、Component Managerによって送られる標準的なセレクトクに応えなければならない。さらに、翻訳拡張は次の仕事を行わなければならない。

その翻訳能力をMacintosh Easy Openへ通信する

具体的な文書及びスクラップのフォーマットを同定する

文書及びスクラップを翻訳する

スクラップは、例えば切り張り作業の対象となる部分のような、文書の1 部をいう。翻訳拡張はファイル、スクラップ、またはこれら両者を同定する。翻訳拡張は'thng'リソース内の構成要素フラグ域にビットをセットすることによってファイル又はスクラップを取り扱う。この原則に対する唯一の例外は、それらがオープンできるのはどのファイルタイプかを動的に決定する必要があるアプリケーションに関する。

30

ファイルタイプとスクラップタイプのデータタイプ：

タイプ

FileType

= OSType; {ファイルタイプ}

ScrapType

= ResType {スクラップタイプ}

ScrapTypeデータタイプはメモリー中にデータのフォーマットを書き込む。普通、本発明により使われるスクラップタイプはScrap Managerによって使われるスクラップタイプと同一である。しかしながら一つ注意しなければならない例外がある。本発明は一定形式のテキストを記述するため、新しいスクラップタイプである'stxt'を定義している。'stxt'フォーマットを持つスクラップは、そのテキストをその形式情報に付加することによって形成される。これは1 スクラップをスクラップ翻訳拡張へと進ませるのに必要である。FileTypeのデータタイプはファイルのフォーマットを記述する。常時あるわけではないがよくあるのは、HFSカタログファイル（以下、ファイルのカタログタイプと呼ぶ）で維持されているように、ファイルのデータのフォーマットがファイルタイプを検査することによって同定し得ることである。しかし翻訳目的のためには、より具体的な同定方法を使うことが必要な場合がしばしばある。例えばデベロッパがアプリケーションを改定するが、そ

40

50

のアプリケーションが作り出す文書のために従来のファイルタイプをそのまま保持するような場合である。これはファイルデータフォーマットの具体的なバージョンを翻訳することはできるが新しいバージョンはできない翻訳拡張にとってトラブルの原因となり得る。同様にMacintoshコンピュータにファイルを作り出す一定のアプリケーション（例えば電子メールプログラムとかディスクマウンティングユーティリティ）は、省略時新ファイルタイプのような標準的ファイルタイプ(例えば'TEXT'とか'BINA')をよく使う。ここでもまた翻訳拡張は、ファイル中のデータの実際のフォーマットを別のフォーマットに翻訳するには、その前にファイル中のデータの実際のフォーマットに関するより多くの情報を必要としている。

ファイルカタログタイプをそのファイルのデータフォーマットに関する唯一の表示として使うことに関係するトラブルを回避するため、本発明は翻訳ファイルタイプを定義している。多くの場合にカタログファイルタイプは翻訳ファイルタイプとして十分である。しかし、本発明は翻訳拡張に具体的なファイルのカタログファイルタイプが本当に翻訳目的のために十分であるかどうかを調べるため、その具体的なファイルを検査する機会を常に与えている。もし拡張が、より特定のフォーマットを同定することができたときは、その情報を本発明に戻すことになる。

翻訳拡張データの構造：

このセクションは翻訳拡張を記述するのに必要なデータ構造について説明する。File Type Specificationsはファイル翻訳リスト中の文書フォーマットについて記載している。ファイルタイプ明細[specification]の域のある一定のものに関する解釈は、その明細がソース文書タイプのリスト中に起こっているか宛先文書タイプのリスト中に起こっているかによって異なる。ファイルタイプ明細はFileTypeSpecデータ構造によって定義されている。

TYPE FileTypeSpec =

フォーマット：

FileType;

ヒント：

LongInt;

フラグ：

TranslationAttributes;

catInfoType;

OStype;

catInfoCreator;

OStype;

END;

フォーマット： 文書の翻訳ファイルのタイプ。本発明では翻訳をするためファイルのフォーマット表現の基準的方法としてこの域を使っている。

ヒント： 貴方の翻訳拡張による使用のために用意されている4バイト値。

フラグ： 翻訳をどのように制御するかにつき特定するビットフラグからなる4バイト値。この域は宛先ファイルのタイプのためにのみ使われる。フラグはあらゆるソースのファイルタイプ明細につき0にセットしておくべきである。現時点では2ビットが定義されているが、その他全てのビットも0にされていなければならない。

定数：

taDstDocNeedsResourceFork = 1;

taDstIsAppTranslation = 2;

本発明は翻訳ルーチンのDoTranslateFileを呼び出す前に、宛先ファイルのデータフォークを作る。翻訳された文書もまたリソースフォークを要求するときは、定数taDstDocNeedsResourceForkによって特定されたビットがセットされる。

関連するアプリケーションがその特定されたファイルのタイプを所望のフォーマットに翻訳することができるので拡張が翻訳を行わないときは、定数taDstIsAppTranslationによ

10

20

30

40

50

って特定されたビットがセットされる。

catInfoType: ボリュームのカタログファイル中に入れられているファイルのタイプ。  
 catInfoCreator: ボリュームのカタログファイル中に入れられているファイルのクリエータ。ファイル翻訳リスト中のソース文書のタイプに関するリスト中に発生するファイルのタイプ明細において、Macintosh Easy Openは、翻訳進捗ダイアログボックス内の'From'フォーマット明細に表示されたカインドストリングを決定するのにフォーマットとcatInfoCreator域を使う。

ファイル翻訳リスト中の宛先文書のタイプに関するリスト中に発生するファイルのタイプ明細において、本発明は、翻訳進捗ダイアログボックス内の'To'フォーマット明細に表示されたカインドストリングを決定するのにフォーマットとcatInfoCreator域を使う。このフォーマットとcatInfoCreator域はまた、Document Converterダイアログボックス中に表示される情報を得るのにも使われる。しかし本発明は翻訳されるファイルのクリエータ及びファイルのタイプを決定するためにcatInfoType域とcatInfoCreator域を使う。

ファイル翻訳リスト:

FileTranslationListデータは、貴方の拡張機能がどのファイルフォーマットを、他のどのファイルフォーマットに翻訳することができるかを記述するように構成される。Translation Managerは、その翻訳能力のマスターデータベースを作るため各翻訳システムから得ることができるファイル翻訳リストを使う。

TYPE FileTranslationList =

RECORD

modDate:

LongInt;

groupCount:

LongInt;

{group1SrcCount:

LongInt;}

{group1SrcEntrySize:

LongInt;}

{group1SrcTypes:

ARRAY[1..group1SrcCount]

OF FileTypeSpec;}

{group1DstCount:

LongInt;}

{group1DstEntrySize:

LongInt;}

{group1DstTypes:

ARRAY[1..group1DstCount]

OF FileTypeSpec;}

{repeat above six lines for a total of groupCount times}

END;

FileTranslationListPtr

=^FileTranslationList;

FileTranslationListHandle

=

^FileTranslationListPtr;

ファイル翻訳リストは、そのリストの改変日を示す域と、上記2つの域に続くグループ数の総数[count]とからなる。拡張によって作られる翻訳リストのサイズは、グループ数、ファイル明細レコードのサイズ、拡張が知っているファイルタイプ数の如何に従い可変である。

modDate: ファイル翻訳リストの作成日。拡張が外部のトランスレータを使っているとき

10

20

30

40

50

は、この域はこれらトランスレータを有するフォルダの改変日にセットされる。

groupCount : 続く翻訳グループの数。

group1SrcCount : 拡張が読むことのできるグループ内にあるファイルのタイプの数。

group1SrcEntrySize : この域に続く列中のファイル明細レコードのサイズ。一般にこの域をSizeOf(FileTypeSize)にセットすることができる。

group1SrcTypes : ファイル明細レコードの列。貴方の拡張が翻訳の仕方を知っている各ファイルタイプのために、この列にファイル明細レコードを含ませよ。

group1DstCount : 拡張が書き込むことのできるグループ中のファイルのタイプの数。

group1DstEntrySize : この域に続く列中のファイル明細レコードのサイズ。一般に貴方はこの域をSizeOf(FileTypeSpec)にセットすることができる。

group1DstTypes : ファイル明細レコード列。この域は拡張が翻訳されることができ  
ファイルタイプの各々のために、この列中のファイル明細レコードを含むべきである。

翻訳リストの作成 :

翻訳拡張は、要求されたときその翻訳能力につきMacintosh Easy Openに情報を与えることができるのでなければならない。これをするため、ファイル翻訳リストがFileTranslationListレコードによって定義されるようにファイル翻訳リストはファイルされているDoGetFileTranslationList機能を定義する。拡張はどのタイプのファイルがどのような他のタイプに翻訳することができるかを示すために戻されたファイル翻訳リストを、本発明は使う。この情報に基づきファイル翻訳リストはその後、具体的な文書を同定し、またできるだけそれを翻訳するために拡張を呼び出すこともある。

FileTranslationListレコードは以下の構成を有している。

```
TYPE FileTranslationList =
```

```
RECORD
```

```
  modDate:
```

```
  LongInt;
```

```
  groupCount:
```

```
  LongInt;
```

```
  {group1SrcCount:
```

```
  LongInt;}
```

```
  {group1SrcEntrySize:
```

```
  LongInt;}
```

```
  {group1SrcTypes:
```

```
  ARRAY[1..group1SrcCount]
```

```
  OF FileTypeSpec;}
```

```
  { group1DstCount:
```

```
  LongInt;}
```

```
  {group1DstEntrySize:
```

```
  LongInt;}
```

```
  {group1DstTypes:
```

```
  ARRAY[1..group1DstCount]
```

```
  OF FileTypeSpec;}
```

```
  {上記 6 行を groupCount の回数だけ反復せよ。 }
```

```
END;
```

このレコードは改変日と続く翻訳グループ数の総数とを有している。ファイル翻訳リスト中の各翻訳グループは、それから拡張が翻訳することができるファイルタイプの集合(group1SrcTypes域)と、拡張がそれに翻訳することができるファイルタイプの集合(group1DstTypes域)を特定する。1翻訳グループ内で拡張は、ソースタイプのものを宛先タイプのものに翻訳できるのでなければならない。

別の翻訳グループたちは、1グループ内にあるさまざまなカテゴリのアプリケーションたちに対応する。例えばワードプロセッシングアプリケーションたちを1グループに、スプ

10

20

30

40

50

レッドシートアプリケーションたちを別のグループに置く、というふうにすることができる。殆ど全てのトランスレータは特定のデータモデルを介して翻訳することで作業するので、殆どの場合group1SrcCountとgroup1DstCountは各々1より大である。こうした殆どの場合、ソースのファイルタイプと宛先のファイルタイプが一部重複したりぴったり一致したりすることが頻発する。

ファイルタイプのいずれかが特定のグループ内で、FileTypeSpecデータタイプで定義されているファイルタイプ明細を使って具体的な文書フォーマットを指定する。

```
TYPE FileTypeSpec=
```

```
RECORD
```

```
format:
```

```
FileType;
```

```
hint:
```

```
LongInt;
```

```
flags:
```

```
TranslationAttributes;
```

```
catInfoType:
```

```
OStype;
```

```
catInfoCreator:
```

```
OStype;
```

```
END;
```

ファイルタイプ明細には、ファイルタイプと、貴方の拡張による使用にとっておかれるヒントと、フラグ域と、もとのファイルタイプとクリエイターとがある。この拡張が翻訳できるファイルタイプのリストは永久に変わらないので、DoGetFileTranslationListはMacintosh Easy Openがそれを呼び出すその最初のときにファイル翻訳リストを満たす。そしてそれが呼び出される1回おきごとにDoGetFileTranslationListは、それが今通過したそのリストを単純に戻す[pass back]。

殆どの場合、翻訳拡張はファイルまたはスクラップを実際に翻訳するには外部のトランスレータに依存している。この場合もそうであるなら、トランスレータをある特定のフォルダ中に又はフォルダから移動させることにより、ユーザーが貴方の拡張によって使われることになるトランスレータを追加したり削除したりすることもできるのが普通である。この場合、DoGetFileTranslationList機能はそのフォルダの改変日を読むことができ、その翻訳リストを再作成するか否かを決定するためmodDate域中に既に入力されている値と比較する。

ファイルの同定：

どのタイプから、またはどのタイプへ拡張が翻訳できるかファイルのタイプを本発明が知ったなら、その拡張が特定のファイルを翻訳することができるか否かを判断するため拡張を呼び出す。ある種の文書は翻訳目的のために十分特定されていないファイルのタイプを持つことがあるので、このような再チェックが必要である。例えば別のオペレーティングシステムから導入された文書は'TEXT'タイプのファイルを持っているかもしれない。さらに別の翻訳拡張は、そのファイルが実際にSurfWriterPCデータを有していて、特殊なフォーマット変換処理をするべきものであるか否かを判断することができるかもしれない。翻訳拡張が特殊な文書を同定するために呼び出されたときは、それは貴方の拡張のDoIdentifyFile機能を呼び出すことになる。

ある種の文書は単にそれらのファイルタイプとクリエイターを調べることによって同定することができる。その他の文書(特にタイプ'TEXT'の文書)は、それら文書が拡張によって翻訳され得るかどうかを判断するためファイルをオープンし、それらの内容を調べることを要求する場合がある。もし拡張がその文書のタイプを認識できなければ、DoIdentifyFileはどのTypeErrにも戻らない。そうでなければDoIdentifyFileはnoErrへ戻り、docKindパラメータがその認識されたファイルタイプにセットされなければならない。DoIdentifyFile機能は'TEXT'をその文書がプレーンな[plain]フォーマット化されていないASCIIテキ

10

20

30

40

50

ストからできていることを確実に知っていない限り、ファイルタイプとして戻してはならない。たとえ拡張がそれが翻訳できるものとして特定の文書を同定したとしても、本発明は実際にはその翻訳をするために拡張を呼び出すことはしないということに注意してもらいたい。

ファイルの翻訳：

もし翻訳拡張が、それが翻訳できるものとして文書を同定し、ユーザーが貴方の翻訳拡張を使うべく選択した場合は、その文書を翻訳するためその拡張が呼び出される。コンポーネントセレクタのディスパッチャはDoTranslateFile機能を呼び出す。

文書の翻訳：

DoTranslateFileルーチンが呼び出される時までには、dstDocパラメータによって特定されたファイルが既に存在している。宛て先ファイルはデータフォークを持っており、またそれは、もし適当な宛て先ファイルタイプ明細内のフラグ域がtaDstDocNeedsResourceFor kビットセットを持っていれば、リソースフォークも持っている。拡張は宛て先ファイルをオープンし、それを翻訳されたデータで満たす。

10

アプリケーションの翻訳拡張：

アプリケーションの翻訳拡張は拡張フォルダ中に置かれるファイルである。殆どのアプリケーションは一定数のファイルタイプしかオープンできず、したがってそれらリソースフォークたちの中に1オープンリソースを含ましめることによってこれらオープン可能なファイルタイプを宣言することができる。しかしアプリケーションによってはどのファイルをオープンできるかにつき動的に決定する必要がある。これはおそらく、これらアプリケーションが外部のフィルタを使うデータ変換能力を既に持っているからであろう。これらのアプリケーションにとってはオープンリソースだけが、どの種類のファイルをこれらアプリケーションがオープンできるのかを特定するのが不適當なものとなっている。

20

アプリケーションがオープン可能なファイルタイプのリストを動的に作成する好ましい方法は、ファイルタイプのリストを作成しファイルを同定することはできるが、実際の翻訳は全く行わないというそんなアプリケーションの翻訳拡張を提供することである。アプリケーションの翻訳拡張は本質的に、アプリケーションがオープンできるファイルのタイプの動的リストを作成するためだけに存在する。拡張が本発明に戻すファイル翻訳リスト中のソースリストは、これらのタイプの各々のためのファイルタイプ明細を含んでいなければならない、タイプの宛て先リストにとって、ファイル翻訳リストはそのフォーマットがある程度随意的でほかには使用されないファイルタイプを有している簡単なファイルタイプ明細を含んでいなければならない。この宛て先ファイルタイプを'VOID'と呼ぶことにする

30

。すると貴方のアプリケーション中のオープンリソースは、ファイル翻訳リスト中の唯一の宛て先ファイルタイプ明細(つまり'VOID')の少なくともフォーマット域中に値を有する静的リストからなる。本発明に係る限りの純粹の効果は、アプリケーションがタイプ'VOID'の文書をオープンでき、別のファイルタイプをタイプ'VOID'に翻訳できる、そんな翻訳拡張が存在することである。その結果、動的に作成されたファイルタイプのリストは、今ではアプリケーションと連携されているのである。





【 5 A 】  
( 従来技術 )

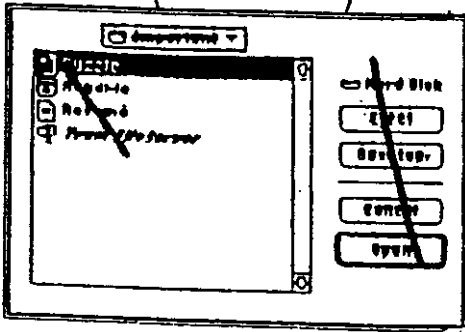


FIG. 5A

【 5 B 】

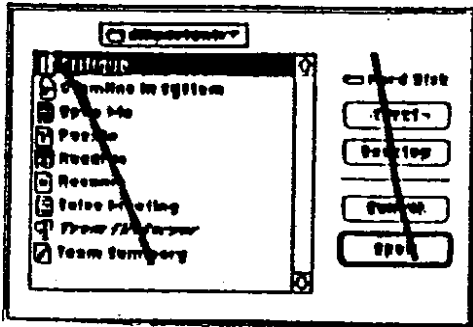


FIG. 5B

【 8 】

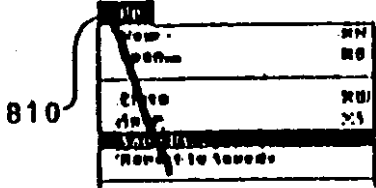


FIG. 8

【 9 】

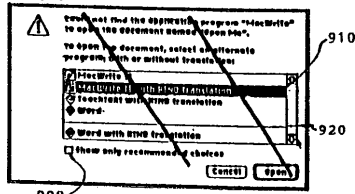


FIG. 9

【 10 】

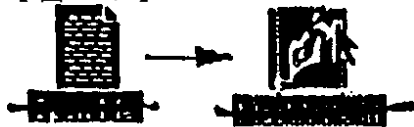


FIG. 10

【 6 】

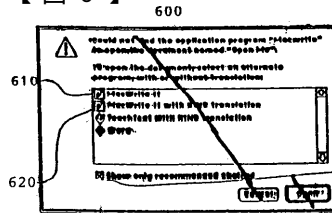


FIG. 6

【 7 】

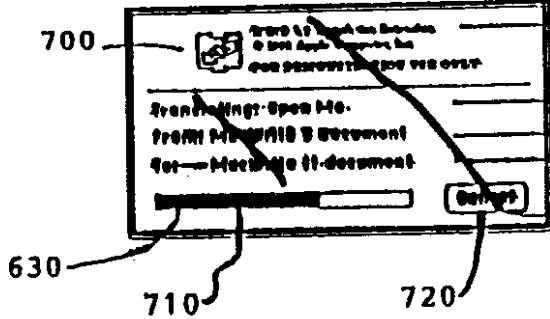


FIG. 7

【 11 A 】

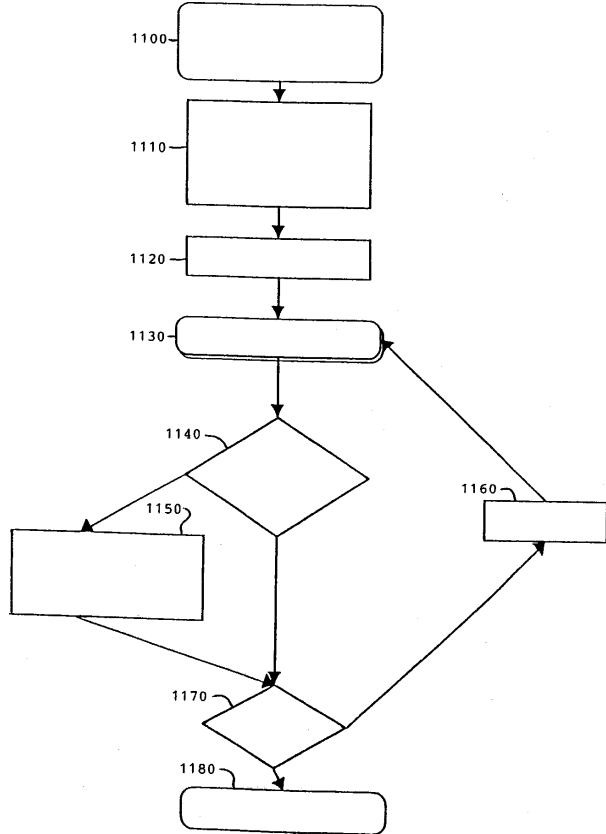
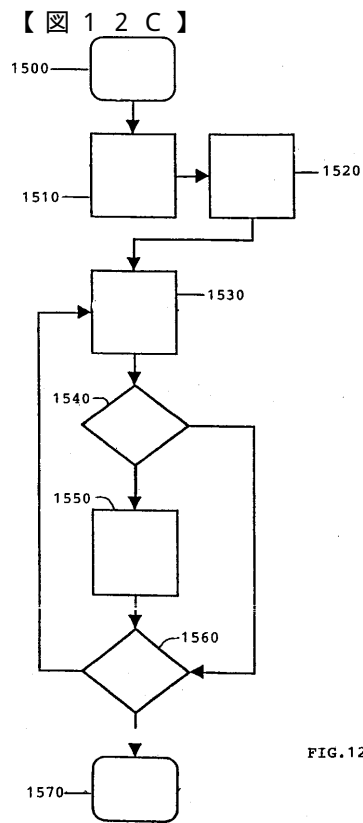
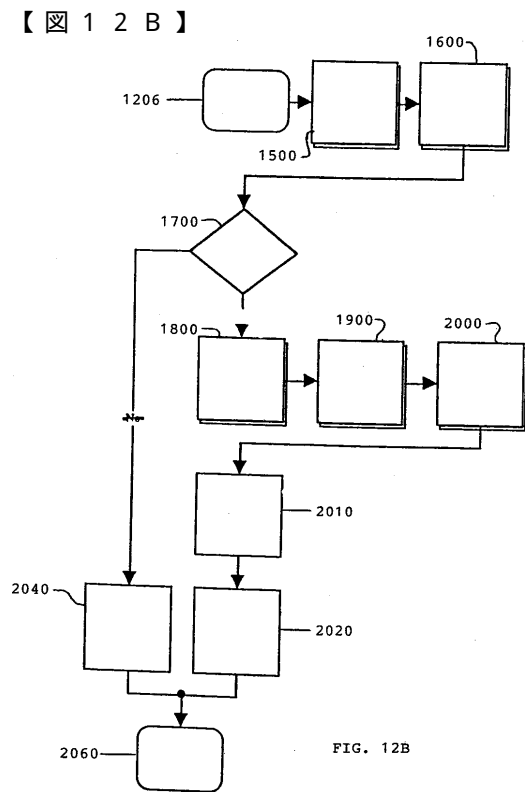
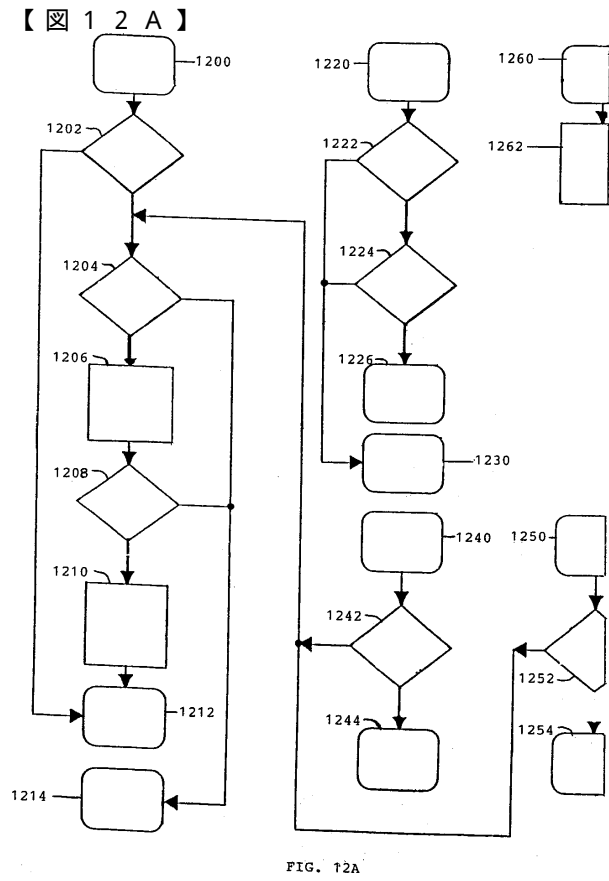
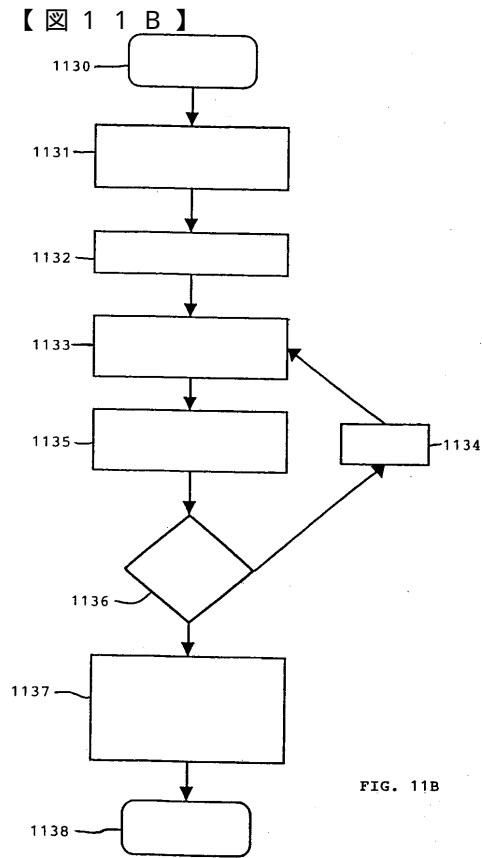


FIG. 11A



【 図 1 2 D 】

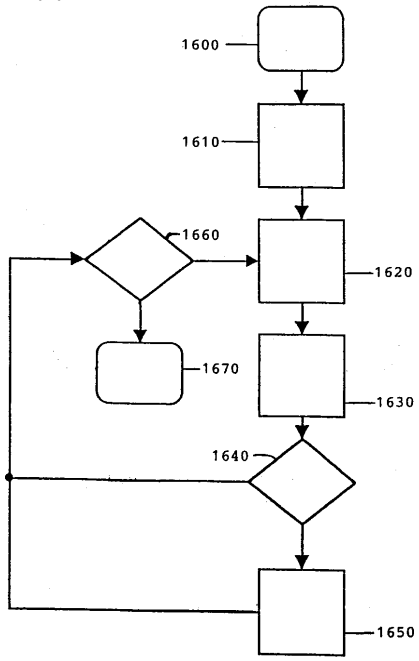


FIG. 12D

【 図 1 2 E 】

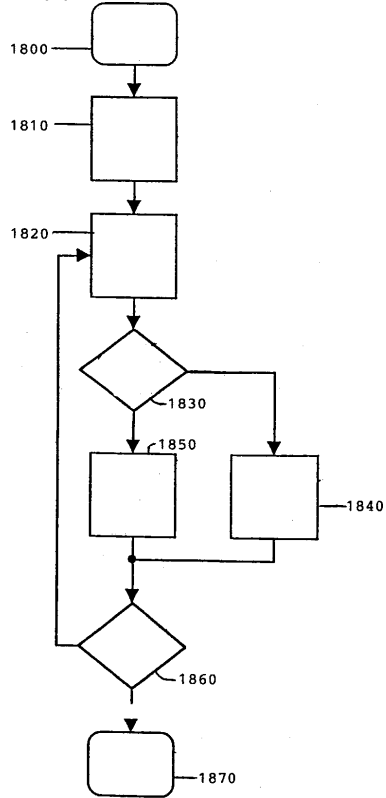


FIG. 12E

【 図 1 2 F 】

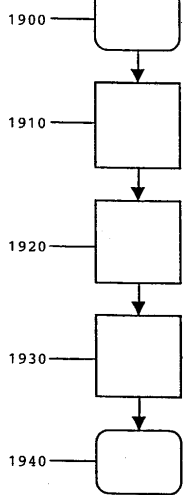


FIG. 12F

【 図 1 2 G 】

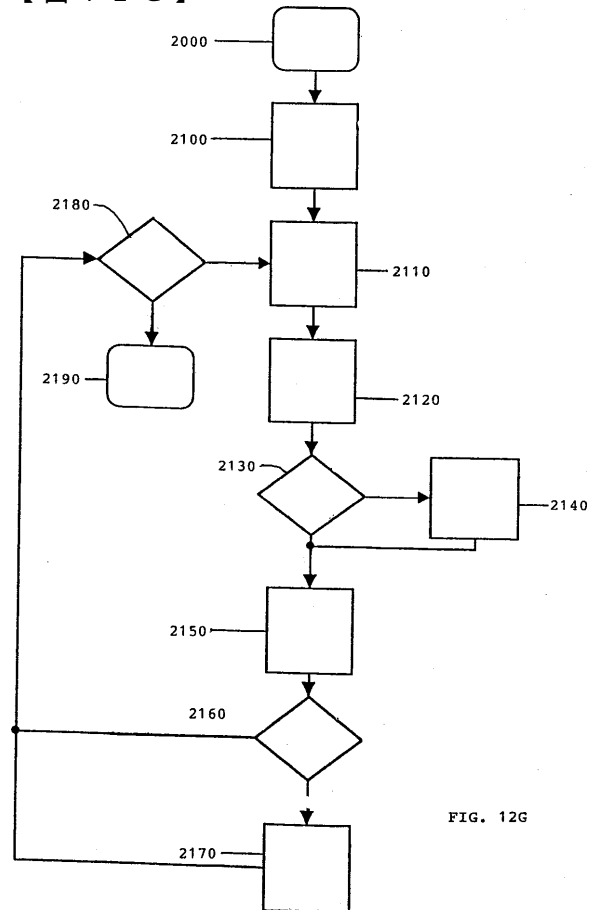


FIG. 12G

【 図 13 】

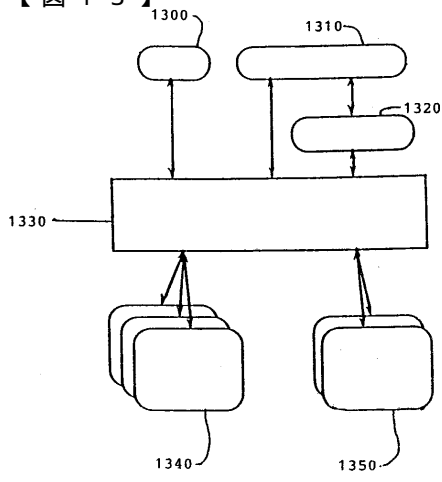


FIG. 13

フロントページの続き

合議体

審判長 吉岡 浩

審判官 相崎 裕恒

審判官 小田 浩

(58)調査した分野(Int.Cl. , D B名)

G06F12/00,G06F15/20