(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3849880号
(P3849880)
(45)発行日　平成18年11月22日(2006.11.22)　(24)登録日　平成18年9月8日(2006.9.8)

(51)Int.Cl.		F I	
G06F 17/21	(2006.01)	G06F 17/21	536
G06F 17/24	(2006.01)	G06F 17/24	554H
G06F 3/14	(2006.01)	G06F 3/14	310B
G06T 11/80	(2006.01)	G06T 11/80	

請求項の数 15　(全 28 頁)

(21)出願番号	特願平6-525566
(86)(22)出願日	平成6年5月4日(1994.5.4)
(65)公表番号	特表平8-510345
(43)公表日	平成8年10月29日(1996.10.29)
(86)国際出願番号	PCT/US1994/005026
(87)国際公開番号	WO1994/027227
(87)国際公開日	平成6年11月24日(1994.11.24)
審査請求日	平成13年1月31日(2001.1.31)
(31)優先権主張番号	08/058,260
(32)優先日	平成5年5月10日(1993.5.10)
(33)優先権主張国	米国(US)

(73)特許権者
アップル　コンピュータ，インコーポレイテッド
アメリカ合衆国　カリフォルニア州　９５０１４　クッパチーノ，ワン　インフィナィト　ループ　（番地なし）
(74)代理人
弁理士　大塚　康徳
(74)代理人
弁理士　松本　研一
(72)発明者　スミス，デイビッド，キャンフィールド
アメリカ合衆国　カリフォルニア州　９５０７０　サラトガ，スカリー　アベニュー　１２１６７

最終頁に続く

(54)【発明の名称】複合文書におけるコンピュータ／ヒューマンインタフェースシステム

(57)【特許請求の範囲】

【請求項１】

文書の内容を前記文書が生成され、編集されたように表示するディスプレイと、編集の間、前記文書の要素の操作を制御するポインティングデバイスと、複合文書の生成と操作のためのコンピュータ・ヒューマンインタフェースとを備えたコンピュータシステムにおいて、

文書の全ての構成オブジェクトを、夫々が固有の内容と前記内容についての付随する操作子とを含むパートとして格納する手段と、

ユーザに前記パートを表示する手段と、

前記デバイスの起動に応じて、前記表示されたパートが、前記ユーザによって選択でき、前記ディスプレイの中で移動でき、他のパート内に置かれ、前記他のパートの内容の一部となり、前記内容が前記付随する操作子によって操作可能であるようにする手段と、

前記コンピュータで前記パートの選択によって、パートに関する操作子を決定し、前記決定された操作子を自動的に立ちあげ、前記パートの機能を前記ユーザに利用可能にする手段とを有し、

前記操作子は、前記パートの内容を表示し、必要であれば前記パートの内容を変更修正するためのユーザインタフェースを提供するソフトウェアであることを特徴とするシステム。

【請求項２】

前記表示手段は、パートをアイコン、或は、フレームで表示し、

前記フレーム内では、前記パートの固有の内容が可視化され、前記付随する操作子によって操作されることを特徴とする請求項１に記載のシステム。
【請求項３】
前記インタフェースは、前記ディスプレイ上にウィンドウを備えるデスクトップメタフォーを含み、
前記ディスプレイでは、文書の内容が表示され、編集と概観目的のためにスクロールされ、
前記デバイスに応答して、パートを、前記アイコンやフレーム表示から、前記パートの内容の表示と操作のためにオープンしてウィンドウにする手段をさらに有することを特徴とする請求項２に記載のシステム。
【請求項４】
前記フレームは、第１のパートと、前記第１のパートの中に含まれる組み込まれた第２のパートとの間の幾何学的な関係を定義し、
前記定義には前記関係を定義するために前記コンピュータに格納される属性が含まれ、
前記属性は、
（ｉ）前記第２のパートの内容のために、利用可能である前記第１のパート内での領域を記述するフレームシェイプと、
（ii）実際に前記第２のパートの内容を含む前記フレームシェイプ内での領域を記述する使用中シェイプとを含む
ことを特徴とする請求項２に記載のシステム。
【請求項５】
フレームはさらに前記格納される属性として,
（iii）前記第２のパートの内容が表示される前記フレームのシェイプ内での領域を記述するクリップシェイプと、
（iv）前記デバイスの起動に関連したイベントが前記第２のパートについての操作子に送られる前記クリップシェイプ内での領域を記述するアクティブシェイプとを含むことを特徴とする請求項４に記載のシステム。
【請求項６】
ユーザがパートに付随する操作子を、前記パートの固有の内容を用いるために適切な異なる操作子で置換できるようにする手段をさらに有することを特徴とする請求項１に記載のシステム。
【請求項７】
他のパートを含むパーツビンとして１つのパートを定義する手段をさらに含み、
前記パートに含まれる他のパートは、前記デバイスによる前記パートに含まれる他のパート内での個々の要素へのアクセスが禁止される一方、前記パートに含まれる他のパートが前記デバイスによって全体として選択され操作されるような凍結特性を有していることを特徴とする請求項１に記載のシステム。
【請求項８】
前記パーツビンは、前記ビンの内容を有するパートがコピーは可能であるが、前記パーツビンから除去されたり、或は、ユーザによって修正されたりすることがないように、ロック可能であることを特徴とする請求項７に記載のシステム。
【請求項９】
前記コンピュータに格納された文書にパーツビンを付随させ、前記文書の内容を前記パーツビンに含まれたパートに限定させる手段をさらに有することを特徴とする請求項７に記載のシステム。
【請求項１０】
文書の第１の要素が前記文書の第２の要素に組み込まれるような複合文書の生成及び操作を行い、前記要素各々は、付随する内容と前記内容の操作子と前記第１と第２の要素の間の関連を定義するインタフェースシステムとを有しているようなコンピュータシステムにおいて、前記インタフェースシステムは、

第１のシェイプを格納する手段と、
第２のシェイプを格納する手段とを有し、
前記第１のシェイプは、前記第２の要素によって決定され、前記第１の要素の内容の配置に対して利用可能である前記第２の要素内に領域を定義し、
前記第２のシェイプは、前記第１の要素によって決定され、前記第１の要素の内容が実際に位置する前記領域の一部を定義し、
前記操作子は、前記内容を表示し、必要であれば前記内容を変更修正するためのユーザインタフェースを提供するソフトウェアであることを特徴とするインタフェースシステム。
【請求項１１】
第３のシェイプを格納する手段と、
第４のシェイプを格納する手段とを有し、
前記第３のシェイプは、前記第２の要素によって決定され、前記第１の要素の内容が表示される前記領域の一部を定義し、
前記第４のシェイプは、前記第１の要素によって決定され、前記システムへのユーザ入力が前記第１の要素に関する前記操作子へ伝えられる前記第３のシェイプの一部を定義していることを特徴とすることを特徴とする請求項１０に記載のインタフェースシステム。
【請求項１２】
各要素が内容を有したオブジェクトとして格納され、前記オブジェクト各々が前記オブジェクトの内容に付随した操作子を有し、第２のオブジェクトの中に第１のオブジェクトが組み込まれているような複合文書をコンピュータシステムにおいて生成する方法であって、
前記第２のオブジェクトに関する前記操作子によって、前記第１のオブジェクトの内容に割り当てられることになる前記第２のオブジェクトの内容の中にある領域を決定する工程と、
前記領域に対応したフレームシェイプを確立し、前記第１のオブジェクトに関する前記操作子に、前記フレームシェイプを通知する工程と、
前記第１のオブジェクトに関する前記操作子によって、前記フレームシェイプの中に前記第１のオブジェクトの内容のレイアウトを決定する工程と、
前記第２のオブジェクトに関する前記操作子によって、前記第１のオブジェクトの内容が表示される前記領域の一部を決定する工程と、
前記第１のオブジェクトに関する前記操作子に、前記領域の一部を通知する工程と、
前記領域の一部に、少なくとも、前記第１のオブジェクトの内容のいくつかを表示する工程とを有し、
前記操作子は、前記オブジェクトの内容を表示し、必要であれば前記オブジェクトの内容を変更修正するためのユーザインタフェースを提供するソフトウェアであることを特徴とする複合文書生成方法。
【請求項１３】
前記第１のオブジェクトに関する前記操作子によって、前記第１のオブジェクトの内容によって占有される前記領域のサブセットに対応した使用中シェイプを決定し、前記使用中シェイプの前記第２のオブジェクトに関する操作子に通知する工程をさらに有することを特徴とする請求項１２に記載の複合文書生成方法。
【請求項１４】
前記使用中シェイプの外側の領域に、少なくとも、前記第２のオブジェクトの内容のいくつかを表示する工程をさらに有することを特徴とする請求項１３に記載の複合文書生成方法。
【請求項１５】
前記第１のオブジェクトに関する前記操作子によって、前記領域の一部のサブセットに対応するアクティブシェイプを決定する工程と、
前記文書の内容の操作に関係するユーザ入力イベントを検出する工程と、
ユーザ入力イベントが前記アクティブシェイプの中で発生したかどうかを判断する工程と

、
前記アクティブシェイプの中での発生が判別されたとき、検出されたユーザ入力イベントを前記第１のオブジェクトに関する前記操作子に提示する工程とをさらに有することを特徴とする請求項１２に記載の複合文書生成方法。
【発明の詳細な説明】
発明の属する分野
本発明はコンピュータとそのユーザとの間のインタフェースに関するものであり、特に、複合文書の生成やその操作をより柔軟に行うことのできるコンピュータ／ヒューマンインタフェースのアークテクチュアに関するものである。
発明の背景
現在、パーソナルコンピュータで使用する多くのアプリケーションは、１つで完全な形をしたプログラム（シングル・モノリシック・プログラム）として記述され、その１つ１つは特定の基本的な機能を実行するように設計されている。例えば、ワードプロセッシングプログラムは、文字の文書を生成・編集するように設計されている。また、文字処理のために設計されている訳ではないが、他の多くのプログラムも、付随的に文字を使用できるようになっている。例えば、帳票プログラムは、その帳票に情報を設定する行と列に、記述子として文字が挿入できるようになっている。同様に、ペインティング、ドローイングプログラムのような図形アプリケーションも、そのペインティング、ドローイングの中にラベルとして文字の使用ができるようになっている。このように、これら種類の異なるプログラムの全てに、文字編集の機能が備えられているが、一方、その文字を編集する方法は互いに異なっている。あるアプリケーションでは、この機能について多くの能力を備えているが、他のアプリケーションではその機能は限定的なものでしかない。その結果、ユーザは同じタスクを実行する、即ち、文字を編集するのにも、異なる種類のアプリケーション各々について、異なる操作方法を学ぶ必要がある。
それ故に、文字編集を行う環境がどうであれ、それが必要とされるときにはどこでも、同じ統一された仕方で、文字のような文書のある１つの要素の生成・編集ができるようなアークテクチュアを備えることが望ましい。そのような考え方のもとに、ワードプロセッシング文書であれ、帳票であれ、図形文書であれ、その他どんな文書であれ、同じ体系のコマンドが文字を編集するために用いられる。ユーザはただ１つのテキストエディタについて学びさえすれば良く、従って、その編集作業はより作業がし易いものになる。
こうした考え方をさらに進めると、文書それ自体の中で、複合文書の要素を編集できることはもっと望ましいことである。本発明においては、２つ以上の異なったプログラムが用いられて、１つの統合された情報の形をとるようになる複合文書について考慮する。例えば、それは、図形プログラムによって生成された図面を含む文字文書であるかもしれない。これら２つのタイプの情報は、互いに相手を包含するような関係をもっている。特に、文字文書は、図形オブジェクトのコンテナとして機能し、図形オブジェクトは文字文書（コンテナ）とともに移動したり、影響を受けたりする。
過去において、この種の複合文書は、２つの異なるアプリケーションを実行させることで、生成される。ワードプロセッシングプログラムは、その文書の文字部を生成するために用いられてきた。別の図形アプリケーションは、図形を生成するために用いられてきた。これら２つのプログラムの生成物を１つの文書に組み込むためには、図形は特殊なデータ様式或は形式で保存され、そのデータ構造の内容が、例えば、カット・ペースト操作によって、ワードプロセッシング文書の中に置かれることになる。しばしば、これら別々の情報を１つの文書にマージすることは、プレゼンテーションマネージャーとして言及される３番目のプログラムによって達成されてきた。
１度、図面がワードプロセッシング文書に組み込まれると、その内容は静的なものになる。もし、その図面を修正する必要があるなら、ユーザは図形プログラムを呼び出し、元々の図面を引出し、その図面をその図形プログラムの実行下で編集する必要がある。その編集が、複合文書の外側でなされると、ユーザはその変更を、その変更がなされたようには見ることができない。その修正された図面は、それから、元々の図面の代わりに、ワード

プロセッシング文書に組み込まれなければならない。
種々のプログラムの間での、そのような切り替えの必要を克服するために、複合文書の種々の要素が、複合文書の中で編集できるようにすることが望ましい。このようなやり方を用いれば、時間を節約するのみならず、手順も簡略化でき、ユーザが修正された内容をそのまま見ることができるようになる。
パーソナルコンピュータに関する従来のオペレーティングシステムの別の特徴には、種々のタイプの情報を操作する方法に種々の制約があった。例えば、文書と他の種類のファイルがコンテナに格納され、それらは、アップルコンピュータ株式会社によって提供されているマッキントッシュ・オペレーティングシステムでは“フォルダ”として認識され、マイクロソフト株式会社によって提供されているＭＳ－ＤＯＳでは“ディレクトリ”或は“サブディレクトリ”として認識される。フォルダやディレクトリは、文書を含んではいるが、それら自身は文書には組み入れることはできない。
同様に、アプリケーションや文書は、マッキントッシュ・ファインダシステムやマイクロソフトのウインドウズ環境のようなデスクトップメタフォ図形インタフェースでアイコンとして表現される。現在、これらのインタフェースでは、アイコンでの操作方法には制約がある。例えば、アイコンは、フォルダ（サブディレクトリ）に属するデスクトップ或はウインドウにのみ存在する。もし、そのウインドウがフォルダ或はボリューム以外の何か、例えば、文字文書を表現していれば、アイコンをそこに置く事はできない。さらに、ある種のアイコンは、他の特定のアイコン上にしか置く事ができない。例えば、あるアイコンが文書を表現しているなら、それはただ、その文書に属しているアプリケーションを表現するアイコン上にしか置く事ができない。その文書のアイコンは、他の種類のアプリケーションに属しているかもしれない他の種類のアイコン上に置く事はできないのである。
従って、本発明の目的は、あるタスクの実行される環境に係わりなく、そのタスクに関しては一貫性のあるユーザインタフェースが備えられたシステムを提供することである。さらに、種々のタイプのデータ操作についての制約をなくし、これによって、全ての種類の複合文書の生成や編集をより柔軟に行うことができるようにすることも本発明の目的である。

本発明の簡単な説明

これらの目的の追求のため、本発明は、あらゆる異なった種類の情報を伝える要素が、以前は不可能であった方法で文書内で結合可能、また、操作可能になる均一なインタフェースを備える。本発明では、文書はインタフェースの１要素と考えられる。また、そのインタフェースの目的は、ユーザに情報を伝えることであり、その情報、例えば、文字文書や図形文書を編集できるようにすることである。
そのインタフェースの均一性は、以下、タスクを達成する“パート”として言及される基本的構造ブロックを用いる文書を中心としたアーキテクチュアによって備えられる。パートは、２つの基本的な構成要素、即ち、内容と、その内容の操作子で構成される自己完結的な要素である。これら２つの構成要素は、これらがシステムのどこに、例えば、文書の要素に、フォルダに、或は、デスクトップに、存在していようと、常にユーザには利用可能となっている。そのパートの操作子は、そのパートの内容に関して、いかなる種類のエディタやビューワともなり得、もし必要であれば、ユーザによって選択可能となっている。例えば、あるパートの内容が文字であれば、そのパートの構成要素であるエディタは、ユーザが親しんでいるどんなワードプロセッサともなり得る。そのワードプロセッサは、そのパートの文字部分を、その文字部分がどこに組み込まれていようと、編集するのに用いられる。従って、たとえそのパートが文字ではない文書、例えば、帳票などに組み込まれていようと、そのパートの文字部分は、依然として、ユーザが選択したワードプロセッサで編集できる。
この特徴を表わすインタフェースは、重要な固有の性質、即ち、より単純で、同時により強力であるという性質をもっている。この単純性は、ユーザはただ、文字を編集するといった特定のタスクを実行するのに１つの方法だけを学べは良いという事実から来ている。同時に、その文字エディタは十分な機能を備えたモジュールとなり得、それは、現在多く

のアプリケーションで見かける能力のないものを置き換えることができるという点で、より強力なものと言える。
パートの２つの基本的な構成要素は、オブジェクト（その固有の内容のために）としてもアプリケーション（その固有の操作子やエディタを備えた結果）としても機能することができる。これらの特徴のために、パートは数多くの機能を備えている。例えば、文書１つ１つは、パートであり、パート１つ１つは１つの文書として振る舞う。このことを説明するために、文字パートは文字を含む文書であり、図形パートは線や円を含む文書である。本発明では、フォルダやデスクトップでさえも、パートを構成し、従って、これらは、以前は文字のような文書のみと結び付けられてきた内容を含むことができる。パートのこの特徴は、フォルダのようなオブジェクトが文書としては機能しない現在利用可能なアプリケーションには存在していない均一性を備えるものとなっている。
パートの別の特徴は、それがコンテナとして機能する、即ち、１つのパートが他のパートを含むことができるという点である。さらに、１つのパートは、それが含む他のパートの種類を制限しない。むしろ、もし、あるパートが他の種類のパートを含むとき、それは、全種類のパートを含むことができる。このパートの特徴は、本発明を、複合文書の生成に関して理想的なアーキテクチュアとしている。１つの文書が複数のパートを含むので、それは自動的に異なる種類の情報を表現する全ての種類のパートを、文書の変更や操作子がなくても、含むことができるようになる。
これらの特徴は、望むような結果を産み出すように異なるパートをアレンジすることで統合される複合文書におけるインタフェースを形造る結果となる。この意味において、本発明のインタフェースは、アプリケーション指向というよりはむしろ、文書中心的である。即ち、従来のアプリケーションプログラムは、パートを形成するように結合されるデータとは分離された本発明のインタフェースにおける役割を備えていなかった。
本発明の別の特徴は、ユーザに対して表現されるパートの方法である。１つのパートは、小さな絵でパートの操作方法を備える１つのアイコンとして表現され、それが全体として操作できるようになっている。例えば、アイコンは、デスクトップ上やウィンドウ間でドラッグされ、ドラッグ操作の間、他のアイコンをドロップする目的ともなり得る。また、パートの内容は、従来のアプリケーションと同様にして、デスクトップ上のウィンドウによって定義された作業領域に拡張される。
パートは、また、アイコンの代わりにフレームとしても表現される。フレームは、パートを表現するディスプレイスクリーンの領域である。アイコンのように、フレームは全体として操作されることが可能なパートへのハンドルを備えている。それらは、デスクトップ上で或はウィンドウ間でドラッグが可能であり、ドラッグ操作でのドロップ先を備えており、ウィンドウにオープンしたり、これをクローズしてフレームに戻すこともできる。しかしながら、アイコンとは異なって、フレームは、パートの内容をみたり適切に修正したりすることができる。
フレームは、それが含むパート、例えば、文書と組み込まれたパートとの間でのインタフェースとして機能するデータ構造である。フレームは、多くの属性から構成され、そのいくつかは、それが含むパートによって制御され、、またいくつかは組み込まれたパートによって制御される。フレームを媒介として、組み込むパートや組み込まれるパートについての操作子は、組み込むパート内でその組み込まれたパートに与えられた空間割り当てを調整することができる。
ユーザインタフェースに関する基本的な構造ブロックとしてのパートを利用することにより、本発明は以前のパーソナルコンピュータアプリケーションでは利用できなかった能力を備えることができる。これらの能力は、これ以降、本発明の好適な実施例を参照して説明する。

【図面の簡単な説明】

図１はコンピュータシステムの基本的な構成要素を示すブロック図である。
図２Ａ及び図２Ｂは、スクリーンディスプレイの２つの画面であり、パートを１つのフォルダより文書にドラッグしている様子を示している。

図３Ａ及び図３Ｂは、スクリーンディスプレイの２つの画面であり、フレームのアクティベーションの様子を示している。
図４Ａ及び図４Ｂは、アクティブフレーム内の選択されたフレームの様子を示す図である。
図５Ａ及び図５Ｂは、フレーム構造の２つの例を示す図である。
図５Ｃ－図５Ｆは、フレーム属性に関し、組み込まれたパートと組み込むパートとの間で発生する調整の例を示すフロー図である。
図６は、エディタを立ちあげる過程を示すフローチャートである。
図７は、デスクトップと文書との間でのパートのドラッグの手順を示すスクリーンディスプレイである。
図８は、パートをドラッグしたり、ドロップする過程を示すフローチャートである。
図９Ａ－図９Ｃ、図１０Ａ－図１０Ｃ、図１１Ａ－図１１Ｃは夫々、マテリアルをあるパートから別のパートにコピーする手順を示すスクリーンディスプレイである。
図１２は、マテリアルをあるパートから別のパートに置くようにする過程を示すフローチャートである。
図１３は、制限されたアクセスをもったパートの様子を示すスクリーンディスプレイである。
図１４は、パートビンを示したスクリーンディスプレイである。

詳細な説明

本発明とその発明によって達成される利点の理解を容易にするために、本発明の特徴は、アップルコンピュータ株式会社によって供給されているマッキントッシュ（登録商標）ブランドのコンピュータでの本発明の使用を参照して説明する。しかしながら、関連する技術に通じている人々にとって、本発明の原理は、この特殊なオペレーティングシステム環境にのみに限定的に適用されるものではないことが理解されよう。むしろ、本発明を基礎とする原理は、文書、特に、複合文書のコンパイルや編集のために、均一なインタフェースを備えることが望ましいあらゆる種類のコンピュータのオペレーティングシステムにでも適用可能である。
本発明が採用されるような種類の典型的なコンピュータシステムは、図１のブロック図に示されるものである。そのコンピュータの構造そのものは、本発明のパートを形成しているのではない。ここで、本発明の特徴とそのコンピュータの構造とが協働した方法に関し、後での理解のために、簡単に、コンピュータシステムについて説明する。図１において、そのシステムは、種々の外部周辺装置１２が接続されるコンピュータ１０を含んでいる。コンピュータ１０は、中央処理装置１４と、通常、ランダムアクセスメモリ１２、リードオンリメモリ１８を含むスタティックメモリがインプリメントされたメインメモリ、そして、磁気、或は、光学的ディスク２０のような永久記憶装置を含んでいる。ＣＰＵ１４は、内部バス２２を経てこれら種々のメモリ１つ１つと交信する。周辺装置１２は、キーボードようなデータ入力装置、マウス、トラックボールなどのようなポインティング、或は、カーソル制御装置２６を含んでいる。ＣＲＴモニタ、ＬＣＤスクリーンのようなディスプレイ装置２８は、コンピュータ内で処理される情報、例えば、文書の内容の可視化表示を行う。この情報のハードコピーは、プリンタ３０或は類似の装置によって備えられる。これら外部周辺装置各々は、ＣＰＵ１４と、そのコンピュータの１つ以上の入出力ポート３２によって、交信する。
本発明は、特に、ＣＰＵ１４がキーボード２４やカーソル制御装置２６によって操作される情報を扱う方法と、操作された情報を表示装置３０をとおしてユーザに描写する方法とを制御するシステムに関するものである。本発明の原理を採用したインタフェースの例が、図２Ａと図２Ｂに示されている。これらの図は、作業領域３６を定義するデスクトップ３４を描写している。また、デスクトップにはメニューバー３８が含まれている。作業領域３６の中には、２つのウィンドウ４０と４２がある。左側のウィンドウ４０は、フォルダウィンドウであり、種々のパートを表現するアイコンを含んでいる。図２Ａに示されているように、３つのアイコン４４は文字パートに属し、４番目のアイコン４５は図形パー

トを表現し、他のアイコン４６は帳票パートを表現している。６番目のアイコン４８は、プレゼンテーションスライドの生成のために用いられるパートを表現するステーショナリアイコンである。図２Ａにおいて、右側のウィンドウ４２は文字文書を含んでいる。即ち、その文書は文字パート５０を含んでいる。この文字パートは、ウィンドウ４２内で文字をタイプすることにより、或は、そのウィンドウ内で以前に生成されアイコンで表現されている文字パートをオープンすることにより生成される。

図２Ｂは、操作の結果を示し、図形アイコン４５が左側ウィンドウ４０のフォルダから右側ウィンドウ４２の文字文書にドラッグされている。図２Ａと図２Ｂに示されたようなデスクトップメタフォーを備えたインタフェースでは知られたことであるが、ドラッギングは、カーソル制御装置２６を動作させてデスクトップ上やウィンドウ内でオブジェクトを移動したりコピーしたりする動作のことである。いったん、図形アイコンが文書の中に置かれると、その内容がフレーム５２の中に表示される。文書は今や文字要素の他に図形要素を含んでいるのであるから、それは、複合文書として言及されるものである。文字５０とフレーム５２の中にある図形要素は別々のパートに属しており、これらが集合して文書を作り上げる。前述したように、パートは、内容とその内容の操作子とから構成される自己完結型のエンティティである。これら２つの構成要素は、そのパートがアクセスされたときは、それが、デスクトップのどこに位置していようと、常に利用可能である。従って、図２Ａに示すように、図形アイコン４５がウィンドウ４０のフォルダに位置しているとき、それは、図形要素とその要素についてのエディタとを含むパートを表現している。そのパートがウィンドウ４２の文書に移動したとき、その内容もパートとともに移動し、それらの内容のエディタの機能もユーザには利用可能である。

この特徴で重要なことは、パートの内容は、それがどこに位置していようとも常に編集可能、さもなければ、操作可能であるという事実である。従って、図２Ｂの複合文書において、文字の内容５０は、その文書内で、例えば、その文字を始めて生成したときに用いられたと同じワードプロセッサを用いて編集可能である。図形エディタが、フレーム５２によって表現されるパートの構成要素として存在しているので、パートの図形要素も適宜に直接編集可能となる。従って、パーソナルコンピュータの以前のアプリケーションとは異なり、ユーザが図形アプリケーションを、例えば、デスクトップ３４の別のウィンドウでオープンし、その図形情報を所望の形に編集し、その後、編集された図形をウィンドウ４２の文書に移動させる必要はない。むしろ、ユーザは、直接、ウィンドウ４２の文書の内容の中で、中間的な過程を組み込むことなく、フレーム５２の内容を編集することができる。ユーザは、アプリケーションプログラムではなく、むしろ、文書の内容に注意を集中することができ、周辺の文書によって備えられた文脈を利用することができる。

それ故に、パートとは、基幹システム技術で自律性のある自己完結的なオブジェクトである。それは、どんな種類のソフトウェアエンジンでも環境でも使用することができ、ユーザはパートについて特別なエディタの操作を学ぶ必要はない。そのカテゴリーのパートのエディタが基幹システムで表現されている限り、ユーザは、そのパートの能力を十分に利用することができる。さらに、もし、文字部分が特別なテキストエディタ（ワードプロセッサ）で生成されたなら、ユーザはただそのテキストエディタをそのパートに用いるように限定されている訳ではない。もし、そのユーザが異なるワードプロセッサのコマンドにより親しんでいるなら、そのプロセッサを、コンピュータシステムでの全ての文字パートに関する好適なエディタとして指定できる。

パートは、文書やその内容にのみ限定されない。むしろ、システム内のどのオブジェクトもパートを構成する。従って、ウィンドウ４０のフォルダは、パートであり、その内容はアイコン４４－４８である。同様に、デスクトップ３４それ自身もパートであり、ダイヤログボックスや他のユーザインタフェースオブジェクトも同様である。

フレームとアイコンは、パートの２者択一的な表示である。ユーザは、パートのフレーム表示とアイコン表示とを切り替えることができ、これは、適当なキーボードからのコマンドやメニューコマンドによってなされる。もし、文書が大きく、例えば、数ページにも及ぶなら、その一部分だけしかフレームに表示されない。それゆえ、アイコンのようなフレ

ームは、ウィンドウ内にオープンすることができ、これによって、例えば、これをスクロールして大きなパートの全ての内容を見る事ができ、ウィンドウ内で利用可能な機能、例えば、拡大、を用いて編集することができる。同様に、そのパートはウィンドウ表示からクローズしてフレーム表示に戻すこともできる。フレームは、その内容を適宜編集することができるという点で、アイコンと異なる。フレームはまた、１つのフレーム、或は、複数のフレームがウィンドウ内に存在するので、ウィンドウとも異なる。ウィンドウは、オブジェクトの遷移的なビューであり、それは、ただパートが編集されたり、検証されたりするときにのみ、オープンされたままになる。これに対して、フレームは、パートの内容の永久的な表示方法である。

パート、それはアイコン或はフレームの表示形式をとっているが、それは、他のパートのコンテナとしても働く。図２Ｂの例において、文字５０はフレーム内に存在するが、そのフレームの境界は、ウィンドウ４２のそれと同じである。この比較的大きなフレームは、図形パートに関する比較的小さなフレーム５２を含んでいる。その文字部分は、ウィンドウの最外部にあるので、それば、“ルートパート”とラベルが付される。一般的にいって、そのルートパートは、ウィンドウに関する基本的な編集方針を確立する。

ルートパートは、図２Ａで示されるアイコン４８のような、ステイショナリアイコンによって生成される。一般に、ステイショナリアイコンは、予め定義されたパートを表現し、物理世界におけるペーパ台についてのメタフォーとして機能する。新しい文書を生成したいとき、人はペーパ台より未使用の用紙を一枚むしりとり、文書の内容、例えば、単語や図面などをその上に置く。ステイショナリアイコンも同様の役割を果たすのである。ユーザが新しい文字文書を生成したい場合、文字文書ステイショナリアイコンが、それをダブルクリックすることによって、オープンされる。これに応答して、そのステイショナリアイコンは、それ自身のコピーを生成し（或は、“むしりとり”）、ウィンドウやフレーム内にそれをオープンする。このコピーは、最初は何の内容も入っていないブランクか、或は、レターヘッドや会社のロゴのような所定の情報をもった普通のパートである。それから、ユーザは、その文書に付加的な内容、例えば、文字をいれる。

ステイショナリそれ自身は、ウィンドウしてもフレームとしてもオープンはされず、ただアイコンとして存在する。はぎとられたコピーのみが、オープンするのである。従って、ステイショナリアイコンが文書内にドラッグされた場合、それ自身をフレーム内にオープンするというよりむしろ、コピーがむしりとられ、文書内に置かれ、そのコピーがフレーム内でオープンするのである。ステイショナリアイコンそれ自身は、移動前に占めていた位置に戻る。これによって、ステイショナリが間違って文書に挿入されたり、或は、失われていまうことを防いでいる。

フレームは、コンテナとして機能するのであるから、それはウィンドウの良く知られた性質を示すことになる。例えば、ユーザがフレーム内に含まれる図形オブジェクトのようなパートを選択すると、そのパートを含むフレームは、アクティブになる。パートがアクティブであると、それはコマンドやキーボードのイベントを受け付け、そのメニューと他のユーザインタフェースオプションが表示される。フレームのアクティベーションの例が図３Ａと図３Ｂに示されている。図３Ａは、文字と図形部分とを有する複合文書を含むウィンドウ５４を有したデスクトップを示す。この複合文書において、文字はその文書のルートパートを形成する。図３Ａの例では、その文字の一部５６が選択され、反転表示されている。この場合、選択されたパートを含むフレームは、ウィンドウ自身の境界に接している。デスクトップのメニューバー５８は、文字を編集するために適切なコマンドを含んでいる。

図３Ｂは、図形要素、即ち、三角形６０が選択された例を示している。その図形要素を含むフレームが、今度はアクティブになり、その境界線は、点線６２によって示されている。そのアクティブフレームが今回は図形パートであるので、メニューバー５８のコマンドは、図形の内容を編集するのに適切なものに変えられる。さらに、パレット６４がデスクトップに表示され、アクティブフレームの図形の内容を編集するのに必要なツールを備える。

フレームをアクティベートすることに加え、ユーザは、また、フレームを選択しても良い。そのフレーム選択は、例えば、フレームをアクティベートしてその境界が見えるようにし、その後、カーソルのポインタをフレームの境界上に位置させる一方、カーソル制御装置のボタンを押してすることで達成される。図４は、図形の内容と文字の内容の両方を含むフレーム６６を図示している。そのフレームの境界６８は、点線によって示され、それがアクティブフレームであるという事実を示している。そのアクティブフレームの中に、文字の内容を含むフレーム７０がある。このフレームは、カーソル７２のポインタをフレームの境界７３の上に位置させる一方、カーソル制御装置のボタンを押すことによって選択される。フレーム７２が選択されたという事実を示すために、選択フレーム７２の境界７３は、アクティブフレーム６６のそれとは異なる外観となる。例えば、その境界は、図４にしめすように、広くなる。さらに加えて、サイズ変更ハンドル７４が備えられ、これを用いてフレームの形やサイズをよく知られた方法で変化させることができる。

一般に、パートの内容の一部或はいくつかが選択されたときはいつでも、選択された要素を含む最小のフレームがアクティブフレームとなる。従って、フレーム７０内の１つのワードが、例えば、カーソルをそのワードに位置付けしてカーソル制御ボタンをクリックすることによって、選択されると、フレーム７０はその状態が選択よりアクティブへと変化する。同時に、フレーム６６の境界は目には見えなくなる。なぜなら、このフレームは、もはやアクティブでもなければ、選択されている訳でもないからである。逆に、もしフレーム６６が、その境界６８上をクリックすることで選択されたなら、境界７３が消滅し、境界６８が現れるようになり、フレーム６６を含むフレーム（或はウィンドウ）の境界（図４には示されていない）が高輝度表示になって、それがアクティブであることを示す。

図３Ａ、図３Ｂ、図４の例において、フレームは矩形であるように示されている。さて、この形はたいていの種類のパートの内容を表示するためには好ましいものであるが、フレームの形は、矩形に限定される必要はない。むしろ、そのフレームは、その表示内容に適するようにどんな形をとっても良い。

フレームは、パートそれ自身の要素ではない。むしろ、フレームは、別のデータ構造であり、それは、コンテナパートと組み込まれるパートとの間のインタフェースとして機能する。フレーム各々は、多くの特徴的な属性を有している。その属性は、種々の“シェイプ”として示される。これらのシェイプ各々は、コンテナパートと組み込まれるパートとによって共有されるある文書の内容を定義する領域の幾何学的な記述を表現している。そのシェイプのいくつかは、コンテナパートによって制御され、一方、他のものは組み込まれるパートによって決定される。一般に、これらの種々のシェイプは、文書の文章の座標空間の任意の領域を表現できる。この点、フレームのシェイプは、隣接領域と同様に分離領域を表現できる。

フレームの種々の特徴的なシェイプを、図５Ａと図５Ｂとを参照して説明する。これらの図において、左側の図は、文字（ルートパート）と三角形によって示される図形パートとを含む複合文書を表わしている。一方、右側の図は、図形パートに関するフレームのシェイプ属性を示している。

フレームの特徴的な属性の１つは、フレームシェイプである。そのシェイプは、コンテナパートによって決定され、その組み込まれたパートにそのレイアウトを示すために用いる幾何学的な領域を記述する。図５Ａと図５Ｂとにおいて、フレームシェイプは、最外部の実線の境界７５によって示されている。そのシェイプによって示される領域は、スケーリングやライン切断や、組み込まれたパートに関するエディタによってなされねばならないその他類似のレイアウト決定に影響する。ただ、コンテナパートだけが直接にフレームシェイプを変更できる。このシェイプがコンテナパートによって変更されたとき、通知が組み込まれたパートに対してなされ、適切な変更が組み込まれたパートの内容においてもなされるようにしている。

フレームの２番目の特徴的な属性は、そのクリップシェイプである。これは、図５Ａと図５Ｂの斜線が施された境界７６によって示されている。そのクリップシェイプは、その組み込まれたパートに、そのパートの内容が記述可能な幾何学的な領域を記述する。図５Ａ

の例において、クリップシェイプは、フレームシェイプと共存する。しかしながら、図５Ｂの例では、三角形の図形パートの左下側の部分は、表がオーバレイしている。従って、図形パートに関するクリップシェイプ７６は、その表によって覆われてしまう領域を除外している。クリップシェイプは、コンテナ、即ち、図５Ａと図５Ｂで示されている例のルートパートを形成する文字文書によって決定される。ただ、そのコンテナだけがクリップシェイプを変更でき、それがなされたときには、コンテナはその変更を組み込まれたパートに通知する。

フレームのさらに別の特徴的な属性は、図５Ａと図５Ｂに実線７７で示されている使用中シェイプである。このシェイプは、組み込まれたパートによって決定され、コンテナに対して、その組み込まれたパートによって実際に使用されている幾何学領域のパートを記述する。このシェイプは、必要に応じてフレームシェイプのサブセットになる。コンテナパートは、使用中シェイプに引張ってくる事が許されているのではなく、使用中シェイプの外側の領域、それがたとえ、クリップシェイプやフレームシェイプの内側であっても、その領域に引張ってくることは自由である。その組み込まれたパートが使用中シェイプを変更でき、それがなされたときには、その組み込まれたパートはその変更をコンテナパートに通知する。

フレームの特徴的な属性である４番目のシェイプは、そのアクティブシェイプであり、これは、図５Ａと図５Ｂの影が施された線７８によって示されている。このシェイプは、組み込まれたパートによって決定され、そのコンテナパートに対して、その組み込まれたパートがカーソル制御入力のような幾何学的に関連したイベントを受け付ける幾何学的な領域を記述する。イベントがその領域で発生したときにはいつでも、コンテナパートはそのイベント自身に応答するのではなく、そのイベントをその組み込まれたパートに伝えねばならない。しかしながら、そのコンテナパートは、その組み込まれたパートのアクティブシェイプの外側の領域で発生するイベントには自由に応答する。

フレームのもう１つの属性は、そのグループタグである。フレームのグループタグは、数字のような識別子であり、それは、与えられたコンテナでの関連するフレームセットをユニークに識別する。コンテナパートは、そのグループタグを用いて、そのグループの種々のフレームの幾何学的な関係を制御する。

そのフレームは、組み込むパートとその組み込まれたパートが、互いの内部制約をできる限り知らないようにして、文書構造を互いに調整できるようにしている。組み込むパートとその組み込まれたパートとの間で発生する調整の例は、図５Ｃ－図５Ｆに示されている。図５Ｃの例では、フレームシェイプは、コンテナで変更されている。この手順は、ユーザからのコマンドで始まり、例えば、フレームのサイズを変更する。ステップ５０１では、組み込むパートは、その組み込まれたパートのために、新たなフレームシェイプを決定し、その新しいシェイプのフレーム（ＦＲＡＭＥ）を通知する。これに応じて、ステップ５０２では、その組み込まれたフレームは、組み込むフレームの座標空間からのシェイプを組み込まれたフレームのそれに変換する。この新しいシェイプは、フレームシェイプとして格納され、その組み込まれたパート（ＰＡＲＴ）に伝えられる。ステップ５０３では、その組み込まれたパートは、自分自身を新しいフレームシェイプに調整する。例えば、テキストエディタが文字のレイアウトのためのマージンをリセットするかもしれない。もし、使用中シェイプ、及び、アクティブシェイプがフレームシェイプのマージンと異なっているならば、その組み込まれたパートもまた、これら新しいシェイプを計算する。その時に、制御は組み込むパートに戻り、ステップ５０４では組み込むパートは、新しいフレームシェイプに対応するためにフレーム（ＦＲＡＭＥ）のクリップシェイプを変更し、その新しいクリップシェイプのフレームを通知する。

図５Ｄは、組み込まれたパートが新しいフレームシェイプを要求するときに発生する手順を図示している。この要求は、内容が正しくレイアウトされるために付加的な空間を必要とする組み込まれたパートの内容の編集結果として、もたらされるかもしれない。ステップ５０５では、組み込まれたパートが新しいフレームシェイプを必要とするという決定がなされ、新しいシェイプのための表示フレームに要求が伝えられる。古いシェイプは、後

で必要とされる場合に備えて、組み込まれたパートによって保存される。ステップ５０６では、表示フレームは、組み込まれたパートの座標空間からのシェイプをコンテナフレームの座標空間でのものに変換し、その要求を組み込むパートに伝える。ステップ５０７では、コンテナパートは、その要求への返答方法を決定する。それは、望まれた形に対する要求を認めるかもしれないし、或は、より小さい形が要求されていると決定するかもしれない。フレームシェイプが一度決定されたなら、そのフレームに関するクリップシェイプもまた変形され、その許されたシェイプに関して、通知がフレームに与えられる。ステップ５０８では、表示フレームは、組み込むフレームの座標でのシェイプを組み込まれたフレームでのシェイプに変換する。その新しい形は、フレームシェイプとして保存され、そのフレームシェイプは、組み込まれたパートに伝えられる。ステップ５０９では、組み込まれたパートは、その表示フレームの使用シェイプとアクティブシェイプを変更して、新しいフレームシェイプに合わせるようにする。

図５Ｅは、コンテナパートがクリップシェイプを変更する場合の手順を図示している。図５Ｂにおいて、例えば、クリップシェイプを変更する必要は、表示フレームによって囲まれた領域の一部が異なるパートの内容に重なるようになってしまうようにするといったユーザの動作の結果として、発生するかもしれない。ステップ５１０では、組み込むパートは、表示フレームのために新しいクリップシェイプを決定し、この新しいシェイプのフレームを通知する。ステップ５１１では、表示フレームは、その通知を組み込まれたパートに伝える。これに応答して、ステップ５１２では、組み込まれたパートは、その内容が適切にレイアウトされるように変更することで、それ自身を新しいクリップシェイプに合わせる。

図５Ｆは、組み込まれたパートが、その使用中シェイプを変更する場合の手順を図示している。再び、そのような変更の必要は、ユーザによってそのパートの内容を編集するときに、発生するかもしれない。ステップ５１３では、組み込まれたパートは、新しい使用中シェイプをその表示フレームのために取り上げ、この新しいシェイプのフレームを通知する。ステップ５１４では、その表示フレームは、組み込むパートにその通知を伝える。これに応答してステップ５１５では、組み込むパートは、それ自身を新しい使用中シェイプに合わせる。例えば、組み込むパートは、その内容がレイアウトされる仕方を変更して、新しいシェイプにそって以前とは異なるようにするかもしれない。

従って、フレームは、組み込まれたパートと組み込むパートとの間での必要とされる空間割り当てを調整するデータ構造である。種々のシェイプの属性を用いることで、フレームは、その組み込むパートや組み込まれたパートが、各々のパートが他のパートの内容や振る舞いに関しての詳細な情報をもつ必要なく、効率的に利用可能な空間を利用できるようにしている。従って、複合文書が非常に柔軟に組み合わせられるのである。

パートに関するエディタや他の操作子は、従来のコンピュータシステムにおけるアプリケーションプログラムに類似している。パートの内容を表示し、そして、適切な場合には、その内容を変更修正するユーザインタフェースを与えるために必要な機能を提供しているのはソフトウェア要素である。それには、メニュー、コントローラ、ツールパレット、他のインタラクション技術が含まれるかもしれない。他のパートを含むことが可能なパートに関し、そのエディタは、その内容が表示される場合には、含まれるパートを考慮する。例えば、テキストエディタは、そのパートの内容を包み込み、それが組み合われた図形パートの使用中シェイプでは現れないようにする。

従来のアプリケーションで提供される機能を用いるために、プログラムが立ちあげられるか或はブートされねばならない。言い換えると、それは、システムのアクティブなメモリ、例えば、ＲＡＭ１６に格納されている必要がある。同じような方法で、パートに関するエディタは、システムのメモリ或はＣＰＵがアクセス可能なメモリで実行されねばならない。これを行うために、エディタは、システムにインストールされている必要がある。例えば、それは、ハードディスク２０のフォルダに格納され、関連パートがアクセスされたときに呼び出される。システム資源を浪費しないために、全てのインストールされたエディタは、常時システムメモリにはロードしなくとも良い。むしろ、それらは必要な時に、

立ちあげることができる。例えば、パートに関するエディタは、そのパートが文書内にドラッグされたとき、その時が編集機能がおそらく必要な時であるので、立ちあげればよい。

エディタや他の操作子を立ちあげる過程は、図６のフローチャートに図示されている。その図を参照して説明すると、ステップ６０１では、システムはアクティベートされるべきパートを探す。このステップのアクセスは、マウスボタンが押されたときにいつでも生成されるイベントフラグに応答してなされる。或は、このアクセスは、スクリプトでのステップでも良い。一度、パートのアクティベーションが検出されると、システムはそのパートに関連するエディタを識別する（ステップ６０２）。この点、夫々のパートは、それに関連した一定の特性を有している。これらの内の１つは、そのパートに関するエディタを指示するポインタである。２つの他の特性は、例えば、それが文字であるか、図形であるか、帳票であるかなどの、そのパートのカテゴリと、そのタイプである。タイプは、そのパートの内容に関するフォーマットの指標となる。例えば、１つの文字パートは、それが同じカテゴリ（文字）に属していても、“ワードプロセッサＡ”タイプのもので有り得るし、別のタイプは“ワードプロセッサＢ”で有り得る。しばしば、パートのタイプは、パートの内容で用いられた最後のエディタによって決定される。一般的に言って、カテゴリとは、１セットのパートのタイプであり、カテゴリが、与えられたパートに適用可能であるパートエディタやビューワのセットを決定する。パートの特性は、その内容とともに保存される。例えば、全てのパートの特性は、そのパートの内容とともに格納される“特性シート”に含まれている。

一度、エディタ（或はパートのタイプ）がステップ６０２で認識されたなら、システムはそのエディタが既にメモリ上で実行しているかどうかを判別する（ステップ６０３）。もし、そうであれば、システムはメインルーチンに戻り、ユーザからのさらなる動作を待つ。もし、エディタがまだ実行していないなら、システムはステップ６０２で識別されたエディタがシステムにインストールされているかどうかをチェックする（ステップ６０４）。もし、ステップ６０２で識別されたエディタがシステムにインストールされているなら、そのエディタはステップ６０５で立ちあげられ、パートの内容を編集できるようにされる。

しかしながら、パートエディタが現在システムにインストールされていないという場合も有り得る。この場合、ステップ６０４での決定は“ネガティブ”となり、これに応答して、システムはユーザがパートのカテゴリに関して好適なエディタを選択したかどうかを判別する（ステップ６０６）。即ち、ユーザは、好適な、或は、デフォルトのエディタをパートの夫々異なるカテゴリに関して選択することができる。従って、もし、ユーザが特定のワードプロセッサの機能コマンドに良く親しんでいるなら、そのプロセッサが文字カテゴリの全てのパートに関しての好適なテキストエディタとして選択される。もし、好適なエディタが選択されたなら、システムはその好適なエディタがシステムにインストールされているかどうかをチェックする（ステップ６０７）。そして、もし、そうであれば、ステップ６０８でそれは立ちあげられる。

もし、好適なエディタが指定されなかったなら、或は、その好適なエディタがシステムに存在しなかったなら、ステップ６０９では、パートのそのカテゴリについて何かのエディタがシステムに存在するかどうかのチェックがなされる。もし、何のエディタもなければ、適切な警告がユーザに与えられ（ステップ６１０）、そして、システムはバックグラウンドルーチンに戻る。もし、そのカテゴリについて１つ以上のエディタが利用可能であれば、それらはリストアップされ、ユーザはその１つを選択するように促される（ステップ６１１）。ユーザによるその選択に応答して、選択されたエディタが立ちあげられ（ステップ６１２）、それは、パートのそのカテゴリについての好適なエディタとして保管される（ステップ６１３）。システムは、それから、バックグラウンドルーチンに戻り、ユーザからのさらなる入力を待つ。

ユーザは、いつでも、例えば、適切なメニューコマンドを通して、パートについて指定されたエディタを変更する機会が与えられている。このコマンドを実施することによって、

ユーザは、パートの特性の１つとして格納されているポインタを、システムにインストールされている異なるエディタの位置に変えるようにできる。メニューコマンドは、また、ユーザが、例えば、文書レベル（文書の全ての文字パートは同じエディタをもつ）やグローバルレベル（システムにわたる）といった異なるレベルで好適なエディタを選択できるように備えられている。ユーザがメニューにアクセスし、パートのエディタを選択したり、変更したりするときにはいつでも、パートのカテゴリについて適切なインストール済みのエディタのセットだけがユーザに選択させるために表示される。

もし望むなら、システムはメモリ上で実行しているエディタ各々が使用された最新時刻の記録を保存できる。もし、エディタが所定時間の間に使用されなかったなら、システムは自動的にそのエディタをメモリから消去し、これによってシステム資源の浪費を防いでいる。また、文書がクローズされたとき、その文書のパートに関連したエディタもまた、それが現在オープンされている他のパート、例えば、他の文書に属していないならば、クローズされる。

パートの自律的な性質のために、種々のパートのコンテナは、決して特別なパイプに限定されるものではない。例えば、過去において、文字文書は特定のデータフォーマットである場合にのみ図形や他の内容を含むことができた。この種の制約は、本発明のアーキテクチュアによって取り除かれる。基本的に、どんなパートも他のパートに対しては、“ブラックボックス”となっている。コンテナパートはその中に含む他のパートの内部構造やセマティックスについての情報をもつ必要はない。コンテナパートは、単に、それが含む他のパートについてはラッパとしてのみ機能する。この種のアーキテクチュアは、複合文書のコンパイルや編集を非常に楽なものにしている。

さらに説明を続けるが、文書はデスクトップそれ自身の上に位置したアイコンによって表現される。図７Ａにおいて、図示されたデスクトップは、文字文書を含むウィンドウ７９を含んでいる。また、デスクトップ上にあるものとして、図形文書のアイコン８０がある。ウィンドウとアイコンの各々は、夫々のパートを表現している。ウィンドウ内に位置した文字文書の場合には、パートはフレームの形で表現されている。そして、そのフレームの境界はウィンドウ自身の境界と接している。本発明に従えば、アイコンによって表現されたパートは、ダイアログボックスなどのようなものを介する必要なく、直接ユーザによって操作され、文字文書の中に置かれる。この操作は、この後、図７Ａ－図７Ｅ、及び、図８Ａ－図８Ｂのフローチャートを参照して説明する。

アイコン８０によって表現される図形文書をウィンドウ７９内の文字文書本体に置くためには、ユーザは、まず、そのアイコン上にマウスのポインタを位置づけ（ステップ８０１）、そして、マウスボタンを押してそのアイコンを選択する（ステップ８０２）。これに応答して、システムはそのアイコンの表示を高輝度にし、それが選択されたという事実を示す（ステップ８０３）。図７Ａの例では、アイコン８０の高輝度化は、そのアイコンの絵とタイトルの反転表示によって示されている。

そのマウスボタンを押さえたまま、ユーザはそのカーソルを図形パートの所望の位置にまで移動させる（ステップ８０４）。図７Ｂに図示されているように、選択されたアイコンは、そのカーソルの動きに合わせて移動する。カーソルに合わせたアイコンの移動は、“ドラッグ”操作として知られている。そのアイコンが所望の位置にまでドラッグされると、アイコンのゴーストアウトライン８１がデスクトップ上には残り、その元々の位置を示す。或は、その代わりに、そのゴーストアウトラインがカーソルとともに移動して、ステップ８０５に図示されているように、完全な形のアイコン表示がその元々の位置に残つていても良い。

このアイコンのドラッグの間、システムはそのカーソルがウィンドウに入ったかどうかを判別する（ステップ８０６）。アイコン８０がウィンドウ７９の内側に位置すると、ウィンドウそれ自身が高輝度表示に変わる。そのシステムがドラッグされたアイコンがウィンドウ内のフレームに入ったことを検出すると（ステップ８０７）、そのフレームは適当に高輝度表示され、そのフレームがドラッグされたオブジェクトを受け付けたことを示す（ステップ８０８）。もし、そのアイコンがドラッグを続け、そのフレームの外側にでてし

まうなら、そのフレームの高輝度表示はなくなる（ステップ８０９）。逆に、複数のフレームが互いに内側に入れ子になっているなら、順々に入れ子になっているフレーム各々は、そのアイコンがその内側にドラッグされていくにつれ、アクティブになる。このようにして、システムは、アクティブになるフレームと、カーソル制御ボタンが放された場合にアイコンがドロップされる場合とに関してフィードバックを備えている。

一旦、ドラッグされたアイコンが所望の位置にきたなら、ユーザはマウスボタンを放す（ステップ８１０）。この時点で、システムはそのパートに関する好適な表示方法を決定する。例えば、それが、アイコンとして表示されるべきか、或は、フレームとして表示されるべきかなどである（ステップ８１１）。図７Ａ－図７Ｅの例では、パートが文書の中に置かれているので、その好適な表示方法は、フレームとなっている。従って、図７Ｃに示されているように、そのマウスボタンが放されたときに、そのカーソルの位置に対応する場所を左上隅として図形パートはフレーム８６として表示される（ステップ８１２）。また、図７Ｃに示されているように、そのフレームは、太い点線の境界で、また、サイズ変更ハンドルが付加されて表現され、それが、選択されたパートであることを示している。

逆の操作が、そのパートを文書からデスクトップに戻すために実行される。さらにその上、そのパートをコピーし、その内容がその文書内に残るようにし、一方でそのパートの別の表示がデスクトップに置かれるようにすることもできる。これを行うために、ユーザは、そのフレームをドラッグしながら、キーボード上のコントロール或はオプションキーのような機能キーを押す。マウスボタンの押下とともに同時に機能キーを押下することで、システムにコピー動作が実行されることを指示する。その時、マウスボタンを押さえつづけながら、図７Ｄに示すようにユーザがフレームをデスクトップまでドラッグする。ここで、図形パートの内容８８が文字文書の中に残り、そのパートの第２のコピーがフレーム８６に現れる。一旦、フレームがデスクトップ上の所望の位置に移動したなら、マウスボタンを放し、システムは再びパートに関する好適な表示方法を決定する。この場合、そのパートはデスクトップに位置しているのであるから、その好適な表示はアイコンである。従って、パートのアイコン表示８０が、図７Ｅに示されるようにデスクトップに表示される。

本発明のシステムにおいて、パートは直接にデスクトップと文書との間で操作されるのであるから、デスクトップのメタフォーはより良く保存される。即ち、ユーザは、ダイアログボックスなどに書き込むといった中間的な操作を行うことなく、デスクトップから所望のパートをピックアップし、それを直接に文書の中に置く。その結果、新しい内容を文書内に持ち込むために必要な時間や数多くのステップを減らすことができる。

デスクトップと文書との間でのオブジェクトをドラッグしたりドロップする機能をシステムがインプリメントした場合の１つの方法に関するさらに詳しい情報は、１９９３年３月３日出願の、出願番号＿＿＿＿＿＿＿＿、発明者がロバート　Ｇ．ジョンソン　ジュニア、マーク　Ｌ．スターン、デビット　Ｌ．エバンス、名称が“コンピュータ制御表示システムにおけるアプリケーションプログラムとファイルシステムとの間の改良型データ操作装置と方法”という発明に説明されている。その開示内容はここで参照のために、本願に組み込まれている。

さて、上述のように、パート各々はある特性をもち、その特性はその好適な表示で参照される。この特性は、与えられたパート中に含まれるパートが通常は、従来よりフォルダと文書ウィンドウの両方に関係づけられている振る舞いを得るために、アイコンとして表示されるか、或は、フレームとして表示されるかを決定する。一般的に言って、全てのパートは、デスクトップパートか或は文書パートかに分類される。デスクトップパートは、フォルダ、プリンタ、或は、メールボックスのような他のパートを含むことか、或は、そこで動作することが、その目的である。デスクトップパートに含まれるパートに関する好適な表示は、アイコンのようなものである。文書パートは、最終的には、文字文書や図形文書のような用紙或は他のハードコピー上にプリントされることが、その目的となるものである。文書パートに含まれるパートに関する好適な表示は、フレームのようなものである。従って、図７Ａ－図７Ｃに図示されている動作を参照すると、図形パートは、それがデ

スクトップにある時には、アイコンとして表示され（図７Ａ）、しかし、それが文書の中に置かれるときにはフレーム表示に変更される（図７Ｃ）。同様に、そのパートは文書からデスクトップに移動すると、その表示はフレームからアイコンに変化する（図７Ｃ－図７Ｅ）。もちろん、ユーザには、例えば、キーボードやメニューコマンドを通して、どんなパートの表示でも変更できるオプションが備えられている。

上述したように、ステーショナリパートは、それが、ただアイコンで表示されるという点で異なる。それは、１つのデスクトップパートから別のパートに移動できる。従って、例えば、フォルダ内の普通のパートのように扱うことができる。しかしながら、ステーショナリパートを文書パートに移動させようと試みても、ステーショナリパートのコピーが代わりに、文書パートに置かれ、そのコピーがフレームとして現れる。

全体としてパートを移動させることに加えて、ユーザはパートの内容のいくつか或は全てを選択し、移動できる。パート各々は、それ固有のタイプの内容を有している。例えば、文字パートはテキスト文字を含んでいるし、図形パートは図形要素などを含んでいる。ユーザは、１つのパートの固有の内容のいくつか、即ち、ドナーパートを選択し、これらを異なるパート、即ち、行き先パートに移動したりコピーしたりできる。この種の動作が発生するときに３つのシナリオが有り得る。即ち、（１）ドナーパートと行き先パートとは両方とも同じ固有のタイプのデータを含んでいる。（２）ドナーパートと行き先パートとは異なるタイプのデータを含むが、行き先パートはドナーパートから受信するデータのカテゴリを組み入れることができる。（３）ドナーパートと行き先パートとは異なるタイプのデータであり、行き先パートはドナーパートから受信するデータのカテゴリを組み入れるようには調整されていない。これら３つの場合の夫々について、図示した例を参照して以下に説明する。

最初のシナリオの場合、ドナーパートと行き先パートとは同じタイプのパートである。例えば、それらの固有の内容は、テキスト文字であるかもしれず、それらは、同じワードプロセッサによって生成されたかもしれない。図９Ａは、２つの文字文書９０、９２が夫々、デスクトップ上の２つのウィンドウにある例について図示している。左側の文書９０の文字の一部９４が選択されており、このことが反転表示で示されている。この選択された文字は、右側のウィンドウの文書９２にコピーされることになっている。従って、ユーザが機能キーを押しながらマウスボタンを押下し、もし必要なら、選択された文字９４のコピー９５を図９Ｂに示されているように、右側の文書９２にドラッグしてもってくる。一旦、カーソルが第２の文書９２に入ると、この文書は高輝度表示になる。左側の文書９０は今やアクティブではないので、それは文書９２の後ろに置かれ、そのタイトルバーはもはや高輝度表示ではなくなる。

一旦、コピーされたテキストが所望の位置にくれば、ユーザはマウスボタンを放し、図９Ｃに図示されているように、その文書にそのテキストを置く。第１の文書から第２の文書にコピーされたテキストは、第２のテキストの固有の内容と同じタイプであるので、そのテキストは第２の文書の内容に組み込まれる。言い換えると、選択されたテキストが一旦、第２の文書に置かれると、それはもはやそれ自身の独立した主体性をもたなくなる。むしろ、行き先の文書の固有の内容の一部となる。従って、行き先の文書の元々の内容のために用いられたのと同じエディタで、それは編集できる。

上述した第２のシナリオの場合、ドナーパートと行き先パートの固有の内容は異なるタイプであるが、行き先パートはドナーパートに含まれるカテゴリの情報を取り扱うことができる。図１０Ａにおいて、テキストエディタの能力を十分に用いて生成された文書９０から選択されたテキスト９６は、ダイヤログボックス９８にコピーされることになっている。この例において、選択されたテキスト９６は、ある形式に合わせられる。即ち、それは太字の属性をもっており、また、イタリックフォントとなっている。しかしながら、ダイヤログボックスはだた普通のテキストを取り扱うだけである。

前の例と類似した方法で、選択されたテキスト９６のコピーが文字文書からダイヤログボックスにドラッグされる。そのカーソルがダイヤログボックスに入ると、それは図１０Ｂに図示されているように表示の最上部に移動する。一旦、選択されたテキストのコピーが

、ダイヤログボックスの適切な位置にくると、ユーザはマウスボタンを放し、これを適切に“ドロップ”する。この時点で、システムはそのドロップされたテキストがダイヤログボックスの固有の内容とは同じタイプではない、即ち、普通の書式ではなくある形式をもっている、ことを判別する。しかしながら、それは、文字であるので、互換性のあるカテゴリに属している。従って、システムはある書式をもったテキストを、そのスタイルの属性を取り除くことによって、普通のテキストに変換し、それから、それを、図１０Ｃに示されているように、固有の内容の一部としてダイヤログボックスの中に組み入れる。さて、オリジナルの文書９０で２行にわたってラップされているコピーされたテキストは、例えば、選択されたテキストのコピーには存在したキャリッジリターンを除去することで、ダイヤログボックス９８の中で変換過程の一部として１行に再ラップされる。

上述の第３のシナリオの場合には、ドナー文書の固有の内容は、行き先パートと比較して、異なるタイプであり、互換性のないカテゴリにある。図１１Ａにおいて、テキスト文書９０から選択されたテキスト９４は、図形文書１００にコピーされることになっている。再び、ユーザは、テキスト文書９０から選択されたテキストのコピーを図形文書１００にドラッグする。そのカーソルが行き先文書に入るや否や、図形文書１００は高輝度表示となる（図１１Ｂ）。そのテキストのコピー９５が一旦、適切な位置にくれば、ユーザはマウスボタンを放し、そのテキストを図形文書にドロップする。この場合、選択されたもの、即ち、テキストのカテゴリは、行き先文書の固有の内容、即ち、図形要素とは、互換性をもっていない。従って、システムはドナー文書と同じタイプ（文字）の新しいパートを生成し、選択されたもののコピーをそのパートに挿入し、行き先文書にその新しいパートを組み込む。このようにして、新しいパートは、行き先文書の中でそれ自身の独自性を保持することなる。それは，それ自身の固有の内容、即ち、テキスト文字と、それに関連したテキストエディタとを有するようになる。その新しい生成物は、分離したパートであるので、図形文書内には、図１１Ｃに図示されるように、フレーム１０２として留まる。フレーム１０２は、そのフレームの回りの境界によって図示されてるように、選択されたオブジェクトとして留まる。

選択されたテキストのコピーが、文字文書から図形文書に移動するとき、デスクトップのメニューバー１０４は、図１１Ａと図１１Ｂとに図示されてるように、文字文書に関連したコマンドを留めている。しかしながら、そのコピーされたテキストが図形文書に一旦、ドロップされたなら、図１１Ｃに示されているように、メニューバー１０４で利用可能なコマンドはテキストエディタに合わせたものから、図形エディタに合わせたものに切り替わる。さらに、図形エディタに関連したツールバー１０６がデスクトップに現れる。なぜなら、図形文書には今や、選択されたパートが位置しており、図形文書がアクティブな文書であるからである。しかしながら、もし、ユーザがフレーム１０２の中にカーソルを位置させ、文字や単語を選択するなら、ツールバー１０６が取り除かれ、メニューバー１０４のコマンドが文字の編集のために適切なものに変わる。

要約すれば、１つの文書から他の文書に内容が移動するとき、システムはその内容が行き先文書にその固有内容の一部として組み込まれるかどうか、或は、行き先文書の中で自律的なパートとして組み込まれねばならないかを判断する。このプロセスを実行するシステムの動作が、図１２のフローチャートには図示されている。ユーザが１つのオブジェクトを選択し、それを意図する行き先にドラッグした後に、システムは選択されたオブジェクトが元々存在したパートのタイプを判別する（ステップ１２０１）。例えば、選択されたオブジェクトは、普通のテキストであるかもしれないし、ある書式のテキストであるかもしれないし、図形要素などであるかもしれない。夫々のパートについて格納されている特性の１つは、それがどのタイプのパートであるかを識別し、そして、そのパートから移動される或はコピーされるオブジェクト各々もまた、その識別されたタイプであるとして認識される。パートのタイプが識別された後、そのタイプが行き先文書のタイプと同じであるかどうか判定される（ステップ１２０２）。もし、そうであれば、選択された内容は、行き先の文書に組み込まれるか或は挿入され、その固有の内容の一部となる（ステップ１２０３）。もし、選択された情報が、行き先文書のタイプと同じではない場合、システム

はそれが同じカテゴリに属しているかどうかを判別する（ステップ１２０４）。例えば、もし、選択された情報が、ある書式スタイルをもったテキストであり、その行き先の文書では普通のテキストであれば、それらは異なるタイプではあるが、同じカテゴリに属している。もし、選択された情報が行き先文書と同じカテゴリに属していれば、システムは選択された情報を行き先文書と同じタイプに変換する（ステップ１２０５）。例えば、あるワードプロセッサで生成されたテキストが、異なるワードプロセッサによって採用されているフォーマットに変換される。或は、図形文書はあるファイルフォーマットから別のものに変換される。一旦、その変換が完了すると、選択された情報は、行き先文書の内容に組み込まれる（ステップ１２０９）。もし、その選択された情報が、行き先文書と同じカテゴリに属していなければ、新しいパートが、その選択された情報がその内容となるように生成され（ステップ１２０６）、そして、その新しいパートがフレームとして行き先文書に組み込まれる（ステップ１２０７）。

パートの自律的な性質のため、パートは、種々のユーザの間での情報交換を非常に容易にする。文書、例えば、ルートパートは、幾つかの他のパートを含んでいる。ユーザはその文書全体を取り出すことができ、それを眺め、所望ならその個々のパートを編集できる。或は、個々のパートは夫々がデスクトップに置かれ、多くのユーザによって共有されるが、一方、その文書へのアクセスは全体として保護される。

これらのことをさらに詳しく述べると、パート１つ１つは限定されたアクセス権で保護される。例えば、幾つかの異なるパートを含む文書は多くの異なる種類の人々への配布が目的となっていることもある。それらのパートのいくつかは、ある種類の人々には公開されたくはない機密情報を含んでいるかもしれない。本発明の主旨によれば、これら個々のパートは、限定されたアクセス権で保護される。そして、その内容は適切なパスワードや或は他のアクセス手段を有した人々によってのみ回覧される。言い換えると、アクセス制限機能はパートの別の特徴的な性質である。

図１３において、図示されている文書１０８は、多くのパートから構成され、そのパート１つ１つは、破線で示されたフレーム内に含まれている。パートの１つ１１０へのアクセスは制限されている。例えば、そのパートは機密の帳票データを含んでいるかもしれない。このパートへのアクセスが制限されているので、その内容はそのフレームには表示されない。むしろ、そのフレームは不透明となっており、キー１１２のような適切なシンボルを表示し、パスワードなどのようなものがそれにアクセスするためには必要であることを示す。

文書１０８はシステムのユーザによってオープンされ、その概要レイアウトと機密ではないパートの情報を見ることができる。しかしながら、機密のパート１１０の中に含まれているデータは、そのパートにアクセスする適切なパスワードを有した人達だけが見ることができる。そのような人がそのパートを選択すると、その不透明カバーが取り去られ、その内容が表示される。

その内容はアクセスされないが、制限されたパートの他の特性は、そのパスワードを有していないシステムユーザに対しても、やはり、利用可能である。例えば、文書中のそのパートの位置は変更できるし、それを選択し、そのフレームの境界にあるサイズ変更ハンドルを移動させて、そのサイズを変更することもできる。同じように、そのアクセスが制限されたパートはどこにでもコピーすることができる。しかしながら、そのアクセス制限は、どこにそれがコピーされようとも、そのコピーされたものにも付加されたままである。制限されたアクセス権をもつパートは、それに組み込まれる他のパートをもつことができる。その結果、あるユーザが適切なパスワードを有しておらず、パートの内容にアクセスするならば、それに含まれる他のパートもまた、制限されて見る事はできない。一旦、パスワードが受け付けられたなら、そのパスワードは自動的に全ての組み込まれたパートにも伝えられ、それらの制限が解除されて、ユーザの手を煩わせる事なく見る事ができる。しかしながら、もちろん、その組み込まれたパートがそれに関連した異なるパスワードをもつことはでき、たとえユーザが１つの制限があるパートにアクセスできようとも、それに含まれる他のパートへのアクセスはさらに制限されていることもある。

従って、本発明は文書の内部レベルで文書の内容へのアクセスを制限できる。そのアクセス制限は、パート毎に指定することができ、パートのデータがどこに存在するかに係わらず、効力をもつ。このような手法によって、文書の中で特権が与えられていないパートは、文書全体にわたるアクセス制限を課すことなく、ユーザによってアクセス可能となる。多くの著者をもつ大きな文書において、パートパート単位にアクセスを制限できることは、その著者たちが文書を概観し、協力して作業を行うことを、非常に柔軟性に富んだものにする。

パート操作や複合文書のコンパイルを容易にするために、“パーツビン”と呼ばれる特殊なタイプのパートが備えられている。パーツビンの例が図１４に示されている。その図を参照して説明すると、文書がウィンドウ１２０でオープンされ、その文書に関するパーツビンが、ウィンドイドとして言及されている別の領域に現れている。そのウィンドイドは、ウィンドウ１２０に関連したものであり、もし、その文書ウィンドウがスクリーンの前面にもってこられたなら、ウィンドイドもともにスクリーンの前面にくる。

パーツビン１２２は、本質的には、他のパートに関するソースとして機能するユーザ定義のパレットである。それは、他のパートだけを含み、固有の内容はもたない。パーツビンの内部にあるパートは、“凍結”した特性をもっている。即ち、ユーザが通常カーソルをフレーム内に位置させながら、マウスボタンを押すと、そのカーソルの位置にある最小の要素が選択される。図１４において、通常の場合、もし、カーソルが帳票フレーム１２４の１つのセルに置かれ、そのカーソル制御ボタンが押下されたなら、そのセルの内容が選択される。さて、ユーザは異なるオブジェクト、例えば、実際に選択されたセルを含む帳票全体、を選択することを意図していたかもしれない。これを行うために、ユーザはフレーム内の別の領域、しかし、所望のパート内に組み込まれた他のパートの外側にカーソルを移動させなければならない。しかしながら、この要求は、パーツビンのパートを凍結させることで取り除かれる。この場合、ユーザはパートのどこにでもカーソルを位置させてマウスをクリックし、そのパート全体を選択する。従って、たとえ、パーツビン内にあるパートが他の組み込まれたパートを含むフレームとして表現されていても、それらは全体として扱われる。

パーツビン内にあるパートは、また、ロックされ、その内容が修正されないようになっている。パーツビンのロックされた特性は、ウィンドイドのヘッダのパッドロックのシンボル１２６によって示される。もし、ユーザがロックされたパーツピンの外にパートをドラッグしようとすると、そのパートのコピーが自動的に作られ、元々のパートはパーツビンの中に留まり、これによってビンの内容の完全性が保証される。ユーザが自分のパーツビンをカスタマイズできるようにするために、パーツビンのロックは、例えば、パッドロックのシンボル１２６をクリックすることで解除され、そのパートが修正される。

本発明のこの特徴のさらなる実施例では、どの文書も付随したパーツビンをもつことができる。“文書パーツビン”は、標準的なパートのセットが文書に結合されるようにしており、その結果、その文書のユーザはすぐに利用可能なパートをもつことができるようになる。あるコマンドを実行することで、そのユーザは文書パーツビンをウィンドイド或はドロワに表示させる。付随したパーツビンとともに文書は、そこに置かれた制限的な特性をもつことができ、その文書に置かれるパートだけがそのビンの中のものとなる。他のパートをその制限された文書に置こうとすると、その試みは拒絶される。その特性は文書のユーザがあるセットのパートとともにだけ作業することを保証し、これによって、その文書作成に関連した訓練、サポート、保守などを減らしている。例えば、データ入力フォームでのパーツビンは、その書式でデータを入力するために必要なパートのみを有する。その結果、訓練を受けていないユーザは、正しく用いることがでいないかもしれない予期しないパートに出くわすこともない。パートについてその制限は、また、組織が文書の内容をコントロールすることにも役立つ。

上述したユーザに対する利点に加えて、本発明のアーキテクチュアはソフトウェアの開発者にも種々の利点を備えている。例えば、帳票プログラムは元々は数字を処理するために設計されている。この分野で経験のある開発者は、この機能に関して最も効率的に時間を

用いることができる。しかしながら、ユーザが公式や数値のあるテキストを編集することができなければならないので、テキストエディタは、帳票アプリケーションの必要な部分である。テキストエディタを書かねばならないことは、帳票プログラムの開発者にとって、従来のプログラムとともに必要なことではあるが、難儀なことでもある。しかしながら、ユーザが選択したテキストエディタが帳票において分離独立した部分として利用可能であるような環境では、帳票プログラムの開発者にとってテキストエディタを書く事はもはや必要なことではなくなる。その開発者は、自分の時間を帳票プログラムの核の部分にいっそうふりむけることができ、以前は帳票の数多くの処理機能にとっては付帯的であった他の部分を開発するために費やさねばならなかった時間のために用いることができなかった時間を用いて、特別な機能などを組み込むことができるようになる。

さらに、そのソフトウェア開発者がさらに新しい機能を追加したり、或は、存在するバグを修正することを決めたときに、その仕事はさらに容易になる。特に、その開発者はプログラムの帳票部分にのみ集中すればよく、テキストエディタのような付帯的な要素に気を払う必要はない。

本発明は、ここで説明された、発明の基本となる原理についての理解をいっそう容易にするためになされた具体的な実施例によって限定されるものではない。例えば、テキストや図形は文書の要素を表現するために例として用いられただけである。パートの内容もまたここでの２つの具体例に限定されるものではない。むしろ、パートの内容は、どんな種類の媒体でも良く、これにはビデオや、音や、動画などを含んでいても良い。従って、本発明の範囲は、以下に付属したクレームによってのみ定義され、前述の説明によって定義されるものではない。そして、クレームの意味に矛盾のない全ての等価物が、その範囲に含められるものである。

【図１】

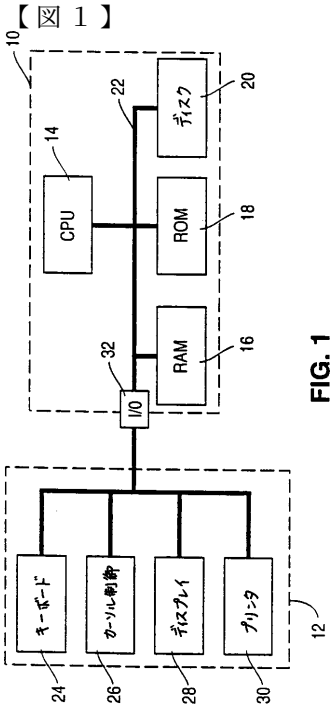

FIG. 1

【図２Ａ】

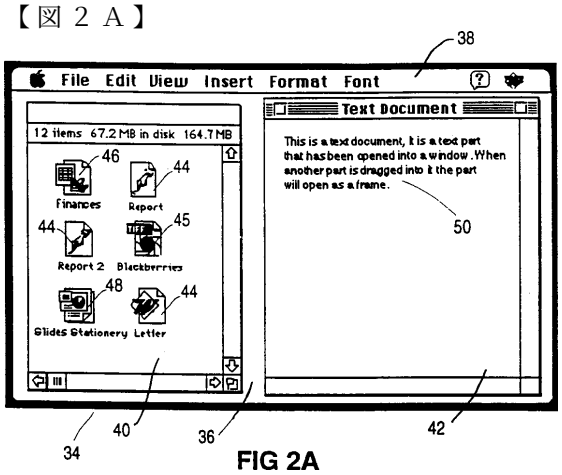

FIG 2A

【図２Ｂ】

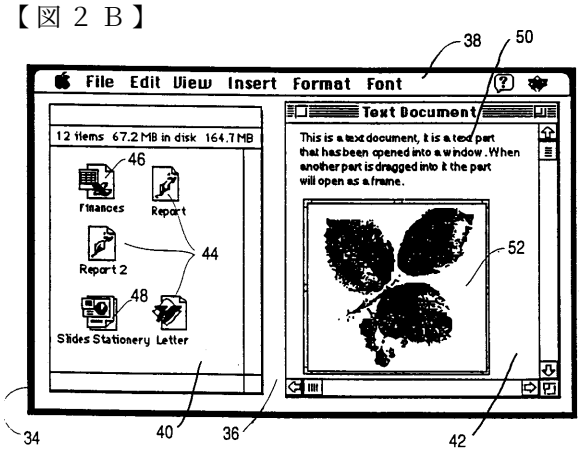

FIG 2B

【図３Ｂ】

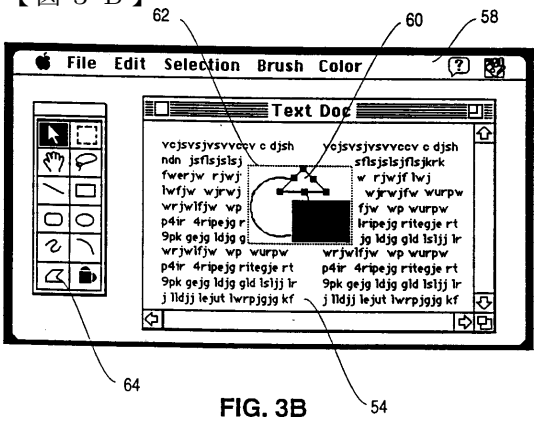

FIG. 3B

【図３Ａ】

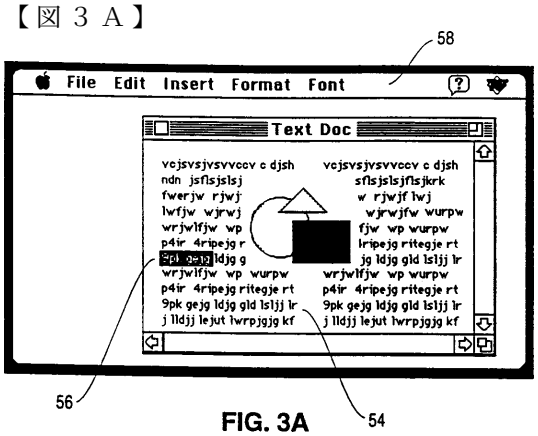

FIG. 3A

【図４】

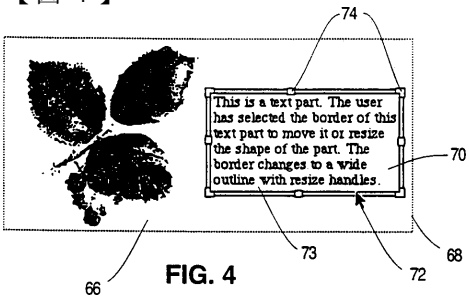

FIG. 4

【図５Ａ】

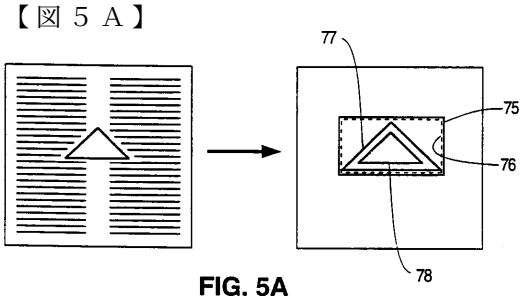

FIG. 5A

【図５Ｂ】

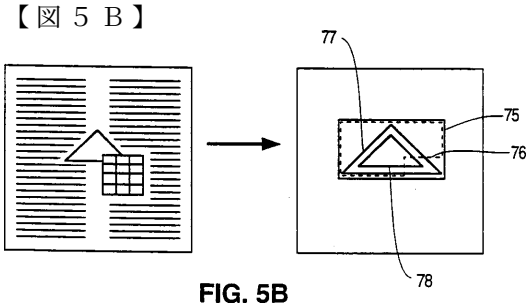

FIG. 5B

【図５Ｃ】

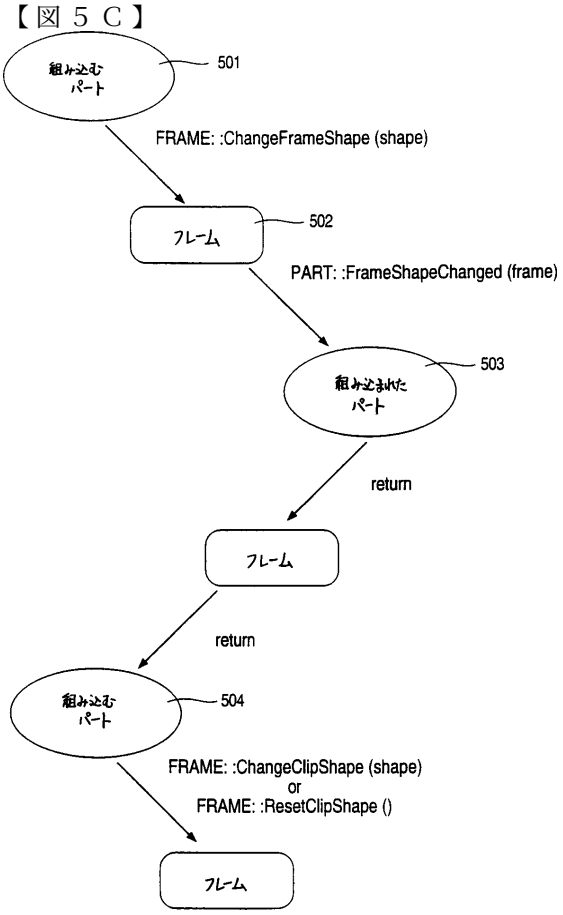

FIG. 5C

【図５Ｄ】

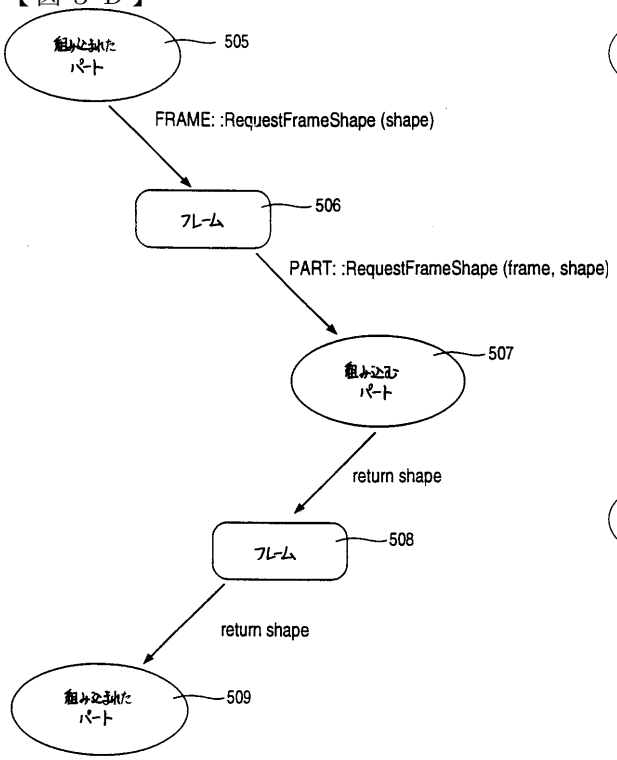

FIG. 5D

【図５Ｅ】

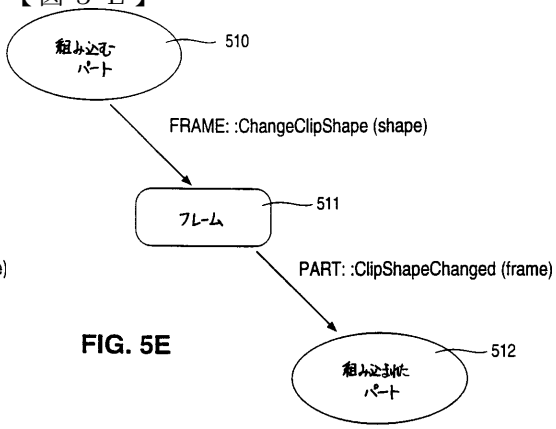

FIG. 5E

【図５Ｆ】

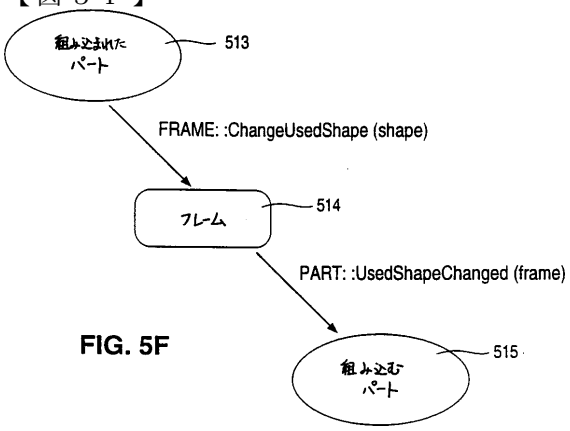

FIG. 5F

【図６】

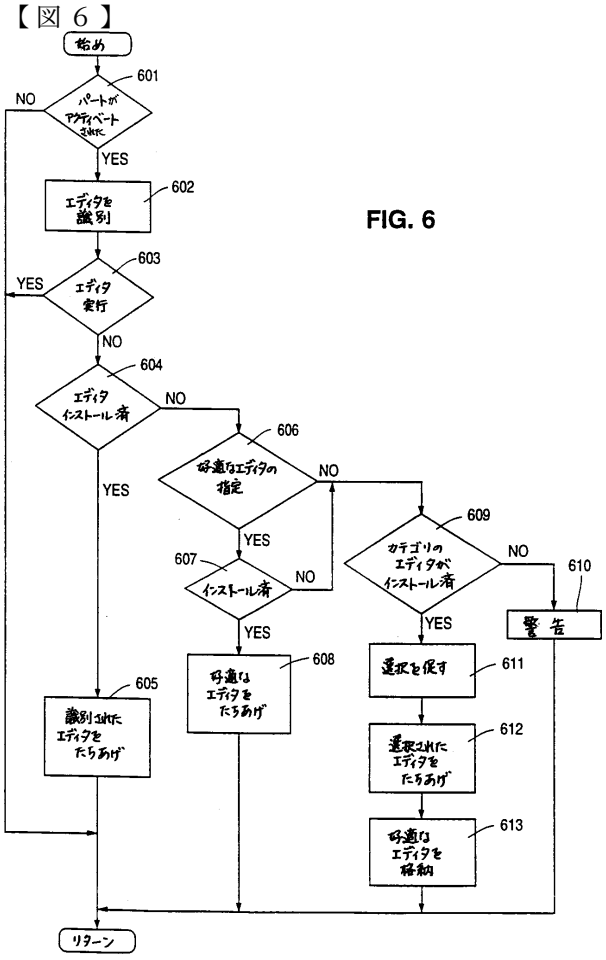

FIG. 6

【図７Ａ】

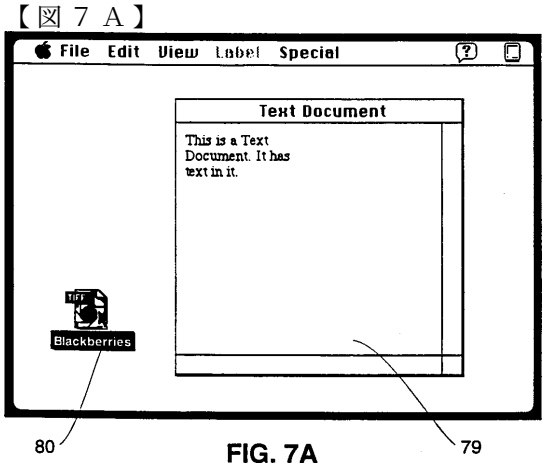

FIG. 7A

【図７Ｂ】

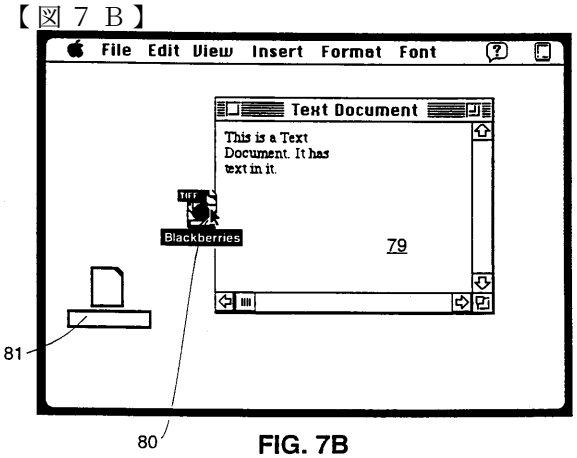

FIG. 7B

【図７Ｃ】

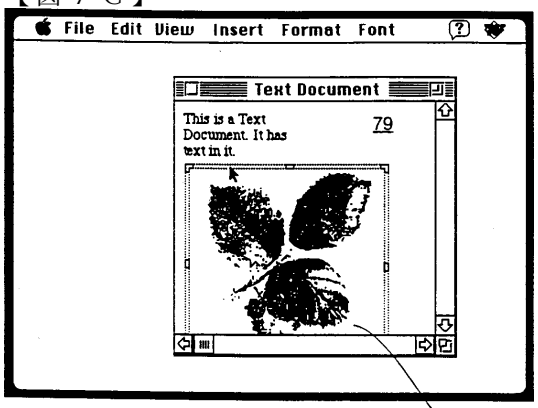

FIG. 7C

【図７Ｄ】

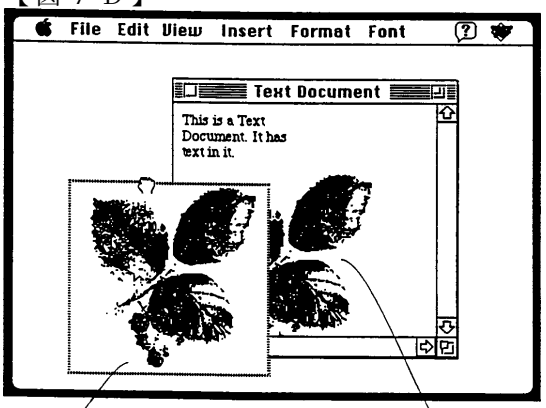

FIG. 7D

【図７Ｅ】

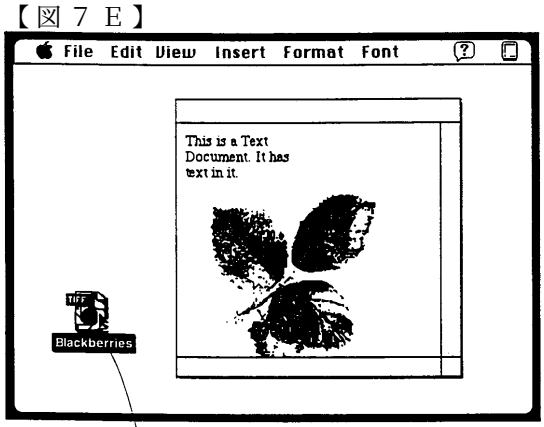

FIG. 7E

【図８Ａ】

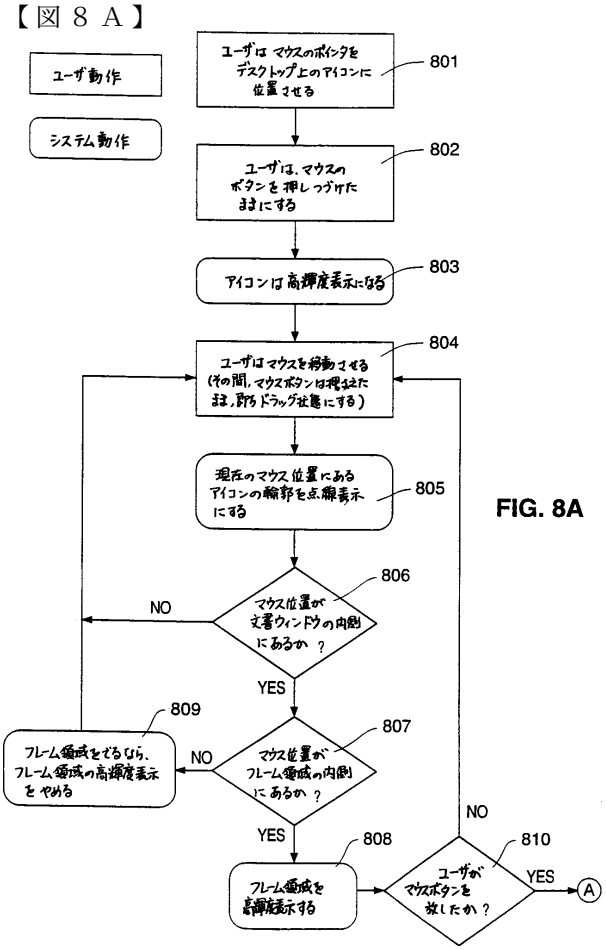

FIG. 8A

【図８Ｂ】

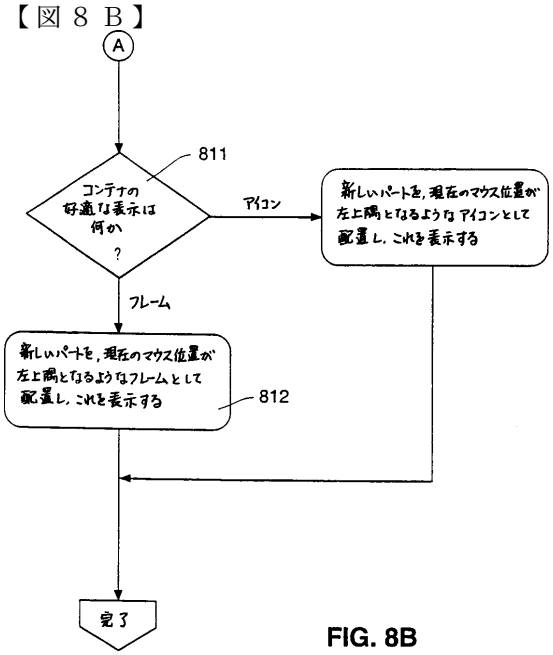

FIG. 8B

【図９Ａ】

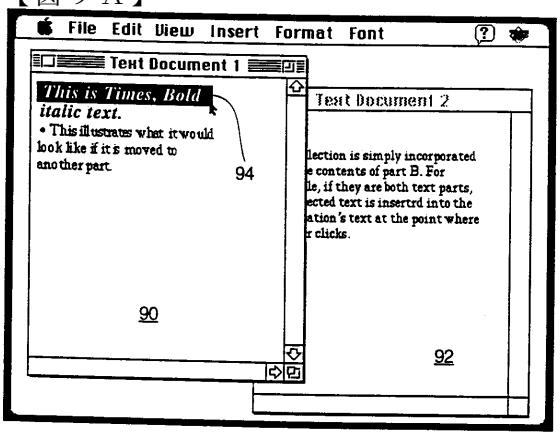

FIG. 9A

【図９Ｂ】

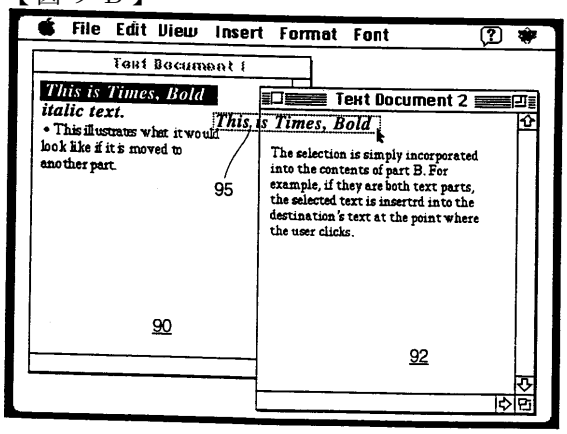

FIG. 9B

【図９Ｃ】

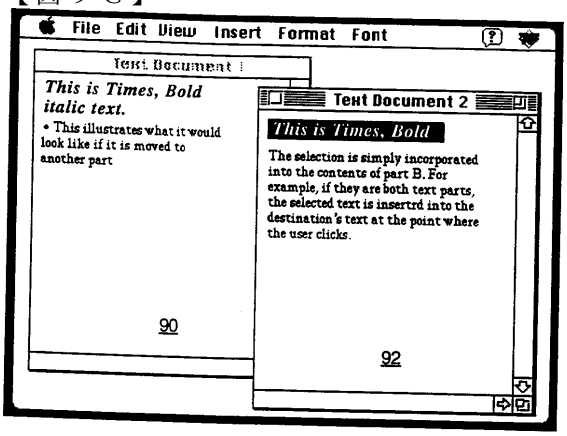

FIG. 9C

【図１０Ａ】

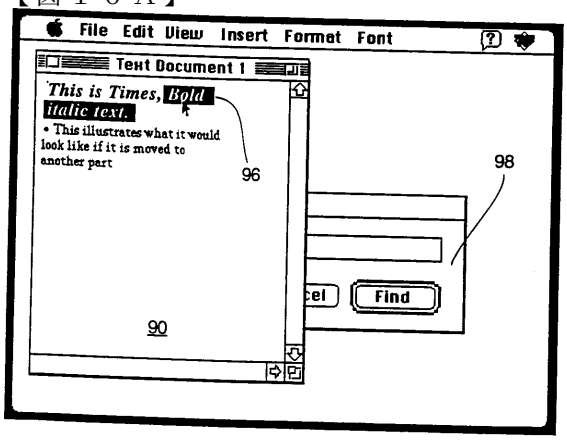

FIG. 10A

【図１０Ｂ】

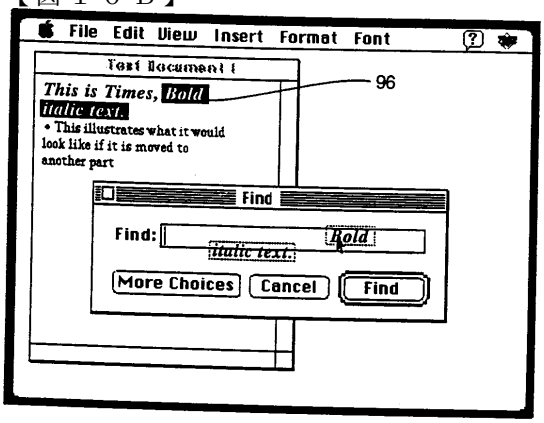

FIG. 10B

【図１０Ｃ】

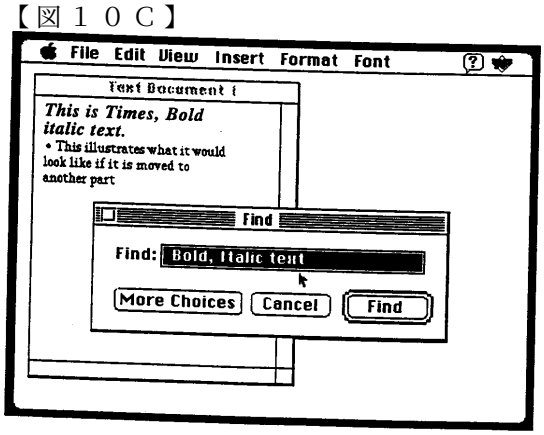

FIG. 10C

【図１１Ａ】

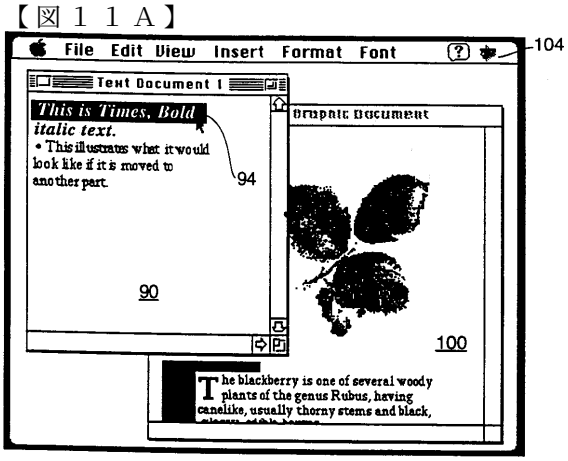

FIG. 11A

【図１１Ｂ】

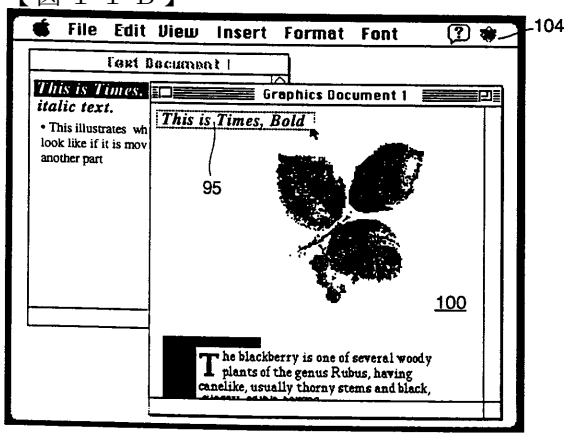

FIG. 11B

【図１１Ｃ】

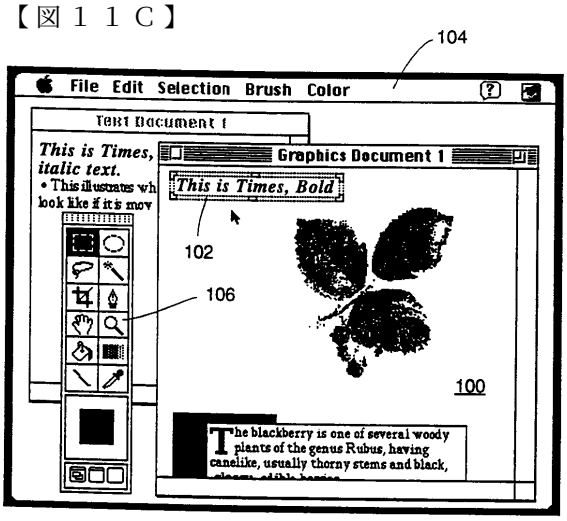

FIG. 11C

【図１２】

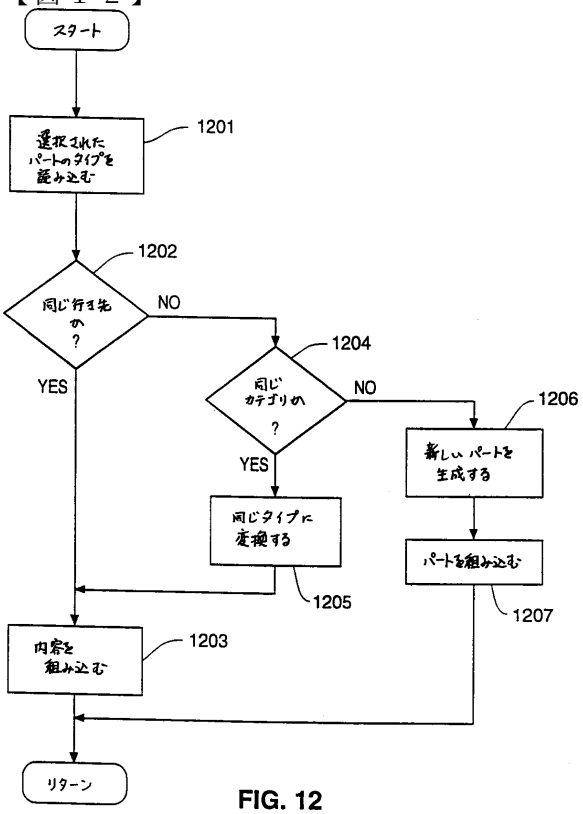

FIG. 12

【図１３】

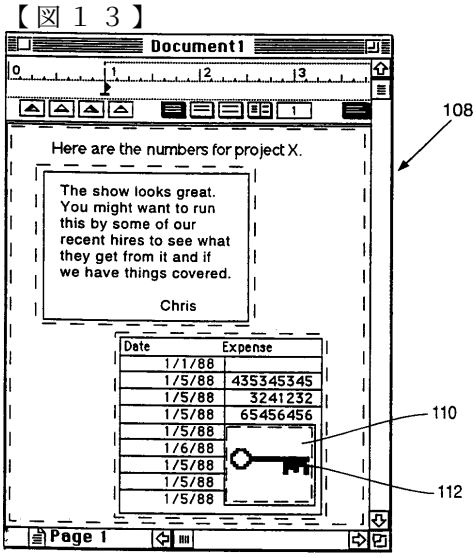

FIG. 13

【図１４】

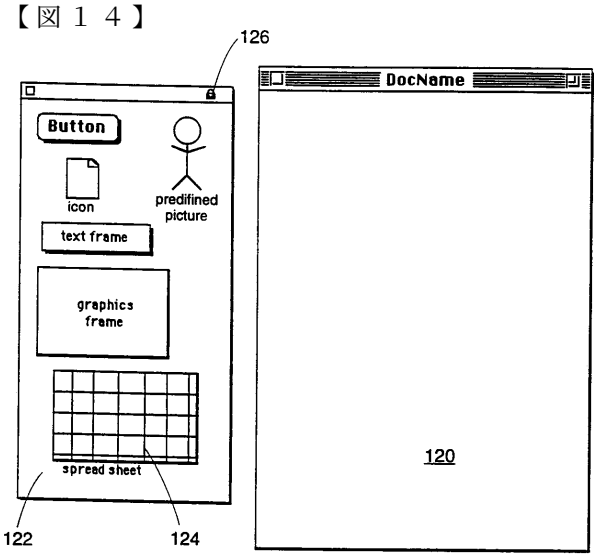
126
Button
icon
predifined picture
text frame
graphics frame
spread sheet
122
124
DocName
120

FIG. 14

フロントページの続き

(72)発明者 スターン，マーク，ルドウィッグ
アメリカ合衆国 カリフォルニア州 ９５０１４ クッパチーノ，ペンダーガスト アベニュー １９０３５
(72)発明者 シャフィー，ジェニファー
アメリカ合衆国 カリフォルニア州 ９５０６２ サンタ クルズ，１７ス ストリート ２１６０
(72)発明者 クリーガー，ジェフリー
アメリカ合衆国 カリフォルニア州 ９４００５ ブリスベン，ジョイ アベニュー ４９エー
(72)発明者 トンプソン，マイケル
アメリカ合衆国 カリフォルニア州 ９４０２５ メンロ パーク，ローブル アベニュー ９１５
(72)発明者 コリック，ジョージ
アメリカ合衆国 カリフォルニア州 ９５０１４ クッパチーノ，メリッサ コート １０４０４
(72)発明者 ジョーダン，ダニエル
アメリカ合衆国 カリフォルニア州 ９４１３１ サン フランシスコ，ノエ ＃３ １３７０
(72)発明者 ピエソル，カート
アメリカ合衆国 カリフォルニア州 ９５０７３ ソークエル，フェアウエイ ドライブ ３２８０
(72)発明者 カーボウ，デイビッド
アメリカ合衆国 カリフォルニア州 ９４０８７ サニーベール，チェシル ウェイ ６１５

審査官 長 由紀子

(56)参考文献 国際公開第９２／００８１９９（WO，Ａ１）
特開平０１－１４２８５９（ＪＰ，Ａ）
特開昭６２－０７２０５９（ＪＰ，Ａ）
特開昭６１－０６２１７０（ＪＰ，Ａ）
特開平０４－００３２４６（ＪＰ，Ａ）
特開平０１－２３４９６７（ＪＰ，Ａ）
特開平０１－２６３７６４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 17/21 - 26
G06F 3/14
G06T 11/80