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

特許第4021474号

(P4021474)

(45)発行日　平成19年12月12日(2007.12.12)　(24)登録日　平成19年10月5日(2007.10.5)

(51)Int.Cl.		F I		
G06F 19/00	(2006.01)	G06F	19/00	140
G06F 12/00	(2006.01)	G06F	12/00	547A
G06F 17/30	(2006.01)	G06F	12/00	547D
		G06F	17/30	170G
		G06F	17/30	180E

請求項の数 1　(全 39 頁)

(21)出願番号　特願平6-525548
(86)(22)出願日　平成6年5月6日(1994.5.6)
(65)公表番号　特表平8-510075
(43)公表日　平成8年10月22日(1996.10.22)
(86)国際出願番号　PCT/US1994/004991
(87)国際公開番号　WO1994/027237
(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4月4日(2001.4.4)
審判番号　不服2003-11641(P2003-11641/J1)
審判請求日　平成15年6月23日(2003.6.23)
(31)優先権主張番号　08/059,542
(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　591049538
アプル・コンピュータ・インコーポレーテッド
アメリカ合衆国　９５０１４　カリフォルニア州・カッパチーノ・マリアニイ　アヴェニュ・２０５２５
(74)代理人　100064621
弁理士　山川　政樹
(74)代理人　100067138
弁理士　黒川　弘朗
(74)代理人　100076392
弁理士　紺野　正幸
(74)代理人　100081743
弁理士　西山　修

最終頁に続く

(54)【発明の名称】マルチメディアデータを対話形式でデリバリする方法

(57)【特許請求の範囲】

【請求項１】

トピックを構成する複数のトピックオブジェクトと、パネルのオブジェクトと、前記パネルを構成するオブジェクトであるところの、テキスト、グラフィック、及び、ビデオデータ、並びに、制御、修飾子、及び、フォーマットの各オブジェクトとを蓄積するデータベースを記憶したデータ記憶装置と、処理資源を提供するプロセッサと、表示装置と、からなるコンピュータ制御表示システムでマルチメディアデータを対話形式でデリバリする方法であって、

ａ）特定のトピックオブジェクトを前記データベースから検索し、前記トピックオブジェクトをインスタンシエートするステップと、

ｂ）前記トピックオブジェクトのリンク情報に基づき、前記トピックにリンクする各パネルのオブジェクトをデータベースから検索し、前記各パネルのオブジェクトをインスタンシエートするステップと、

ｃ）前記パネルのオブジェクトにリンクされた前記パネルを構成するオブジェクトのリストに基づき、前記リストに対応するオブジェクトの記述子をデータベースから検索し、前記リストに対応するオブジェクトの記述子をインスタンシエートするステップと、

ｄ）以下のｉ）ないしvi）ステップを実施することによって表示すべきパネルを識別するステップと、

ｉ）前記トピックにリンクする各パネルのオブジェクトのうち、現在着目しているパネルのオブジェクトが少なくとも一つの前記修飾子オブジェクトを持っているかどうかを検査

するステップと、
ii）前記現在着目しているパネルのオブジェクトが前記修飾子オブジェクトを有していない場合、前記現在着目しているパネルのオブジェクトを表示すべきパネルのオブジェクトとして識別するステップと、
iii）前記現在着目しているパネルのオブジェクトが前記修飾子オブジェクトを有する場合、前記修飾子オブジェクトにかかわる条件をデータベースから検索し、前記条件をオブジェクトとしてインスタンシエートし、前記インスタンシエートされた条件のオブジェクトを評価するステップと、
iv）前記評価の結果が第１状態である場合、前記現在着目しているパネルのオブジェクトを表示すべきパネルのオブジェクトとして識別するステップと、
ｖ）前記評価の結果が第２状態である場合、前記現在着目しているパネルのオブジェクトをスキップするステップと、
vi）前記修飾子オブジェクトを持たないか、又は、前記修飾子オブジェクトの評価結果が前記第１状態であるパネルのオブジェクトが見いだされるまで前記トピックにリンクする各パネルのオブジェクトに対して上記ｉ）～ｖ）のステップを続けるステップと、
ｅ）前記表示すべきパネルのオブジェクトにリンクされた前記パネルを構成するオブジェクトのリストに基づき、前記リストに対応するオブジェクトの内容をデータベースから検索するステップと、
ｆ）前記オブジェクトの内容を前記表示装置のプレゼンテーションウィンドウのパネル表示領域に表示することにより前記表示すべきパネルを表示するステップと、
を有する方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明はデータの記憶及び検索の分野に関し、特に、教示(instrauctional)又は教育を目的として使用されるような対話型データデリバリ表示システムにおいて使用するためのマルチメディアデータを編成し且つ記憶することに関する。
【０００２】
【従来技術】
コンピュータシステムの使用範囲がますます広がるにつれて、マルチメディア利用の要望がさらに差し迫ってきている。マルチメディアとは、テキスト情報メディア，グラフィック情報メディア，ビデオ情報メディア及び音声情報メディアを統合して使用することを指す。マルチメディアを効率良く使用するためには、データをユーザに引き渡すためのメカニズムを開発しなければならない。たとえば、対話型教示アプリケーションを作成するために、その教示情報を含む様々な画面、すなわち、パネルが作成される。それらのパネルは対話型教示の状態に対応する順序で論理的に連係（リンク）され且つ表示されなければならない。
【０００３】
マルチメディアデリバリ能力を発揮する既知のシステムは２つある。すなわち、ＭａｃｒｏＭｅｄｉａ Ｉｎｃ．の製品であるＭａｃｒｏＭｅｄｉａ Ｄｉｒｅｃｔｏｒ及びＨｙｐｅｒＣａｒｄ（登録商標）。ＭａｃｒｏＭｅｄｉａ Ｄｉｒｅｃｔｏｒはスコア，ステージ及びキャストのメタファを使用して、対話型マルチメディアをデリバリする。スコア（時間線）は、キャストメンバ（マルチメディアオブジェクト）をステージ（表示装置）上に乗せるために使用される。視覚の上でオブジェクトをスコアの縦軸に沿って配置することにより、また、時間の上ではオブジェクトを横軸に沿って配置することにより、オブジェクトを層構造にすることができる。任意の時点でユーザが見るものはスコア中の１つの欄によって表わされる。デリバリの流れ（すなわち、教示）はスコアにより制御される。キャストメンバ及びスコアにスクリプトを付加して、デリバリの流れを変更することができる。スクリプトはデリバリの流れの外側での処理を要求する。さらに、その後の流れの変更はわずらわしい。

【０００４】
【発明が解決しようとする課題】
Ａｐｐｌｅ（登録商標）Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．の製品であるＨｙｐｅｒＣａｒｄはノートカード及びオブジェクトのメタファを使用して、対話型マルチメディア教示を受け渡す。視覚オブジェクトをカード上に配置する。次に、カードを順序付けする。デリバリの流れはカードの順序によって制御される。視覚オブジェクト及びカードにスクリプトを付加して、教示の流れを変更することができる。ＨｙｐｅｒＣａｒｄカードはＭａｃｒｏｍｅｄｉａ　Ｄｉｒｅｃｔｏｒのスコアにおける欄と同等である。ＨｙｐｅｒＣａｒｄオブジェクトはＭａｃｒｏＭｅｄｉａ　Ｄｉｒｅｃｔｏｒのキャストメンバと同等である。双方の製品の要素には共にスクリプトを付加できる。
【０００５】
従来の技術のシステムは様々な欠点を有する。まず、スクリプティングを経なければ、データデリバリの流れを変更できない。教示のデザインの中に内在して教示の流れを変更できることが望ましいであろう。さらに、教示の流れが始まってしまうと、バックトラッキングはない。
【０００６】
従って、従来の技術の上述の欠点を克服する対話型マルチメディアデリバリシステムを有することが望ましいであろう。
【０００７】
【課題を解決するための手段】
マルチメディアデータを対話形式で提示し且つ表示するデリバリエンジンを開示する。デリバリメカニズムはトピック及びパネルのメタファを前提としている。トピックは複数のパネルから構成されている。各パネルは、そのパネルを表示するときに使用されるマルチメディアデータを定義する。各パネルは、表示すべき次のパネルを確定するために使用される制御要素と修飾子要素をさらに含んでいても良い。制御要素はユーザからの情報を請求するために使用され、その後、その情報は表示すべき次のパネルを確定するときに修飾子要素によって使用される。
【０００８】
各々の要素は個別のオブジェクトとしてデータベースに記憶される。これにより、様々に異なるパネル又はトピックの間でオブジェクトを再利用できる。データのデリバリに先立って、トピックをセットアップしなければならない。セットアップは、必要な全てのパネルオブジェクト、制御修飾子オブジェクト及びデータ記述子オブジェクトを検索することによって起こる。データメディアオブジェクトと関連する実際のデータは、データを表示すべきときにのみ検索される。様々な修飾子オブジェクトは、パネルを観察している間に評価される。
【０００９】
データベースはオーサリング過程を経て生成される。オーサリング過程では、トピックと様々な要素を組み合わせて、マルチメディアデータを提示するための流れを作成する。オーサリングが完了したならば、ソフトウェアプログラムのコンパイルに類似する方式で要素を処理して、データベースを作成する。
【００１０】
【発明の実施の形態】
本出願は、Ａｐｐｌｅ　Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．という同一の譲受人に譲渡されており、１９９３年１月２７日に出願された名称「Ｍｅｔｈｏｄ　ａｎｄ　Ａｐｐａｒａｔｕｓ　Ｆｏｒ　Ｐｒｏｖｉｄｉｎｇ　Ａ　Ｈｅｌｐ　Ｂａｓｅｄ　Ｗｉｎｄｏｗ　Ｓｙｓｔｅｍ　Ｕｓｉｎｇ　Ｍｕｌｔｉｐｌｅ　Ａｃｃｅｓｓ　Ｍｅｔｈｏｄｓ」の出願番号第０８／０１００６３号；１９９３年１月２７日に出願された名称「Ｍｅｔｈｏｄ　ａｎｄ　Ａｐｐａｒａｔｕｓ　Ｆｏｒ　Ｄｉｓｐｌａｙｉｎｇ　Ａｎｄ　Ｓｃｒｏｌｌｉｎｇ　Ｄａｔａ　Ｉｎ　Ａ　Ｗｉｎｄｏｗ－Ｂａｓｅｄ　Ｇｒａｐｈｉｃ　Ｕｓｅｒ　Ｉｎｔｅｒｆａｃｅ」の出願番号第０８／０１００６１号；１９９３年１月２７日に出願された名称

「Ｍｅｔｈｏｄ ａｎｄ Ａｐｐａｒａｔｕｓ Ｆｏｒ Ｐｒｅｓｅｎｔｉｎｇ Ｉｎｆｏｒｍａｔｉｏｎ Ｉｎ Ａ Ｄｉｓｐｌａｙ Ｕｓｉｎｇ Ｆｌｏａｔｉｎｇ Ｗｉｎｄｏｗｓ」の出願番号第０８／０１００６４号；及び１９９３年1月２７日に出願された名称「Ｍｅｔｈｏｄ ａｎｄ Ａｐｐａｒａｔｕｓ Ｆｏｒ Ｐｒｏｖｉｄｉｎｇ Ｖｉｓｕａｌ Ｃｕｅｓ Ｉｎ Ａ Ｇｒａｐｈｉｃａｌ Ｕｓｅｒ Ｉｎｔｅｒｆａｃｅ」の出願番号第０８／０１００６２号に関連している。

【００１１】

対話型マルチメディア教示デリバリエンジンを説明する。以下の説明中、本発明を完全に理解させるために、特定の詳細な事項を数多く挙げる。しかしながら、それらの特定の詳細な事項がなくとも本発明を実施しうることは当業者には明白であろう。また、別の場合には、本発明を無用にわかりにくくしないために、テキスト、グラフィック及びビデオデータを表示するための周知の回路を詳細には示さなかった。

【００１２】

図１を参照して、好ましい実施形態のコンピュータシステムを説明する。本発明は、Ａｐｐｌｅ（登録商標）Ｍａｃｉｎｔｏｓｈ（登録商標）ファミリのコンピュータに属するコンピュータの１つのような汎用マイクロコンピュータにおいて実現されるのが好ましい。いずれにしても、好ましい実施形態により利用しうるようなコンピュータシステムは、一般に、情報を通信するバス又は他の通信手段１０１と、前記バス１０１と結合し、情報を処理する処理手段１０２と、前記バス１０１と結合し、前記プロセッサ１０２に関わる情報及び命令を記憶するランダムアクセスメモリ（ＲＡＭ）又は他の記憶装置１０３（一般には主メモリという）と、前記バス１０１と結合し、前記プロセッサ１０２に関わる静的情報及び命令を記憶する読取り専用メモリ（ＲＯＭ）又は他の静的記憶装置１０４と、前記バス１０１と結合し、情報及び命令を記憶する磁気ディスク及びディスクドライブなどのデータ記憶装置１０５と、前記バス１０１に結合し、情報及び指令選択を前記プロセッサ１０２へ通信する英数字キー及びその他のキーを含む英数字入力装置１０６と、前記バス１０１に結合し、情報及び指令選択を前記プロセッサ１０２へ通信し且つカーソルの動きを制御するマウス、トラックボール、カーソル制御キーなどのカーソル制御装置１０７と、バス１０１に結合し、テキスト出力、グラフィック出力又はビデオ出力を表示する表示装置１０８とを具備する。加えて、システムが情報の永久コピーを作成するためのプリンタなどのハードコピー装置１０９を含んでいると有用である。ハードコピー装置１０９はバス１０１を介してプロセッサ１０２と結合している。図１のコンピュータシステムは、音声データを処理し且つ再生する手段（図示せず）を含んでいても良い。

【００１３】

図１に示すコンピュータシステムは、一般に良く知られているウィンドウ利用グラフィカル・ユーザインタフェースを形成するコンピュータシステム（たとえば、Ａｐｐｌｅ Ｃｏｍｐｕｔｅｒｓ，Ｉｎｃ．から入手可能なＡｐｐｌｅ Ｍａｃｉｎｔｏｓｈファミリのコンピュータ）であり、従って、ここでは詳細な説明は不要である。動作中、現時点で好ましい実施形態のウィンドウ利用ユーザインタフェースは、周知のように、ＲＡＭ１０３又はＲＯＭ１０４に記憶され且つプロセッサ１０２により実行されるソフトウェアを使用して生成、表示される。

【００１４】

以下の説明の中で明白になるであろうが、様々なコンピュータ素子は説明する機能性を提供する。たとえば、プレゼンテーションウィンドウは表示装置１０８に表示されるであろう。プロセッサ１０２はオブジェクト生成手段と、修飾子の評価と、デリバリエンジンを構成する他の処理手段とに関わる処理資源を提供するであろう。生成されたデータベースはデータ記憶装置１０５に記憶され、デリバリプロセスの間にＲＡＭ１０３にロードされるであろう。

【００１５】

現時点で好ましい実施形態では、コンピュータシステムの動作の補助を実行するヘルプ機能の一部として、マルチメディアデリバリエンジンを実現する。以下の説明から、本発明

の特徴や属性を、たとえば、対話型プレゼンテーション、教示、ゲーム開発又は参照ガイドを目的として多様な分野で適用できることは明白であろう。従って、本発明は以下の現時点で好ましい実施形態の説明によって限定されるのでなく、多様なメーカーにより提供されているコンピュータシステムや、教示用途と非教示用途の双方を有するシステムにも本発明を適用しうることは当業者により理解されるであろう。

【００１６】

デリバリエンジンの概要

デリバリエンジンは複数のメディアを使用して情報の提示を実行する。現時点で好ましい実施形態は、ユーザ教示を実行するヘルプシステムにある。教示は一連のトピックとして編成されている。各トピックは一般トピック情報と、パネルに対する参照とを含む。各パネルはテキスト、グラフィック、ビデオなどの視覚メディア並びに制御オブジェクト及び修飾子オブジェクトに対する参照を含む。トピック、パネル、視覚メディア及び修飾子は全て個別のエンティティとして１つのデータベースの中にカプセル化されている。すなわち、それらは全てデータベース中のオブジェクトである。図２は、デリバリエンジンの基本構成要素及び現時点で好ましい実施形態のプレゼンテーションウィンドウを示す。プレゼンテーションウィンドウは、パネルの視覚メディアデータを表示するために使用される様々な表示制御を有する事前定義済ウィンドウである。図２を参照すると、プレゼンテーションウィンドウ２０１はデータベース検索エンジン２０２に結合している。現時点で好ましい実施形態のプレゼンテーションウィンドウについては、以下にさらに詳細に説明する。データベース検索エンジン２０２は現在オペレータ入力及び先行オペレータ入力に応答して、データベース２０３から次に表示すべきデータを検索するように動作する。

【００１７】

データベース２０３はトピックオブジェクト２０４と、パネルオブジェクト２０５と、パネルオブジェクト２０６とから構成される複数のオブジェクトを含む。一般的に言えば、様々なオブジェクトは次のように定義される。

【００１８】

トピック　一連のパネル。

パネル　パネルオブジェクト、修飾子及びユーザが見てとるものを構成する表示フォーマットの集合体。

パネルオブジェクト　テキスト、グラフィック及びビデオデータ、制御、修飾子及びフォーマット。

制御　押しボタン、ラジオボタン及びチェックボックスなどの標準Ｍａｃｉｎｔｏｓｈ制御項目。ラジオボタンとチェックボックスは修飾子に関わる条件付き試験のためのユーザ入力を獲得するために使用される。

修飾子　ユーザに対しどのパネルを表示するかを確定するために使用される条件。

フォーマット　パネルオブジェクトをどのようにして表示するかのオプションの記述子。

ムービー　同期音声を伴う又は伴わないグラフィック動画像。

【００１９】

現時点で好ましい実施形態では、オブジェクトを他の２つ以上のオブジェクトにより使用しても良い。この再利用の概念は、一般にオブジェクト指向プログラミングと呼ばれているもののいくつかの面の１つである。たとえば、テキストデータの単一の項目を複数のパネルにより使用しても良い。先に挙げたオブジェクトはそれらのオブジェクトだけが網羅される又はそれらのオブジェクトに限定されることを意味するものではない。本発明の趣旨からの逸脱を生じさせずに他のオブジェクトを実現することは可能であろう。

【００２０】

現時点で好ましい実施形態においては、Ｃ＋＋プログラミング言語を使用する。Ｃ＋＋にはオブジェクト指向プログラミングを支援する固有の能力があるので、この言語を使用する。しかしながら、ＡＤＡ又はＭｏｄｕｌａ－２などのＣ＋＋に代わるプログラミング言語を使用しても、本発明の趣旨からの逸脱を生じさせないことは当業者には明白であろう。

【００２１】
図３は、データベース中のオブジェクトの論理的関係を示す。図３を参照すると、トピック３０１はパネル１$_{302}$、パネル２$_{302}$及びパネルｎ３０４といった一連のパネルを指示する。図３では、パネル１$_{302}$は修飾子３０５と、グラフィックデータ３０６と、制御３０７と、ビデオデータ３０８と、テキストデータ３０９とに連係している。内容に応じて、特定の１つのパネルは先行するパネルオブジェクトを全く有していなくとも良く、１つ又は複数有していても良い。再び図３を参照すると、パネル２$_{303}$もテキストデータ３０９に連係している。従って、テキストデータ３０９は、パネル１$_{302}$又はパネル２$_{303}$のいずれかが表示されるときに表示される。
【００２２】
プレゼンテーションウィンドウ
現時点で好ましい実施形態で使用できるプレゼンテーションウィンドウを図４に示す。図４を参照すると、プレゼンテーションウィンドウ４０１はトピックタイトル領域４０２と、パネル視覚データ領域４０３とを含む。トピックタイトル領域４０２は観察しているトピックの名前を含む。パネル視覚データ領域４０３は、観察中のパネルと関連する視覚データを含む。データデリバリエンジンが表示のためにデータを提供するのはパネル視覚データ領域４０３である。現時点で好ましい実施形態では、パネル視覚データ領域４０３は３４０画素の幅である。長さはパネル上のオブジェクトと、画面の大きさとによって確定される。長いテキストを含むパネルはごくわずかなテキストを伴うパネルより長くなる。パネルを構成するグラフィックオブジェクト、テキストオブジェクト、プロンプトオブジェクト、制御オブジェクト及びビデオオブジェクトの数は任意である。パネル上のオブジェクトの数を限定する唯一の要件は利用可能なＲＡＭである。
【００２３】
オブジェクトはパネルオブジェクトの中で指定される順序でパネルに描示される。そのため、シーケンスの後のほうで現れるオブジェクトはシーケンス中の先行オブジェクトの上に重なり合うこともありうる。テキストをパネル上にタグ、本体、フルパネル及びオーサー位置決めという４つの異なる方式で位置決めすることができる。タグテキスト、本体テキスト及びフルパネルテキストはデリバリエンジンにより位置決めされる。タグテキストは典型的には肉太字体活字で右寄せされる。本体テキストは典型的には正規字体活字をもって左寄せされる。フルパネルテキストは、パネルにテキストのみが現れる場合に使用される。追加フォーマットを作成し、データベースに記憶し、使用しても良い。テキストデータ、グラフィックデータ、ビデオ（ムービー）データ及び制御データの位置は、データベースに記憶されているオブジェクト定義の中のパラメータにより確定される。
【００２４】
プレゼンテーションウィンドウ４０１はプレゼンテーションウィンドウ制御領域４０４をさらに含む。制御領域４０４は教示経路を介してトラバースする手段を構成しており、前パネルボタン４０５と、次パネルボタン４０７とを含む。典型的には指示及びクリックの操作によってそれらのボタンを起動すると、トピックを構成している一連のパネルを通して逆方向トラバース（ボタン４０５）と順方向トラバース（ボタン４０７）が起こる。現在パネル番号インジケータ４０６により、ユーザはその時点でどのパネルを見ているかを知ることができる。制御領域４０４には、たとえば、制御ボタン４０８などの三（３）つまでの追加ボタンを追加して良い。制御はパネルの提示の流れを変更する（たとえば、トピックを切り換える）ためのものであっても良い。それらのオプションの制御はデータベースに記憶される。トピックは、それらの制御が現れるべきか否かを定義する。デリバリエンジンはそのオプションの制御を検索し、それらを制御領域４０４の中に挿入する。
【００２５】
最後に、プレゼンテーションウィンドウ４０１は、ウィンドウ利用グラフィカル・ユーザインタフェースを使用する者には良く知られているウィンドウ制御領域を含む。ウィンドウドラグバー４１０によって、プレゼンテーションウィンドウをカーソルドラグ動作を経て別の場所に位置決めすることができる。ウィンドウサイズ直しボックス４１１はウィン

ドウの大きさを希望に応じて伸縮させる。ウィンドウクローズボックス４０９はウィンドウを閉じ、それにより教示（ヘルプ）セッションを終了させる即時手段を構成する。

【００２６】

データベースの概要

本発明はデータに関わる初期ユーザ問い合わせに応答して動作する。初期問い合わせがなされたならば、本発明はプレゼンテーションウィンドウに表示するためのデータを検索するように動作する。データベース中の様々なオブジェクトの関係を図５ａ～図５ｃにさらに示す。

【００２７】

図５ａを参照すると、データベース５００はトピック５０１，５０８及び５１５と、グラフィックオブジェクト５０２と、修飾子オブジェクト５０３と、ボタン５０４と、パネル５０７及び５１３と、制御オブジェクト５０９及び５１２と、フォーマットオブジェクト５０５及び５１０と、テキストオブジェクト５０６及び５１４とを伴うものとして示されている。図５ｂは、トピック５０１とパネル５０７及び５１３並びにボタン５０４との連係を示す。最後に、図５ｃはパネル５０７とグラフィックオブジェクト５０２と、修飾子５０３と、制御オブジェクト５０９と、テキストオブジェクト５１４との連係を示す。それらの連係が実行される方式は以下の説明の中で明白になるであろう。

データベースの生成

【００２８】

先に説明した通り、データベースは複数のデータオブジェクトから構成されている。データベースの生成は本質的にはオーサリング、オブジェクト生成という２つのステップから成るプロセスである。これはソフトウェアプログラムの書込みとコンパイルに類似している。オーサリングはプログラム又はスクリプトを書込むことに類似している。この場合、オーサリングはトピック内容を生成するために使用される。次に、トピック内容をオブジェクト生成手段に与えて、データベースに入力すべきオブジェクトを作成する。オブジェクト生成手段は、コンパイルに類似するステップを実行する。データベース自体は、コンパイルの間に作成されるオブジェクトファイルに類似している。

【００２９】

現時点で好ましい実施形態では、様々なオーサリングツールを使用して良い。ある場合には、トピック内容を１組の教示として生成しても良い。トピック内容はＷｈａｔ　Ｙｏｕ　Ｓｅｅ　Ｉｓ　Ｗｈａｔ　Ｙｏｕ　Ｇｅｔ（ＷＹＳＩＷＩＧ）ツールを介して生成されても良い。この場合、パネルは表示されるであろうものとして生成される。事例ごとに異なるオブジェクト生成手段が使用されることは言うまでもないであろう。データベースを生成するために使用されるトピック内容を作成するための他の技法は当業者には明白であろうし、本発明の趣旨から逸脱しないであろう。

【００３０】

概念の上では、データベースは資源及びデータから構成されている。資源はデータオブジェクトである。データは表示される実際のテキスト、グラフィックデータ又はビデオデータである。データオブジェクトはオブジェクト型と、識別番号（「ｉｄ」）とにより定義される。現時点で好ましい実施形態においては、オブジェクト型は４文字ストリングであり、オブジェクトｉｄは整数である。表Ａは、本発明の現時点で好ましい実施形態におけるプレゼンテーションウィンドウに利用可能である１組の「ビルディングブロック」データオブジェクトを挙げている。このリストは全てを網羅しているのではなく、単に一般に使用されるオブジェクトを代表しているにすぎない。

【００３１】

【表１】

表Ａ
データオブジェクトのリスト

カテゴリ	型	ラベル	説　明
トピック	acPT	kResTopic	トピックスクリプト
パネル	ppUA	kResPanel	パネル
内　容	pcTX;	kResSTxt	テキスト
内　容	pcTF;	kResSTxtF	フルパネルテキスト
内　容	pcTB;	kResSTxtB	本体テキスト
内　容	pcTT;	kResSTxtT	タグテキスト
内　容	pcGA;	kResGraphic	グラフィック
内　容	pcGF;	kResGraphicF	フルパネルグラフィック
内　容	pcPR;	kResPrompt	プロンプト
内　容	pcCI;	kResCtrlItem	制御項目
内　容	pcMO;	kResMovie	ムービー
内　容	pcMC;	kResMovieC	コントローラを伴う ムービー
修飾子	pqCN	kShowIf	条件が真であれば パネルを示す
修飾子	pqCS	kSkipIf	条件が真であれば パネルをスキップする
修飾子	pqMS	kResMakeSure	このパネルを示す前に条件が真であることを確認する。 真でなければ、Oopsトピックを示す。
修飾子	pqCN	kResCondition	条件
コンテキスト	pxCA	kResContext	コンテキスト検査のための指定

【００３２】
現時点で好ましい実施形態では、各オブジェクトをＭａｃｉｎｔｏｓｈ資源として定義する。Ｍａｃｉｎｔｏｓｈ資源は、様々なアプリケーションにより使用しうるコード又はデータである。Ｍａｃｉｎｔｏｓｈ資源はＡｄｄｉｓｏｎ－Ｗｅｓｌｅｙ　Ｐｕｂｌｉｓｈｉｎｇ　Ｃｏｍｐａｎｙ，Ｉｎｃ．刊行の文献「Ｉｎｓｉｄ　Ｍａｃｉｎｔｏｓｈ　Ｖｏｌｕｍｅ　１」の中で説明されている。表Ａで説明したオブジェクトの多くについての実際の資源定義は付録Ａの中で規定されている。それらの資源定義は、資源をどのように定義できるかを示す例であり、全てを定義し尽くしているわけではない。
【００３３】
表Ａに戻ると、第１の欄のカテゴリはデータオブジェクトの一般カテゴリ、すなわち、トピック、パネル、内容、アクション、修飾子又はコンテキストを指示する。第２の欄の型はオブジェクト型を挙げている。第３の欄はオブジェクトラベルである。オブジェクトラベルはオブジェクトを参照するための代替的方法であり、オーサリングツールと、デリバリエンジン実現コード（Ｃ＋＋）とによって使用される。最後に、第４の欄は特定のオブジェクトの用途の簡単な説明である。
【００３４】
デリバリエンジンの動作
本発明の現時点で好ましい実施形態においては、必要なときに限って素子を使用する。教示（パネル表示）を開始する前に、デリバリエンジンは見るべきトピックをセットアップしなければならない。図６のフローチャートに関して、トピックをセットアップするステップを説明する。図６を参照すると、ステップ６０１ではデータベースからトピックａを検索し、オブジェクトとしてインスタンシエートする。オブジェクトとしてインスタンシエートするということは、そのクラスの１つのインスタンスを作成することを意味している。トピックオブジェクトは一般トピック情報と、そのトピックと関連するパネルのリス

トとを含む。次に、ステップ６０２では、トピックにリンクする各パネル（すなわち、そのトピックのパネルのリストの中にある）をデータベースから検索し、同様にオブジェクトとしてインスタンシエートする。検索されたパネルは、それぞれ、パネルオブジェクトのリストを含む。次に、ステップ６０３で、各パネルオブジェクトをデータベースから検索し、オブジェクトとしてインスタンシエートする。パネルオブジェクトの内容ではなく、パネルオブジェクト記述それ自体のみを検索し、インスタンシエートする。たとえば、テキストパネルオブジェクトはテキスト内容に対する参照と、パネル上のテキストの場所又はフォーマットとを含む。パネルが表示されるまでテキスト内容を検索する。

【００３５】

次に、ステップ６０４では、パネルにリンクする各修飾子を検索し、オブジェクトとしてインスタンシエートする。修飾子はコラボレータとしてパネルオブジェクトに付加される。コラボレータは、別のオブジェクトを支援、援助又は補助するように働く独立オブジェクトである。最後に、トピックリスト（パネルリストからは明確に区別される）の中のパネルに続くパネルオブジェクトがあれば、ステップ６０５で、それらのパネルオブジェクトを検索し、オブジェクトとしてインスタンシエートする。それらのオブジェクトは、先にインスタンシエートされたパネルに付加される。これにより、トピックの中でパネル記述子を変更することができる。たとえば、トピックからのパネルにプロンプトを付加することができる。この時点で、教示のトピックはデリバリ可能な状態になる。

【００３６】

教示の進行中のパネルのデリバリを図のフローチャートのステップに関して説明する。図７を参照すると、全てのパネルとパネルオブジェクトを検索し且つインスタンシエートしたならば、ステップ７０１で、表示できることを確定する修飾子をもつ第１のパネル又は修飾子をもたない第１のパネルと共にデリバリシーケンスが始まる。修飾子を検査する方式については、以下に図８及び図９を参照して説明する。ここでは、第１のパネルは識別されているものと仮定する。次に、ステップ７０２ではパネルオブジェクト内容を検索する。先に述べた通り、当初はパネルオブジェクト記述子のみを検索した。パネルオブジェクト内容はパネルに表示すべきテキストデータ、グラフィックデータ又はビデオデータである。パネル内容を検索したならば、ステップ７０３で、それらの内容をプレゼンテーションウィンドウのパネル表示領域に表示する。この表示ステップの中で、オーサー又はデリバリエンジンにより、オーサーがデータのオブジェクト型を選択することによって、オブジェクトの表示をフォーマットしても良い。

【００３７】

この時点で、パネルを表示し、オーサーはユーザに様々なオプションを提供しても良い。オーサーがユーザに提供しうる１つのオプションはゴーバックオプションである。ゴーバックオプションでは、ユーザは先に表示されていたパネルに戻るようにジャンプしても良い。ゴーバックオプションについては、以下に図１０に関してさらに詳細に説明する。その他のオプションは、次に説明するパネルの現在リストを巡る逆方向又は順方向のトラバースである。それらのオプションはプレゼンテーションウィンドウの機能性（図４を参照）の一部として提供される。図７に戻ると、ステップ７０４，７０５では、表示すべき前進（次）パネルがそれ以上存在しているか否かの確定をデリバリエンジンにより実行する。現在パネルを検査している間、この確定はユーザに気付かせない。順方向に表示すべきパネルがそれ以上存在していないのであれば、ステップ７０６で「次」オプションをディスエーブルする。この場合、ステップ７０７では、ユーザオプションは前パネルを示すこと又は教示を終了させることに限定される。ユーザが教示を終了させることを選択すれば、教示を終了する。そうでなければ、ステップ７１０で前パネルを表示する。

【００３８】

ステップ７０５の表示すべきパネルがまだあるかという質問に肯定応答すると、「次」オプション及び「前」オプションがイネーブルされる。ユーザは、ステップ７０８で、オプションの中の１つを選択することにより応答しなければならない。「次」オプションを選択したならば、ステップ７０９で次の望ましいパネルを表示する。次の望ましいパネルは

、次に続くパネルと関連する修飾子の状態により（又はパネルに関わる修飾子の欠如により）識別される。次に、デリバリエンジンは表示すべきパネルが他に残っているか否かを判定することによってプロセスを繰り返す（ステップ７０４）。先行オプションを選択した場合には、ステップ７１０で前パネルを表示し、デリバリエンジンは表示すべき他のパネルが残っているか否かを判定することによりプロセスを繰り返す（ステップ７０４）。

【００３９】

以上の説明から、デリバリエンジンは表示されていたパネルの表示リストを維持していることが明らかである。この表示リストはゴーバックオプション及び（以下に説明するような）ｍａｋｅ－ｓｕｒｅ修飾子を使用するパネルのトラバースを容易にする。

【００４０】

修飾子の評価

パネルと関連させうる修飾子は、トピックのパネルを表示するときの対話形式を容易にする。修飾子とそこで参照される条件はそのパネル表示の流れの制御を行う。修飾子は、各パネルを示す前に、ユーザがデリバリエンジンで対話していない間にバックグラウンドでデリバリエンジンにより評価される。修飾子は必ずしもその時点で示されているパネルでなくとも良い任意のパネルの中のユーザ制御に基づくか、又はユーザのコンテキストの試験に基づくことができる。たとえば、ある機能を実行するためのショートカット技法又はある機能を実行するための遠回り技法を表示するという教示があれば、その機能を実行するためにどの方法を学習することを望むかを決定するオプションをパネルＸにおいてユーザに与えても良い。その教示に関わるパネルはパネルＸからパネルＸ＋ｍとして順次連係されると仮定する。さらに、パネルＸ＋１からＸ＋ｎはショートカット技法を示し、パネルＸ＋ｎ＋１からＸ＋ｍは遠回り技法を示すものと仮定する。ユーザがパネルＸで遠回り技法を学習することにより進むことを選択した場合、パネルＸ＋１からＸ＋ｎ（ショートカット技法に対応する）はそれらのパネルをスキップさせる修飾子を有するであろう。

【００４１】

ｓｈｏｗ－ｉｆ修飾子及びｓｋｉｐ－ｉｆ修飾子の評価を図８を参照して説明する。図８を参照すると、パネルがＳｈｏｗ－Ｉｆ修飾子又はＳｋｉｐ－Ｉｆ修飾子を有している場合、ステップ８０１で、その修飾子に関わる条件をデータベースから検索し、それをオブジェクトとしてインスタンシエートする。次に、ステップ８０２，８０３で条件を評価する。条件はユーザ制御、コンテキスト検査又は別の条件から構成されているであろう。それらの要素は対を成し且つ／又は論理的であり（オーサーの選択）、要素を無限に入れ子にすることができる。従って、ｓｈｏｗ－Ｉｆ修飾子又はＳｋｉｐ－Ｉｆ修飾子は２進決定ツリーである。そのツリーを評価する。最終結果が偽であれば、ステップ８０４でパネルをスキップする（Ｓｈｏｗ－Ｉｆ）か、又は示す（Ｓｋｉｐ－Ｉｆ）。最終結果が真である場合には、ステップ８０５でパネルを示す（Ｓｈｏｗ－Ｉｆ）か、又はスキップする（Ｓｋｉｐ－Ｉｆ）。

【００４２】

第３の修飾子はＭａｋｅ－Ｓｕｒｅ修飾子である。Ｍａｋｅ－Ｓｕｒｅ修飾子は（たとえば、補修的な教示又はユーザが先のステップに戻らなければならないことの通知のための）トピックの切替えをゴーバック動作において検査される基準として実行する。第１に挙げた例のＭａｋｅ－Ｓｕｒｅ修飾子を図９を参照して説明する。まず、ステップ９０１では、修飾子に関わる条件をデータベースから検索し、それをオブジェクトとしてインスタンシエートする。次に、ステップ９０２及び９０３で、検索したパネル条件を評価する。条件はＳｈｏｗ－Ｉｆ又はＳｋｉｐ－Ｉｆと同じ方法で評価されるが、結果の利用方式は異なる。結果が偽であれば、ステップ９０４で別のトピックを自動的に示す。どのトピックを示すべきかの識別はＭａｋｅ－Ｓｕｒｅ記述子の中に含まれている（記述子の説明については付録Ａを参照）。偽の結果を最初にもたらしたＭａｋｅ－Ｓｕｒｅ修飾子がそのトピックを表示させる。結果が真であれば、ステップ９０５で追加のＭａｋｅ－Ｓｕｒｅ修飾子を求めて検査を実行する。パネルに関してＭａｋｅ－Ｓｕｒｅ修飾子がまだ存在しているならば、ステップ９０１によってそれらの修飾子を評価する。それ以上のＭａｋｅ

－Ｓｕｒｅ修飾子が存在せず且つパネルと関連する全てのＭａｋｅ－Ｓｕｒｅ修飾子が真であると評価された場合には、ステップ９０６でパネルを示して良い。

【００４３】

ゴーバックオプションと共に使用される場合のＭａｋｅ－Ｓｕｒｅ修飾子を図１０を参照して説明する。図１０を参照すると、ステップ１００１で、ユーザはオーサーにより「ゴーバック」オプションを提示され、ユーザはそれを取り上げることを選択する。この場合、ステップ１００２では、修飾子をもたないか又はパネルを表示させる条件（Ｓｋｉｐ－Ｉｆ，Ｓｈｏｗ－Ｉｆ，Ｍａｋｅ－Ｓｕｒｅ）を伴う修飾子を有するパネルが見出されるまで、先に示されていたパネルを逆の順序で探索する。次に、ステップ１００３でパネルを表示する。Ｍａｋｅ－Ｓｕｒｅ修飾子はオーサーに、変更があった場合には自動的に警告を発生する１組の条件（Ｍａｋｅ－Ｓｕｒｅ）を指定させる。次に、エンジンはシーケンスの中で条件が受け入れ可能であったところを見出すためにサーチバックすることができ、そこで、ユーザはそれらの条件を正確に再実行できる。

【００４４】

コンテキスト検査は条件付き動作のオプションの１要素である。１つの条件が次に挙げるものの１つ又は２つ以上の組み合わせを指定できることに注意する。

・ラジオボタン又はチェックボックスなどのパネル制御項目
・コンテキスト検査指定
・その他の条件

【００４５】

コンテキスト検査指定は、ユーザのコンテキストに照らして評価したときに真又は偽の結果をもたらす資源（オブジェクト）である。たとえば、オープン「ウィンドウ」に関わる検査はコンテキスト検査指定である。現時点で好ましい実施形態において実行できる他のコンテキスト検査は次の通りである。

・ウィンドウ＜タイトル＞は開いており且つ前面にある、
・ウィンドウ＜タイトル＞は開いているが見えない、
・アプリケーション＜名＞は開いている＜ラン中＞、
・アプリケーション＜名＞は開いており且つ前面にある。

【００４６】

トピックの切替え

先に述べた通り、パネルのＭａｋｅ－Ｓｕｒｅオプションを経てトピックを切替えても良い。さらに、現時点で好ましい実施形態では、複数のトピックを同時に示すことができる。オーサーは旧トピックを表示したままで新たなトピックを提示し、旧トピックを閉じ且つそれを新たなトピックを置換えるか、もしくは旧トピックを隠し且つ新たなトピック（「Ｏｏｐｓ」トピックと呼ばれる）を示すことを選択しても良い。「Ｏｏｐｓ」トピックが閉じられたとき、隠れトピックは再び自動的に示される。これはデリバリエンジンにより維持される表示パネルによって可能になる。

【００４７】

言うまでもなく、トピックが切替わるたびに、図６を参照して説明したトピックのセットアップを実行しなければならない。

【００４８】

実施例

本発明の一実施例をここで示す。図１１ａ～図１１ｃは、オーサリングすべきトピックに関わるパネルである。この実施例では、コンピュータシステムにおいていくつかの動作を実行するための教示を提示している。この場合、教示をその代わりにテキスト形教示を経て提供するか又は実行中の教示を示すビデオを経て提供しても良い。そのため、ウィンドウを閉じるための教示であったならば、ユーザはクローズボックスを指示するためのカーソルをいかに操作するかを読み取り、次にカーソル制御装置の１つのボタンをクリックするか、又は単に実行中のアクションを監視するというオプションを得る。

【００４９】

図１１ａを参照すると、パネル１１０１はテキストオブジェクト１１２０及び制御オブジェクト１１３０という２つのオブジェクトを有する。テキストオブジェクト１１２０は教示のトピックを記述するテキストを含む。制御オブジェクト１１３０は、ユーザがビデオ教示を希望することを指定しうるチェックボックスである。図１１ｂは、所望の動作をいかに実行すべきを記述する１つのオブジェクト、すなわち、フルパネルテキストオブジェクト１１２１を有するパネル１１０２を示す。図１１ｃは、実行中の所望の動作を示すムービーである１つのオブジェクト、すなわち、ビデオオブジェクトを有するパネル１１０３を示す。

【００５０】

トピックのセットアップは既に起こっていると仮定すると、パネル１１０１は表示される。教示をテキスト又はビデオにより出現させるというオプションをユーザに提示する。ビデオ教示はＶＩＤＥＯ　ＩＮＳＴＲＵＣＴＩＯＮチェックボックスを検査することにより指定される。プレゼンテーションウィンドウのパネルトラバース演算子を経て次パネルに向かうとき、チェックボックスが検査されない場合はパネル１１０２を表示し、チェックボックスが検査される場合にはパネル１１０３を表示する。

【００５１】

図１２は、先の表Ａの中で記載したオブジェクトを使用して先の実施例をいかにして符号化しうるかの擬似コードの例を含む。図１２を参照すると、第１行はトピックオブジェクト定義を含む。第２行～第４行はパネルオブジェクト内容の定義、すなわち、テキストデータ及びビデオデータを含む。第５行はパネル１１０１の定義を開始する。パネル１１０１と関連するオブジェクトであるテキストオブジェクトとチェックボックスオブジェクトは、それぞれ、（明瞭を期するために）第６行と、第７行にある。第８行はチェックボックス制御オブジェクトの定義を含む。チェックボックスオブジェクトはパネル１１０１により参照されるのであるが、それはオブジェクトの外で定義されることに注意する。これにより、チェックボックスオブジェクトを複数のパネルによって利用できる。さらに、パネルオブジェクト内容もオブジェクトが使用されるパネル定義の中で定義しうることに注意する。

【００５２】

第９行はパネル１１０２の定義を開始する。パネル１１０２は第１０行と第１１行に、テキストオブジェクトと、Ｓｋｉｐ－Ｉｆ修飾子オブジェクトとをそれぞれ含む。第１２行では、パネル１１０２により連係する修飾子オブジェクトを定義する。Ｓｋｉｐ－Ｉｆ修飾子オブジェクトは条件としてチェックボックスオブジェクトに連係する（第１３行）ことに注意する。

【００５３】

第１４行はパネル１１０３の定義を開始する。パネル１１０３は第１５行と第１６行にビデオオブジェクトと、Ｓｈｏｗ－Ｉｆ修飾子オブジェクトとをそれぞれ含む。最後に、第１７行～第１８行ではＳｈｏｗ－Ｉｆ修飾子オブジェクトを定義している。Ｓｋｉｐ－Ｉｆ修飾子の場合と同様に、Ｓｈｏｗ－Ｉｆ修飾子は条件としてチェックボックスオブジェクトに連係する（第１８行）。

【００５４】

図１３は、この実施例におけるオブジェクトの連係を示す。図１３を参照すると、トピック１１００はパネル１１０１，１１０２及び１１０３に連係しているであろう。パネル１１０１はテキスト１１２０及び制御１１３０へのリンクを有しているであろう。明白であるように、制御１１３０には複数のオブジェクトがリンクする。パネル１１０２はテキストオブジェクト１１２１及びＳｋｉｐ－Ｉｆ修飾子１１６０にリンクする。修飾子１１６０自体は制御１１３０にリンクする。最後に、パネル１１０３はビデオオブジェクト１１４０と、Ｓｈｏｗ－Ｉｆ修飾子１１７０とにリンクする。Ｓｈｏｗ－Ｉｆ修飾子自体は制御１１３０にリンクする。

【００５５】

トピックのセットアップ中、オブジェクトは先に図６を参照して説明したようにロードさ

れる。そこで、パネルのデリバリは図７～図１０に関連して説明した方式で起こるであろう。

【００５６】

以上、コンピュータシステムで表示するためのマルチメディアデータを対話形式でデリバリする方法及び装置を説明した。

APPENDIX A

```
// Copyright © 1993 Apple Computer, Inc. (17 U.S.C. 401)
  All rights reserved.
// Date:      05-May-93
// Delivery Engine Data Types
     // Coach Mark resouce
 type kResCoachMark {
 longint none=kTargetNone,     // target application
 signature
  self=kTargetSelf,
  front=kTargetFront,
  topic=kTargetTopic;
 longint balloon='ndex', location='mLoc',
  menu='mMrk',
  object='draw';
  // event id
integer = $$Count of(keys);
array keys {
 switch {
 case Number:
  key longint='iNum';
  pstring;      // balloon or dialog or menu item number
case Origin:
 key longint='oCon';      // origin reference for rectangle
 pstring topLeft="0", bottomLeft="1",
  topRight="2", bottomRight="3";
 case Rect:
  key longint='iRct';     // marking rectangle
  pstring;     // "rct=" top,left,bottom,right
 case Macro:
 key longint='lMac';       // location macro
pstring GrowBox="GrowBox", ZoomBox="ZoomBox",
  CloseBox="CloseBox", TitleBar="TitleBar",
  BootDisk="BootDisk";
case WindowName:
```

```
   key longint='wNam';
   pstring; // window name
  case FrontWindow:
   key longint='wNam';
   pstring=" _ FRONTWINDOW _ "; // window name
  case DesktopWindow:
   key longint='wNam';
   pstring=" _ DESKWINDOW _ "; // window name
  case BalloonNumber:
   key longint='iNum';
   pstring; // balloon item number
  case DialogItemNumber:
   key longint='iNum';
   pstring; // dialog item number
  case MenuTitleName:
   key longint='mNam';
   pstring; // menu title name
  case MenuTitleNumber:
   key longint='mNum';
   pstring; // menu title number
  case MenuItemName:
   key longint='iNam';
   pstring;       // menu item name
  case MenuItemNumber:
   key longint='iNum';
   pstring;       // menu item number
  case ObjectName:
   key longint='iNam';
   pstring;       // menu item name
  case MarkStyle:
   key longint='sNam';
   pstring red_circle="defaultl",
       yellow x="default2",
       red_underline="default3",
       red_arrow="default4";     // mark style name
case MenuItemColor:
```

```
  key longint='mCol';
  pstring myblack="1", yellow="2",   // myblack instead of
  black
          // because black is defined in ViewTypes.r
   magenta="3", red="4", cyan="5",
   green="6", blue="7", white="8";   // menu item color
   number
 case MenuItemStyle:
  key longint='mFac';
  pstring cc_bold="1", cc_italic="2",
      cc_underline="4", cc_outline="8",
      cc_shadow="16", cc_condense="32",
      cc_extend="64";           // font style number
 case StyleOption:
  key longint='sOpt';
  pstring; // "Lng=0,0,S,F" where S is start
      //   and F is finish
      // 1=NW, 2=N., 3=NE, 4=E, ... 8=W
      }; // switch
    ); // array
    align word;
);
   // Topic
type kResTopic {
 longint none=0;          // target application signature
 integer none=kNone, noButton=kNone,
   defaultButton=kDefault,
   huhButton=kHuhBtnId,
   goStartButton=kGoStartBtnId;      // optional button #1
   id
integer none=kNone, noButton=kNone,
   defaultButton=kDefault,
   huhButton=kHuhBtnId,
   goStartButton=kGoStartBtnId;      // optional button #2
   id
integer none=kNone, noButton=kNone,
```

```
    defaultButton=kDefault,
    huhButton=kHuhBtnId,
    goStartButton=kGoStartBtnId;      // optional button #3
    id
 integer none=kNone, noMinimum=kNone,
    defaultMinimum=kNone;  // minimum panel height
 integer none=kNoMaxHeight, noMaximum=kNoMaxHeight,
    defaultMaximum=kNoMaxHeight;      // maximum panel height
 integer;        // topic title TEXT ID number
 integer;        // reserved field, always enter 0.
 integer;        // reserved field, always enter 0.
 integer;        // reserved field, always enter 0
 integer = $$Countof(entries);
 array entries {
    longint; // entry type
    longint; // resource id or other
  } ;
} ;
      // Panel


type kResPanelUnivA {
  integer = $$CountOf(items);
  array items {
    longint; // scope
    integer; // resId (DBRef) for scope
  }; // array
}; // type

   // Text format- for full, tag, body, and custom formats.

type kResFormat {
  integer teFlushDefault=0,
    teCenter=1,
    teFlushRight=-1,
    teFlushLeft=-2;    // justification
```

```
  integer normal=0;   // face
  integer;            // size
  integer;            // top
  integer;            // left
  integer;            // bottom
  integer;            // right
  pstring;            // font name
  align word;         //   Espy Serif,     Espy Sans,
      Espy Serif Bold....
};

      // Text item

type kResSTxt   {
  integer; // resId for TEXT/styl
  integer;       // top
  integer;       // left
  integer;       // bottom
  integer;       // right
};

      // Graphic item

type kResGraphic {
  integer;            // resId for PICT
  integer;            // top
  integer;       // left
};
      // Movie item

type kResMovie {
  integer;       // top
  integer;       // left
  pstring;       // movie file name
  align word;
};
```

```
      // Ctrl item
      // Standard Macintosh controls

type kResCtrlItem {
 switch {
  case PushButton:
   key int = pushButProcReno;
   integer active=0, inactive=255;
   integer unchecked=0, checked=1;
   integer; // top
   integer; // left
   integer; // bottom
   integer; // right
   integer none=0; // action, kResEvent Id number
   pstring; // title
case RadioButton:
   key int = radioButProcReno;
   integer active=0, inactive=255;
   integer unchecked=0, checked=1;
   integer; // top
   integer; // left
   integer; // bottom
   integer; // right
   integer none=0; // action, kResEvent Id number
   pstring; // title
case CheckBox:
   key int = checkBoxProcReno;
   integer active=0, inactive=255;
   integer unchecked=0, checked=1;
   integer; // top
   integer; // left
   integer; // bottom
   integer; // right
   integer none=0; // action, kResEvent Id number
   pstring; // title
```

```
case PushButtonSysFont:
 key int = pushButProcRenoSysFont;
 integer active=0, inactive=255;
 integer unchecked=0, checked=1;
 integer; // top
 integer; // left
 integer; // bottom
 integer; // right
 integer none=0; // action, kResEvent Id number
 pstring; // title
case RadioButtonSysFont:
 key int = radioButProcRenoSysFont;
 integer active=0, inactive=255;
 integer unchecked=0, checked=1;
 integer; // top
 integer; // left
 integer; // bottom
 integer; // right
 integer none=0; // action, kResEvent Id number
 pstring; // title
case CheckBoxSysFont:
  key int = checkBoxProcRenoSysFont;
  integer active=0, inactive=255;
  integer unchecked=0, checked=1;
  integer; // top
  integer; // left
  integer; // bottom
  integer; // right
  integer none=0; // action, kResEvent Id number
  pstring; // title
}; // switch
  align word;
}; // type

  // Apple Event
```

```
type kResEvent {
 longint none=kTargetNone,     // target application
 signature
  self=kTargetSelf,
  front=kTargetFront,
  topic=kTargetTopic,
  reno=kTargetReno;
 longint;       // class
 longint none=0;     // event id
 integer none=0;     // optional short integer with key
 'IOPT'
 longint none=0;     // optional key
 pstring none="";    // data for optional key
 align word;
};
  // Condition
  // Referenced by Show-If, Skip-If, and Make-Sure.

type kResCondition {
 integer none=0, not=kNot;     // optional not or kNot
 integer update=kUpdate, dontUpdate=kDontUpdate;
 longint;       // type (EG kResCtrlItem, kResContext,
 kResCondition)
 integer;       // resId
 integer none=kNone, or=kOr, and=kAnd; // logic
 integer none=0, not=kNot;     // optional not or kNot
 integer update=kUpdate, dontUpdate=kDontUpdate;
 longint none=0; // type
 integer none=0; // resId
};
 // Make-Sure
type kResMakeSure {
 integer; // resource Id of kResCondition resource.
 integer; // resource Id of kResTopic resource for Oops.
};
  // Prompt
```

```
type kResPrompt {
   integer;     // Id of first (beginning) TEXT/styl
   resource
   integer;     // Id of second (middle) TEXT/styl resource
   integer;     // Id of third (end) TEXT/styl resource
   integer;     // Id of fourth (choice) TEXT/styl resource
};

 // Hot panel area

type kResHot {
 integer;       // top
 integer;       // left
 integer;       // bottom
 integer;       // right
 integer none=0; // action, kResEventId number
 };

     // Custom button

type kResRenoBtn {
 integer=1;     // (always momentary)
 integer; // off-up PICT id
 integer; // on-down PICT id
 integer=0;     // (dimmed not available)
 longint none=0;     // action, kResEvent id number
 integer=1;     // (not used, kResEvent is always immediate)
 integer; // top of top-left corner location
 integer; // left of top-left corner location
 integer=0;     // (always fixed)
 integer none=0;          // string index for Help Balloon
 text.
};

 // PreWin button
```

```
  // Placed in the control area along the bottom
  // of the presentation window.

 type kResPreWinBtn {
  integer=1;     // (always momentary)
  integer;            // off-up PICT id
  integer;            // on-down PICT id
  integer;            // dimmed
  longint none=0,     // action, kResEvent id number
  integer=1;     // (not used, kResEvent is always immediate)
  integer=0;     // top of top-left corner location is preset
  integer=0;     // left of top-left corner location is
  preset
  integer align=1,
   movable=1,
   fixed=0;      align (1) or fixed (0)
  integer none=0;     // string index for Help Balloon text.
);

   // PerWin Button.
   // Placed in the control area along the bottom
   // of the presentation window.

type kResPreWinBtn {
 integer=1;     // (always momentary)
 integer; // off-up PICT id
 integer; // on down PICT id
 integer; // dimmed
 longint none=0;     //action, KResEvent id number
 integer=1;     //not used, kResEvent is always immediate)
 integer=0;     // top of top-left corner location is preset
 integer=0;     // left of top-left corner location is
 preset
 integer align=1,
     moveable=1,
     fixed=0;   // align (1) or fixed (0)
```

```
  integer non=0;            // string index for Help Ballon
  text.
 };

 type kResContext {
  longint none=kTargetNone,     // target Process signature
      self=kTargetSelf,
      front=kTargetFront,
      topic=kTargetTopic,
      reno=kTargetReno;
 switch {
  case Chooser:
  key longint = kCntxChooser;
 switch {
  case isPrinterDirect:
   key longint = 0;
   fill byte[16];     // filler for 3 longs & 2 StringSpecs

 case isPrintersOnZone:
  key longint = 1;
  longint equals=2,
   notEqualTo=3,
   greaterThan=4,
   lessThan=5,
   greaterThanOrEqualTo=6,
   lessThanOrEqualTo=7; // Number compare selector
longint;   // Compare value
fill byte[8];   // filler for 1 long & 2 StringSpecs

case isPrinterSerial:
 key longint = 2;
 longint ModemPort=0,
   PrinterPort=1,
   EitherPort=2;
fill byte[12];       // filler for 2 longs & 2 StringSpecs
```

```
case isPrinterSCSI:
 key longint = 3;
 fill byte[16];      // filler for 3 longs & 2 StringSpecs

case isPrinterType:
 key longint = 4;
 fill byte[4];       // not used
 longint containsAnything=8,
 empty=9,
 matches=10,
 contains=11,
 startsWith=12,
 endsWith=13; // StringCompareMethod
pstring;   // Printer name (variable-length)
align word;
fill byte[6];   // filler for 1 long & 1 StringSpec

case isAppleTalk:
key longint = 5;
longint off=0,
 on=1;
fill byte[12];       // filler for 2 longs & 2 StringSpecs

case isNumberOfZones:
 key longint = 6;
 longint equals=2,
  notEqualTo=3,
  greaterThan=4,
  lessThan=5, greaterThanOrEqualTo=6,
  lessThanOrEqualTo=7; // Number compare selector
longint;        // Compare value
fill byte[8];   // filler for 1 long & 2 StringSpecs

case isPortSelected:
key longint = 7;
longint ModemPort=0,
```

```
    PrinterPort=1,
    EitherPort=2;
 fill byte[12];        // filler for 2 longs & 2 StringSpecs

 case isPortInUse:
  key longint = 8;
  longint ModemPort=0,
    PrinterPort=1,
    EitherPort=2;
  fill byte[12];       // filler for 2 longs & 2 StringSpecs
 }; // Chooser switch

 case Dialog:
  key longint = kCntxDialog;
  switch {
   case isFront:
    key longint = 0;
    longint;       // Resource ID
    fill byte[12]; // filler for 2 longs & 2 StringSpecs

 case isOpen:
  key longint = 1;
  longint;       // Resource ID
  fill byte[12];        // filler for 2 longs & 2 StringSpecs

 case isInvisible:
  key longint = 2;
  longint;       // Resource ID
  fill byte[12];      // filler for 2 longs & 2 StringSpecs

 case isFindDialog:
 key longint = 3;
 fill byte[16];        // filler for 3 longs & 2 StringSpecs

 case isFindMoreDialog:
 key longint = 4;
```

```
fill byte[16];       // filler for 3 longs & 2 StringSpecs

case isAnyDialogActive:
 key longint = 5;
 fill byte[16];      // filler for 3 longs & 2 StringSpecs
}; // Dialog switch

case File:
 key longint = kCntxFile;
 switch {
  case isFrontStartup:
   key longint = 0;
   fill byte[16];   // filler for 3 longs & 2 StringSpecs

 case isOpenStartup:
  key longint = 1;
  fill byte[16];    // filler for 3 longs & 2 StringSpecs
 case isCDStartup:
  key longint = 2;
fill byte[16]; // filler for 3 longs & 2 StringSpecs

case isPrintMonitor:
 key longint = 3;
 fill byte[16]; // filler for 3 longs & 2 StringSpecs

case isAppleShare:
 key longint = 4;
 fill byte[16];      // filler for 3 longs & 2 StringSpecs

case isEtherTalk:
 key longint = 5;
 fill byte[16];      // filler for 3 longs & 2 StringSpecs

case isTokenTalk:
 key longint = 6;
 fill byte[16];      // filler for 3 longs & 2 StringSpecs
```

```
case isControlPanel:
 key longint = 7;
 fill byte[8]; // filler for 2 longs
 pstring; // Nameof Control Panel
 align word;
 fill byte [6];      // filler for 1 long & 1 StringSpec

case isFileExists:
 key longint = 9;
 longint SystemFolder='macs',
  DesktopFolder='desk',
  TrashFolder='trsh',
  EmptyTrashFolder='empt',
  PrintMonitorFolder='prnt',
  StartupFolder='strt',
  AppleMenuFolder='amnu',
  ControlPanelFolder='ctrl',
  ExtensionsFolder='extn',
  PreferencesFolder='pref',
  TemporaryFolder='temp';      // System folder selector
 fill byte[4];       // filler for 1 long
 pstring;       // Name of desired file
 align word;
 fill byte[6];       // filler for 1 long & 1 StringSpec
 }; // File switch

case Finder:
 key longint = kCntxFinder;
 fill long;     // Finder selector
 fill byte[16];       // filler for 3 longs & 2 StringSpecs

case Process:
 key longint =  kCntxProcess;
 switch {
  case isFront:
```

```
    key longint = 0;
    longint;    // Process signature
    fill byte[12];    // filler for 2 longs & 2 StringSpecs

  case isOpen:
    key longint = 1;
    longint;          // Process signature
    fill byte[12];    // filler for 2 longs & 2 StringSpecs
); // Process switch
case System:
  key longint = kCntxSystem;
  switch (
   case gestaltCheck:
    key longint = 0;
    longint;     // Gestalt Selector
    longint      equals=2,
     notEqualTo=3,
     greaterThan=4,
     lessThan=5,
     greaterThanOrEqualTo=6,
     lessThanOrEqualTo=7; // Number compare selector
fill byte[2];   // not used
longint;   // Gestalt compare value
fill byte[2];   // filler for 1 StringSpec

case menuItemExists:
  key longint = 1;
  fill byte[4];       // not used
  longint containsAnything=8,
    empty=9,
    matches=10,
    contains=11,
    startsWith=12,
    endsWith=13;       // StringCompareMethod
  pstring;       // Menu name (variable-length)
  align word;
```

```
  longint containsAnything=8,
    empty=9,
    matches=10,
    contains=11,
    startsWith=12,
    endsWith=13;       // StringCompareMethod
  pstring;       // Menu Item name (variable-length)
  align word;

  case menuItemMarked:
  key longint = 2;
  fill byte[4];       // not used
  longint  containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
endsWith=13;    // StringCompareMethod
pstring;  // Menu name (variable-length)
align word;
  longint containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
  endsWith=13;  //    StringCompareMethod
  pstring;       // Menu Item name (variable-length)
  align word;

case menuItemEnabled:
  key longint = 3;
  fill byte[4];       // not used
  longint containsAnything=8,
    empty=9,
    matches=10,
    contains=11,
```

```
    startsWith=12,
    endsWith=13;       // StringCompareMethod
 pstring;   // Menu name (variable-length)
 align word;
 longint    containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
  endsWith=13;   // StringCompareMethod
pstring;   // Menu Item name (variable-length)
align word;

case menuItemDisabled:
 key longint = 4;
 fill byte[4];       // not used
 longint  containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
  endsWith=13;       // StringCompareMethod
 pstring;       // Menu name (variable-length)
 align word;
 longint containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
  endsWith=13;       // StringCompareMethod
 pstring;       // Menu Item name (variable-length)
 align word;

case isFileSharing:
 key longint = 6;
 longint off=0,
```

```
   on=1;
fill byte[12];       // filler for 2 longs & 2 StringSpecs

case isVideoHWBitDepth:
 key longint = 7;
 longint equals=2,
  notEqualTo=3,
  greaterThan=4,
  lessThan=5,
  greaterThanOrEqualTo=6,
  lessThanOrEqualTo=7;   // Number compare selector
 longint; // Compare value
 fill byte[8];       // filler for 1 long & 2 StringSpecs

case isMonitorBitDepth:
 key longint = 8;
 longint equals=2,
  notEqualTo=3,
  greaterThan=4,
  lessThan=5, greaterThanOrEqualTo=6,
  lessThanOrEqualTo=7;   // Number compare selector
 longint;       // Compare value
 fill byte[8];       // filler for 1 long & 2 StringSpecs

case isComputerNamed:
 key longint = 9;
 fill byte[4];       // not used
 longint  containsAnything=8,
  empty=9,
  matches=10,
  contains=11,
  startsWith=12,
  endsWith=13; // StringCompareMethod
 pstring; // Printer name (variable-length)
 align word;
 fill byte[6];       // filler for 1 long & 1 StringSpec
```

```
 case isGuestAccess:
  key longint = 10;
  longint off=0,
   on=1;
  fill byte[12];      // filler for 2 longs & 2 StringSpecs

 case isNumberMonitors:
  key longint = 11;
  longint equals=2,
   notEqualTo=3,
   greaterThan=4,
   lessThan=5,
   greaterThanOrEqualTo=6,
   lessThanOrEqualTo=7;   // Number compare selector
  longint;       // Compare value
  fill byte[8];       // filler for 1 long & 2 StringSpecs
 }; // System switch

 case Window:
  key longint = kCntxWindow;
  switch {
   case isFront:
   key longint = 0;
   fill byte[4];      // not used
   longint containsAnything=8,
    empty=9,
    matches=10,
    contains=11,
    startsWith=12, endsWith=13; // StringCompareMethod
  pstring;       // Window name (variable-length)
  align word;
 fill byte[6];  // filler for 1 long & 1 StringSpec

 case isOpen:
  key longint = 1;
```

```
  fill byte[4];       // not used
  longint containsAnything=8,
   empty=9,
   matches=10,
   contains=11,
   startsWith=12,
   endsWith=13; // StringCompareMethod
  pstring;       // Window name (variable-length)
  align word;
  fill byte[6]; // filler for 1 long & 1 StringSpec

case isInvisible:
  key longint = 2;
  fill byte[4];       // not used
  longint containsAnything=8,
   empty=9,
   matches=10,
   contains=11,
   startsWith=12,
   endsWith=13; // StringCompareMethod
  pstring;       // Window name (variable-length)
  align word;
  fill byte[6];       // filler for 1 long & 1 StringSpec

case isShareWindowOpen:
  key longint = 3;
  fill byte[16];      // filler for 3 longs & 2 StringSpecs

case isShareWIndowActive:
  key longint = 4;
  fill byte[16];      // filler for 3 longs & 2 StringSpecs
); // Window switch

); // switch
); // type
```

【図面の簡単な説明】

【図１】本発明の現時点で好ましい実施形態で利用できるようなコンピュータシステムの構成要素を示すブロック線図である。

【図２】本発明の現時点で好ましい実施形態のデリバリエンジンの構成要素を示すブロック線図である。

【図３】本発明の現時点で好ましい実施形態のデータベースオブジェクトの間の論理的関係を示す図である。

【図４】本発明の現時点で好ましい実施形態で利用できるようなプレゼンテーションウィンドウ「（著作権）１９９３　Ａｐｐｌｅ　Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．」（１７Ｕ．Ｓ．Ｃ．４０１）を示す。
【図５ａ】本発明の現時点で好ましい実施形態で生成されるデータベースのブロック線図である。
【図５ｂ】トピックと図５ａのデータベースの対応するパネルとの連係を示す。
【図５ｃ】パネルと図５ａのデータベースの対応する視覚オブジェクト及び修飾子との連係を示す。
【図６】本発明の現時点で好ましい実施形態において受け渡すべきトピックをセットアップするためにオブジェクトを検索する過程を示すフローチャートである。
【図７】本発明の現時点で好ましい実施形態においてパネルを受け渡すためにオブジェクトを検索する過程を示すフローチャートである。
【図８】本発明の現時点で好ましい実施形態においてＳｈｏｗ－Ｉｆ又はＳｋｉｐ－Ｉｆ修飾子の評価中に実行される過程を示すフローチャートである。
【図９】本発明の現時点で好ましい実施形態においてＭａｋｅ－Ｓｕｒｅ修飾子を評価するために実行される過程を示すフローチャートである。
【図１０】本発明の現時点で好ましい実施形態においてユーザがゴーバックオプションを呼び出すのに応答してデリバリエンジンにより実行される過程を示すフローチャートである。
【図１１ａ】本発明の現時点で好ましい実施形態で利用できる３つのパネルから構成されるトピックの１例「１９９３　Ａｐｐｌｅ　Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．」（１７Ｕ．Ｓ．Ｃ．４０１）を表わす。
【図１１ｂ】本発明の現時点で好ましい実施形態で利用できる３つのパネルから構成されるトピックの１例「１９９３　Ａｐｐｌｅ　Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．」（１７Ｕ．Ｓ．Ｃ．４０１）を表わす。
【図１１ｃ】本発明の現時点で好ましい実施形態で利用できる３つのパネルから構成されるトピックの１例「１９９３　Ａｐｐｌｅ　Ｃｏｍｐｕｔｅｒ，Ｉｎｃ．」（１７Ｕ．Ｓ．Ｃ．４０１）を表わす。
【図１２】図１１ａ～図１１ｃの例の中に示されているトピックを実現するためのオブジェクト定義のリスティングである。
【図１３】図１１ａ～図１１ｃの例のオブジェクトの連係を示す。

【図１】

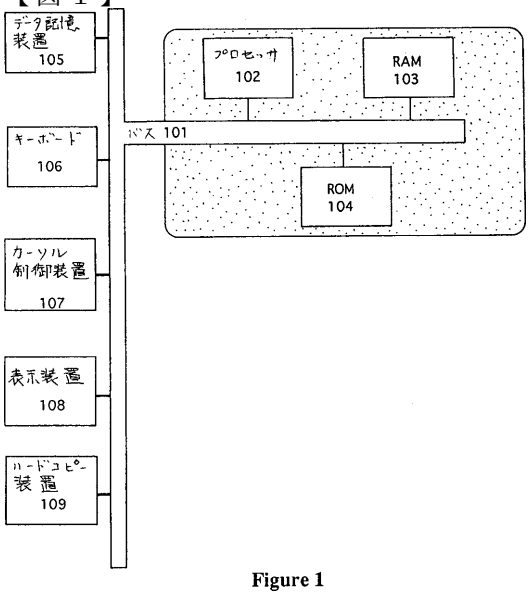


Figure 1

【図２】

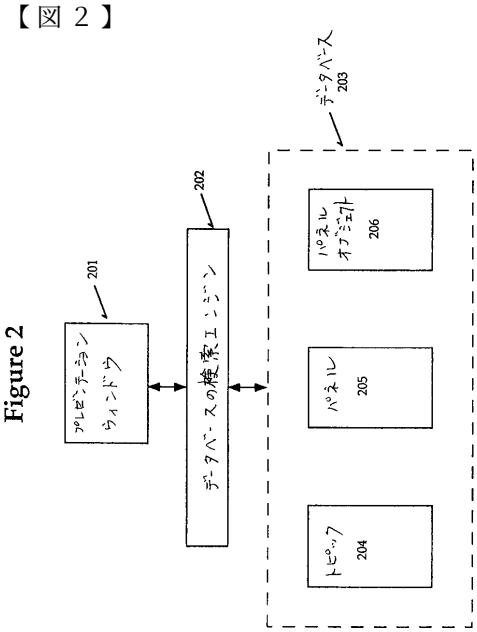


Figure 2

【図３】

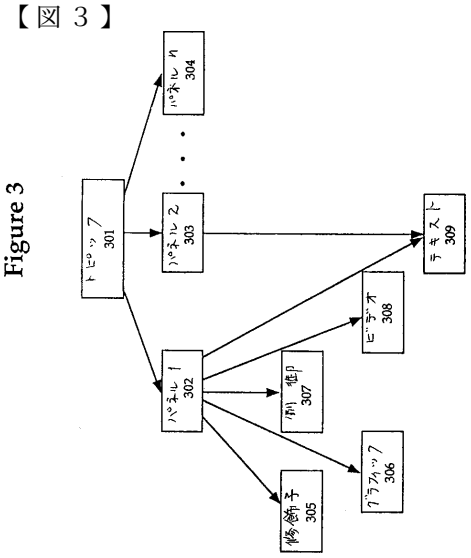


Figure 3

【図４】

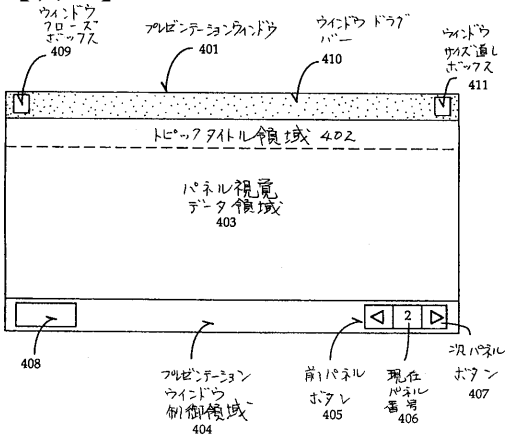
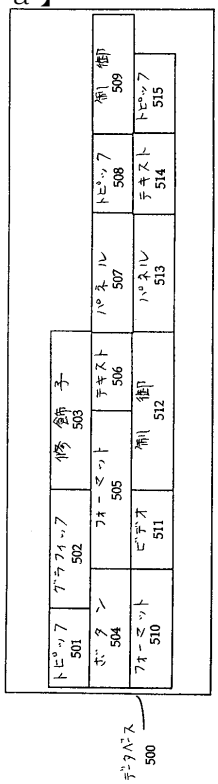


Figure 4

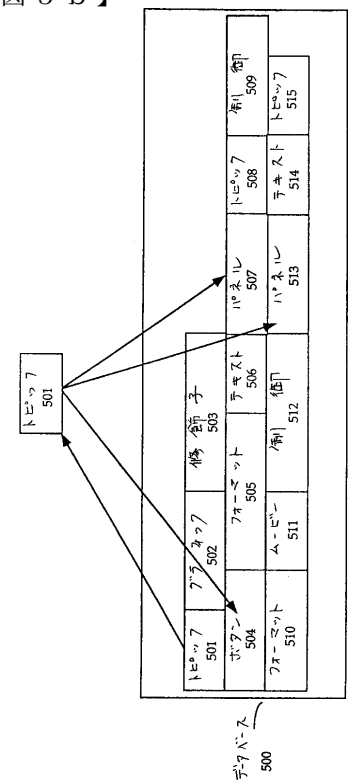
【図５ａ】

Figure 5a



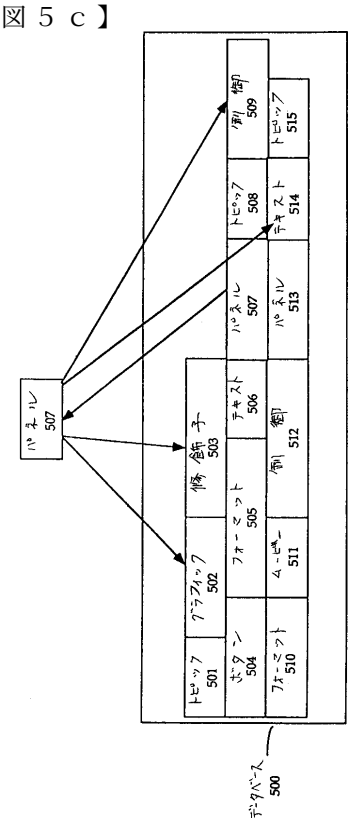
【図５ｂ】

Figure 5b



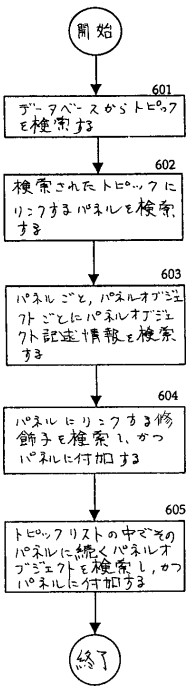
【図５ｃ】

Figure 5c

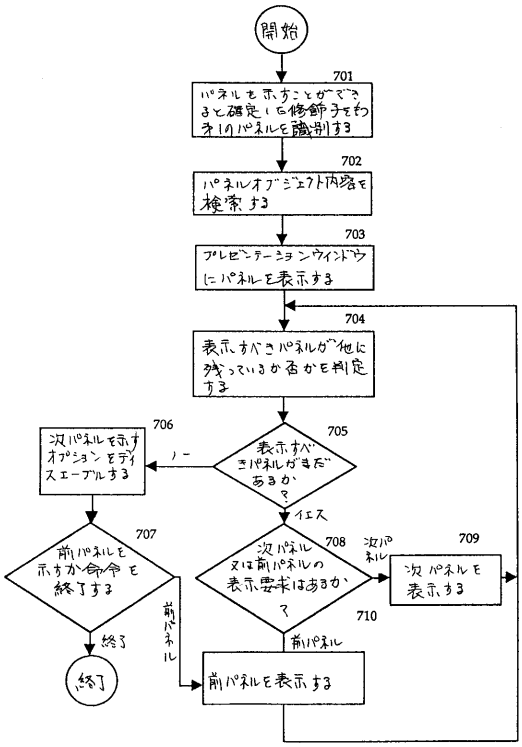


【図６】

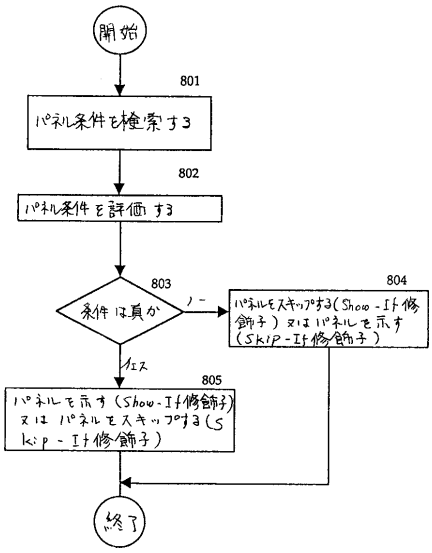
Figure 6



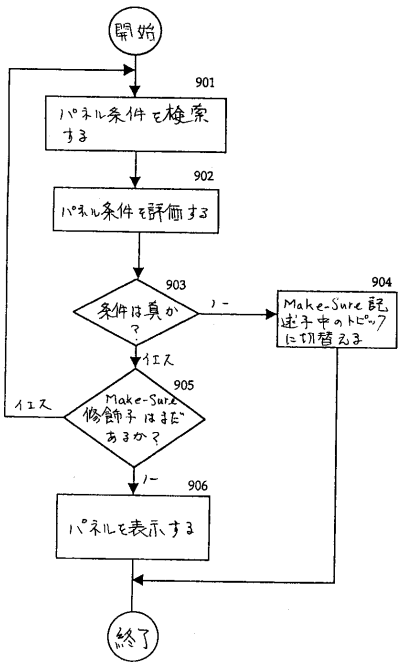
【図７】
Figure 7



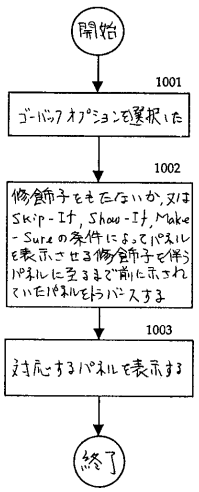
【図８】
Figure 8



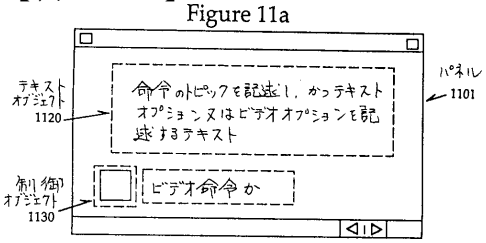
【図９】
Figure 9



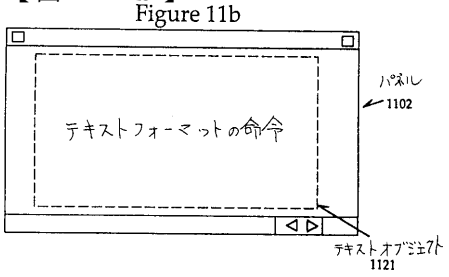
【図１０】
Figure 10



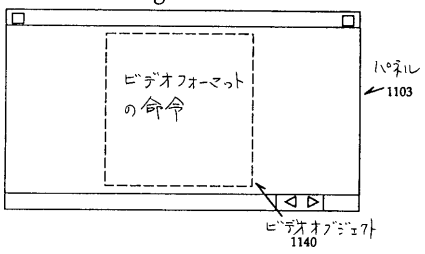
【図１１ａ】
Figure 11a



【図１１ｂ】
Figure 11b



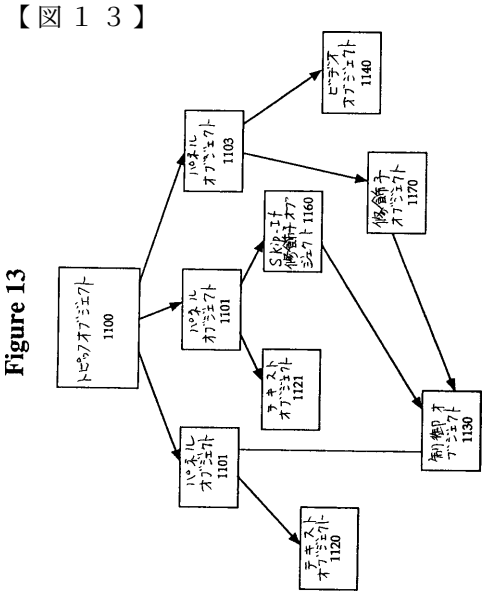
【図１１ｃ】
Figure 11c



【図１２】
FIGURE 12

行番号	オブジェクト定義		コメント
1.	acPT	1100	これはトピックオブジェクトである
2.	(テキスト)	1120	パネル1101に対するテキスト内容
3.	(テキスト)	1121	パネル1102に対するテキスト内容
4.	(ビデオ)	1140	パネル1103に対するビデオ内容
5.	ppUA	1101	パネル1101の定義
6.	pcTX	1120	パネル1101に対するテキストオブジェクト参照
7.	pcCI	1130	パネル1101に対するチェックボックス参照
8.	pcCI	1130 = チェックボックス = ビデオ命令か	チェックボックス制御オブジェクトの定義
9.	ppUA	1102	パネル1102の定義
10.	pcTF	1121	パネル1102に対するテキストオブジェクト参照
11.	pqCS	1160	Skip - If 修飾子参照
12.	pqCS	1160	Skip - If 修飾子の定義
13.	pcCI	1130	条件としてのチェックボックス参照
14.	ppUA	1103	パネル1103の定義
15.	pcMC	1140	パネル1103に対するビデオオブジェクト参照
16.	pqCN	1170	Show - If 修飾子参照
17.	pqCN	1170	Show - If 修飾子の定義
18.	pcCI	1130	条件としてのチェックボックス参照

【図１３】
Figure 13



フロントページの続き

(74)代理人 100098394
弁理士 山川 茂樹
(72)発明者 パワーズ, ジョン・リチャード, ザ サード
アメリカ合衆国 ９５０３０ カリフォルニア州・モンテセレノ・アレクサンダー アヴェニュ・１６３７３

合議体
審判長 小林 信雄
審判官 岩間 直純
審判官 山本 穂積

(56)参考文献 特開平５－７３５２９（ＪＰ, Ａ）
特開平５－２０５０９（ＪＰ, Ａ）
特開平５－６２５１（ＪＰ, Ａ）
YANKELOVICH, N. ET AL, Intermedia:The concept and construction of a seamless information enviroment, COMPUTER, 1988年1月, pp. 81－86
UEDA, S., Hypertext System, FUJITSU－SCIENTIFIC AND TECHNICAL JOUNAL, 1990年, pp. 187－196
Simpson, A., Mastering WordPerfect 5. 1&5. 2 For Windows, 1992年, pp. 68－69
Nielsen, J., HYPERText & HYPERMedia, 1991年, pp. 84－99

(58)調査した分野(Int.Cl., ＤＢ名)
G06F19/00
G06F12/00
G06F17/30