

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成18年4月6日(2006.4.6)

【公表番号】特表2004-523013(P2004-523013A)

【公表日】平成16年7月29日(2004.7.29)

【年通号数】公開・登録公報2004-029

【出願番号】特願2001-545284(P2001-545284)

【国際特許分類】

G 06 F 15/00 (2006.01)

G 06 F 3/048 (2006.01)

【F I】

G 06 F 15/00 3 1 0 S

G 06 F 3/00 6 5 3 A

【誤訳訂正書】

【提出日】平成18年2月16日(2006.2.16)

【誤訳訂正1】

【訂正対象書類名】明細書

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【書類名】明細書

【発明の名称】

アプリケーションの実施を教示するための、ウェブ・ベースのコンピュータ・システム

【特許請求の範囲】

【請求項1】

ユーザにアプリケーションの実施を教示するための、ウェブ・ベースのコンピュータ・システムであって、

1つまたは複数の事前定義された対話型アプリケーションであって、事前定義された対話型アプリケーションが、前記ウェブ・ベースのコンピュータ・システムのユーザによって選択的に実行可能である、前記対話型アプリケーションと、

1つまたは複数の注釈を含む注釈ページであって、前記注釈ページが事前定義された対話型アプリケーションを注釈内で説明する、前記注釈ページと

を含み、

前記注釈ページが、

1つまたは複数のリンクと、

前記アプリケーションの実施の詳細と、

前記アプリケーションにおいて使用されるコードの説明と

を含み、

事前定義された対話型アプリケーションの選択的実行に応答して、前記注釈ページ内に異なる注釈が自動的に提供される、

システム。

【請求項2】

前記ユーザが、事前定義された対話型アプリケーションに関連付けられたソース・コードにアクセスすることができるユーティリティをさらに含む、請求項1に記載のシステム。

【請求項3】

前記ユーティリティが、事前定義された対話型アプリケーションのソース・コードを前記ユーザに表示またはコピーすることを可能にする、請求項2に記載のシステム。

**【請求項4】**

前記アプリケーションの実施の詳細が、前記アプリケーションを説明するテキスト、前記アプリケーションに関連付けられたソース・コードのフラグメント、またはこれらの両方を含む、請求項1に記載のシステム。

**【請求項5】**

リンクが、前記ユーザに参考文書の本文へのアクセスを提供するキーワード・リンク又は前記ユーザに前記注釈ページと別の注釈ページへのアクセスを提供する注釈リンクを含む、請求項1に記載のシステム。

**【請求項6】**

前記注釈を表示する内容フレームと、  
ナビゲーション・バーを表示するフレームワーク・アプレットと、  
リンクの目次階層を表示する目次フレームと  
を有するフレームワークを含むウェブ・ブラウザ・ウィンドウをさらに含む、請求項1に記載のシステム。

**【請求項7】**

前記フレームワーク・アプレットがJava(商標)アプレットを含む、請求項6に記載のシステム。

**【請求項8】**

JavaScriptが、  
前記ウェブ・ブラウザ・ウィンドウにフレームワークが提示されているかどうかを自動的に判断し、前記フレームワークが提示されている場合、前記フレームワーク・アプレットに前記フレームワーク内の内容に関して通知する、請求項6に記載のシステム。

**【請求項9】**

前記目次が、前記フレームワーク内の内容に基づいて前記階層内のリンクを自動的に強調表示する、請求項8に記載のシステム。

**【請求項10】**

前記ユーザが、前記目次階層のリンクを選択することによって、注釈ページにアクセスする、請求項9に記載のシステム。

**【請求項11】**

前記ユーザが、前記ナビゲーション・バーと対話することによって、注釈ページにアクセスする、請求項9に記載のシステム。

**【請求項12】**

前記目次が、内容フレームに表示された注釈ページに基づいて前記階層を強調表示する、請求項9に記載のシステム。

**【請求項13】**

前記目次が除去可能またはサイズ変更可能である、請求項6に記載のシステム。

**【請求項14】**

ユーザにアプリケーションの実施を教示するための、ウェブ・ベースのコンピュータ・システムであって、

注釈を表示する内容フレーム、ナビゲーション・バーを表示するフレームワーク・アプレット及び前記内容フレーム内の内容に関するリンクのグローバル目次階層を表示する目次フレームを含むウェブ・ブラウザ・ウィンドウと、

前記内容フレームに表示される1つまたは複数の注釈であって、前記注釈の夫々が、事前定義された対話型アプリケーションを説明し且つ他の内容へのリンクを含む、前記1つまたは複数の注釈と、

前記表示された注釈のローカル内容に関するリンクのローカル目次階層を表示する目次のウィンドウと

を含む、請求項1に記載のシステム。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】**

本願は、ウェブ(Web)・ベースの文書化および説明に関する。

**【0002】**

本願は、米国仮特許第60/172,134号明細書および米国特許第08/888,925号明細書の特典を請求するものであり、これらの開示の全体を参照により本明細書に組み込む。

**【従来の技術】****【0003】**

図1に示すような典型的なコンピュータ・システムは、中央演算処理装置105と、入出力装置110と、例えばオペレーティング・システム120および1つまたは複数のアプリケーション・プログラム125などの、コンピュータ100によって使用される様々なプログラムを含んでいるメモリ115とを有するコンピュータ100を含む。コンピュータ・システムのエンド・ユーザは、情報を入出力装置110を介してコンピュータ100に転送する様々な入力装置(キーボード130、マウス135)によってコンピュータ100と対話する。コンピュータ100は、この入力されたデータに対して、特に、例えば表示モニター140の画面上に適切なテキストおよび画像を表示することによるなどのエンド・ユーザへの応答出力を提供する方法によって応答する。

**【0004】**

オペレーティング・システム120は、グラフィカル・ユーザ・インターフェース(GUI)を含むことができる。このGUIによって、オペレーティング・システムおよび実行中の可能性を有するいかなるアプリケーション(例えば、ワードプロセッシング・プログラム)でも、そのコンピュータ・システムのユーザと対話することができる。一般に使用されるGUI実施態様は、モニター画面を仮想デスクトップに見立てたデスクトップ・メタフォームを使用する。デスクトップは、基本的に、1つまたは複数の表示領域を含む、様々なグラフィック・オブジェクトをサポートする2次元作業テンプレート領域である。図2に示すように、アプリケーション・プログラムまたはオペレーティング・システムによって生成された情報は、表示領域205(例えば、ウィンドウ、ダイアログ・ボックス、ポップアップ・メニュー、プルダウン・メニュー、ドロップダウン・リスト、アイコン)の範囲内でデスクトップ200上に表示することができる。ユーザは、表示領域内で適切にカーソル210を操作することによって、また、キーボードまたは他の入力装置を用いて情報を入力することによって、オペレーティング・システム、および実行中の可能性のあるいかなるアプリケーションとでも対話することができる。

**【0005】**

コンピュータ100は、通信リンク155(例えば、電話回線)を介してネットワーク150とデータを交換するためのある種の通信カードまたは装置145(例えば、モ뎀またはネットワーク・アダプタ)も含む。ネットワーク150は、例えば、ローカル・エリア・ネットワーク(LAN)、インターネット、またはインターネットであってよい。サービス・プロバイダはネットワークへのアクセスを提供するが、さらに、ネットワークに関連付けられた様々なユーティリティまたはサービス(電子メールなど)を提供することもできる。サービス・プロバイダの例には、AT&T WorldNetなどのインターネット・サービス・プロバイダ(ISP)またはアメリカ・オンライン(America Online)およびコンピューターハウス(CompuServe)などのオンライン・サービス・プロバイダ(OSP)が含まれる。

**【0006】**

開発者は、アプリケーション・プログラムを実施するために、プログラミングの概念を知る必要がある。したがって、アプリケーションの実施態様の説明(アプリケーションの操作の説明だけでなく)が役立つだろう。

**【0007】**

多くのコンピュータ・アプリケーションは、アプリケーションの使用を支援するオンライン・ヘルプ/文書化機能を提供する。図3に示すような典型的なオンライン・ヘルプ・システムはGUIによってアクセスされるが、ここでは、ヘルプ・ウィンドウ300内でユーザに対してテキスト情報およびグラフィック情報の画面が表示される。次いでユーザは

、アプリケーションとその様々な機能のより良い理解を得るためにヘルプ・テキストの画面を読むことができる。

#### 【0008】

ユーザは、キー・シーケンスによって（例えば、キーボードのF1キーを押し下げるここと）、または適切なグラフィック・アイコンまたはメニュー項目上でマウスをクリックすることによって、ヘルプ・システムを呼び出す。それに応答して、ヘルプ・システムは、ユーザが要望通りにアクセスし表示することができる使用可能なヘルプ・トピックスおよびサブトピックスを列挙する目次305を表示することができる。ユーザは、目次305全体を参照し、関心のあるヘルプ・トピックをクリックして、対応する情報の本文をヘルプ・ウインドウの中に表示させることができる。図3に示すヘルプ・ウインドウ300で、ユーザは、対応するヘルプ画面315を図4に示すウインドウ300に表示させるよう既に「マイクロソフト・ワードでプログラムする（Programming with Microsoft Word）」を選択済みである。

#### 【0009】

図4に示す「マイクロソフト・ワードでプログラムする（Programming with Microsoft Word）」トピック310は、別個の「リンク」によって示される複数のサブトピック320を含む。ユーザがそれらのリンクの1つ、例えば「エラー・メッセージ（Error Messages）」リンク325の上でマウスをクリックすると、図5に示すように、対応するヘルプ・トピックのテキストがヘルプ・ウインドウ300に自動的に表示される。この実施例では、「エラー・メッセージ（Error Messages）」トピック330は、特定種類のエラー・メッセージに関するさらなるサブトピックへの複数のリンクを含む。図6に示すように、ユーザがそれらリンクのうちの1つ、例えば「メモリ不足（スタックスペース）（Out of memory (stack space)）」リンク335をクリックすると、その選択されたトピックに関する対応するヘルプ情報（「メモリを解放する（Freeing up memory）」）を表示するために新しいヘルプ・ウインドウ340が作成される。ウインドウ340に表示されたヘルプ情報は、別のサブトピック「アクティブ・ウインドウ（active window）」へのさらに別のリンク345を含むが、これは、ユーザがそれをクリックすると、対応するヘルプ・テキストをポップアップ・ダイアログ・ボックス350内に表示させる。ほぼすべてのレベルのそのようなネストされたヘルプ表示が可能である。ヘルプ情報を表示するために使用される表示領域（ウインドウ、ダイアログ・ボックスなど）の量および種類は、主としてヘルプ・システム開発者の好みに基づく設計上の選択の問題である。

#### 【0010】

ヘルプ・システムは「文脈に依存した」ヘルプ情報を提供することができる。すなわち、そのヘルプ・システムが、すべての入手可能なヘルプ・トピックスを単純に表示して、適切なヘルプ・トピックをユーザが手動で識別して読み出すように強いるのではなく、アプリケーションの現行タスクに特に関係したヘルプ情報を自動的に表示するということである。文脈に依存したヘルプ・システムは、アプリケーションの現行状況（例えば、ユーザによって呼び出された特定の関数）および現在のカーソル位置などの要因に基づいて、どのヘルプ情報を表示するかを決定する。

#### 【0011】

大部分のオンライン・ヘルプ・システムによって提供される情報は、アプリケーションの使用機能の機構に関する。図7では、例えば、選択されたヘルプ・トピック405「印刷をキャンセル（Cancel printing）」に対応するテキスト400は、アプリケーション410（マイクロソフト（商標）ワード（Microsoft（商標）Word））によって提供される印刷機能をどのように制御するかを説明している。

#### 【0012】

ヘルプ・システムは、所望の目的を達成するためにアプリケーションをどのように活用するかに関する価値ある情報を提供することもできる。図8では、例えば、オンライン・ヘルプ・システムは、ワードベーシック（WordBasic）プログラミング言語用の参考資料500と、ワードベーシック（WordBasic）を使用して有用なプログラムを作成するための方

法に関する実践的な説明 5 0 5 の二種類の価値ある情報を提供する。参考資料 5 0 0 は、AddAddInステートメントなどの様々なワードベーシック(WordBasic)ステートメントの構文規則および意味を説明するテキスト注釈を含むが、それに関するヘルプ・テキストを図 9 に示す。実践的な説明 5 0 5 は、ワードベーシック(WordBasic)プログラミング言語をより良く理解するためにユーザが学ぶことのできるプログラム・コードの静的な例を含むことができる。図 1 0 に、GetCurValuesWordBasicステートメントを使用するプログラム・コードの一例を示す。

#### 【0 0 1 3】

##### 【発明が解決しようとする課題】

オンライン・ヘルプ・システムは、通常、テクニカル・ライターによって既に作成済みのヘルプ情報を含んでいる複数の異なるヘルプ・ソース・コード・ファイルをコンパイルすることによって「構築」される（すなわち、ランタイム操作を容易にする形式で処理される）。一般に、これらのヘルプ・ソース・コード・ファイルは、ヘルプ・システムが対応するアプリケーションとは別の情報の別個の本文として保存される。したがって、アプリケーション開発者がアプリケーションの機能性を変更または更新するとき、テクニカル・ライターは、オンライン・ヘルプ・システムがそのアプリケーションの動作を正確に説明することを保障するために、ヘルプ・ソース・コード・ファイルに対して対応する変更または更新を行う必要がある。しかし、一般に、オンライン・ヘルプ・システムは、アプリケーションの実施を説明できていない。

#### 【0 0 1 4】

ヘルプ・システムは「ブラウザ」を使用してネットワーク環境で実施することができ、このブラウザは、ユーザがネットワーク環境で保管されている電子内容にアクセスし、これを表示させることを可能にする。ブラウザは、通常、ハイパーテキスト・マークアップ言語（HTML）で記述された文書を表示するために使用され、インターネットなどのネットワークに接続されたサーバ上に保管される。図 1 1 は、典型的なHTML文書またはウェブ・ページ 6 0 5 を表示するブラウザ・アプリケーション 6 0 0 （この場合はインターネット・エクスプローラ）の画面の一場面である。ユーザは、所望の文書が常駐するネットワーク・アドレス 6 1 0 、すなわちユニフォーム・リソース・リケータ（URL）を指定することによってウェブ・ページ 6 0 5 にアクセスするようブラウザ 6 0 0 に指示する。それに応答して、ブラウザ 6 0 0 は、要求されたウェブ・ページをホストする対応するサーバに接触し、そのウェブ・ページを構成する 1 つまたは複数のファイルを取り出し、次いでコンピュータのディスプレイ 1 4 0 にそのウェブ・ページを表示する。

#### 【0 0 1 5】

1 つのウェブ・ページは、潜在的にデータ・タイプの異なる複数の異なるファイル（例えば、テキスト 6 1 5 、イメージ 6 2 0 、仮想世界、音声、またはムービー）から構成されることができる。さらに、ウェブ・ページは、ネットワーク上で使用可能な別の資源（例えば、ウェブ・ページ、個別ファイル、またはダウンロード可能なファイル）へのリンク 6 2 5 またはポインタを含むことができる。各リンクは、ネットワーク上の位置を指す、関連付けられた URL を有している。表示されたリンクをユーザがクリックするか、または別の手段で選択すると、ブラウザは、そのリンクの関連付けられた URL に対応するウェブ・ページまたは他の資源を取り出し、それをユーザに対して表示または実行する。

#### 【0 0 1 6】

##### 【課題を解決するための手段】

図 1 2 を参照すると、ウェブ・ページ 6 0 5 は、内容 7 0 0 の他に、そのウェブ・ページに関連付けられたすべてのリンクを通してユーザがナビゲートするのに役立つサイト・ガイド 7 0 5 を提供することができる。サイト・ガイド 7 0 5 は目次に類似したもので、通常はツリー構造に似ている。同様に、ウェブ・ページ 6 0 5 は、そのウェブ・ページに関連付けられたリンク内にある特定のキーワードをユーザが検索することを可能にする検索機能 7 1 0 を含むことができる。このウェブ・ページは、すべての内容およびリンクにそこからアクセスすることができるメインウェブ・ページにユーザを引き戻す「ホーム」

リンク 715 を提供することができる。このウェブ・ページは、アクセスされたときに別のウェブ・ページまたはコンピュータからユーザのコンピュータにファイルを送信するダウンロード・リンク 720 を提供することができる。

【0017】

#### 【発明の実施の形態】

本発明の一態様によれば、コンピュータ・システムのウェブ・ベース環境で実行される方法は、ユーザにアプリケーションの実施の仕方を教示する。この方法は、所定のアプリケーションを提供することと、所定のアプリケーションのソース・コード・ファイルを説明する1つまたは複数の注釈を含む注釈ページを提示することを含む。各注釈は、そのアプリケーションのキーワード・リンク、注釈リンク、および実施の詳細を含む。方法は、ユーザが注釈内のリンクを選択できるようにする。ユーザがキーワード・リンクを選択した場合、そのキーワードに関連付けられた参考文書が提示される。ユーザが注釈リンクを選択した場合、所定アプリケーションの別のソース・コード・ファイルを説明する別の注釈が提示される。

【0018】

実施形態は、1つまたは複数の以下の機能を含むことができる。例えば、所定アプリケーションを実行することができ、実行されたアプリケーションを説明する1つまたは複数の注釈を所定アプリケーションの実行と連携して提示することができる。所定アプリケーションを実行することは、ユーザからの入力を受け取ることを含むことができる。ユーザからの入力に基づいて、所定アプリケーションの実行と連携して別の注釈ページを提示することができる。

【0019】

別の注釈ページを提示することは、詳細が表示されるべきプログラム・ユニットのアプリケーション、ファイル、クラスおよび関数の名称を含む注釈要求モジュールを自動的かつ同時に呼び出すことを含むことができる。別の注釈ページを提示することは、その要求を注釈にマッピングすること、ウェブ・ベース環境のブラウザ・ウィンドウに別の注釈ページを表示するよう伝達することも含む。

【0020】

所定アプリケーションの実行と連携して別の注釈ページを提示することができる。注釈ページを含むウェブ・ページの構造化リンクを構文解析することによって、注釈へのリンクを含むグローバル目次が自動的に生成される。グローバル目次内のリンクの生成は、現行の注釈ページに対応するリンクを強調表示することによって注釈の提示と同期することができる。グローバル目次は第1のブラウザ・ウィンドウの第1のフレーム内に提示することができ、注釈ページは第1のブラウザ・ウィンドウの第2のフレーム内に提示することができ、所定アプリケーションは第2のブラウザ・ウィンドウ内で実行することができる。

【0021】

所定アプリケーションを実行することは、Java(商標)アプレットまたはアプリケーションを起動することを含むことができ、これは、ウェブ・ブラウザに注釈ページを表示するよう要求するためにJava(商標)アプリケーション・プログラム・インターフェースを呼び出すことを含むことができる。所定アプリケーションを実行することは、Java(商標)アプレットを含んでいるハイパーテキスト・マークアップ言語のページをダウンロードすることを含むことができる。

【0022】

所定アプリケーションを実行することは、共通ゲートウェイ・インターフェース要求をウェブ・サーバに送信することを含むことができる。このウェブ・サーバは、ウェブ・ベース環境のウィンドウでアプリケーションを起動する。アプリケーションは、ウェブ・ブラウザに1つまたは複数の注釈を表示するよう要求するJavaScriptを含むハイパーテキスト・マークアップ言語のページを戻すことができる。

【0023】

注釈ページを第1のブラウザ・ウィンドウに提示することができ、所定アプリケーションを第2のブラウザ・ウィンドウで実行することができる。アプリケーションの実施の詳細は、そのアプリケーションを記述しているテキスト、そのアプリケーションからのソース・コードのフラグメント、またはこれらの両方を含むことができる。ソース・コードのフラグメントは、提示されたアプリケーションのソース・コード・ファイルから直接的にインポートされることができる。

#### 【0024】

注釈ページを自動的に生成することができる。この生成には、説明がマークアップされたテキストを既に埋め込んでいるソース・コード・ファイルを受け取ることが含まれる。さらに、所定アプリケーションの構造を決定するためにソース・コードを構文解析することができ、所定アプリケーションの構造および説明に基づいて1つまたは複数の注釈を生成することができる。注釈ページの生成は、所定アプリケーションの注釈をナビゲートするための1つまたは複数の注釈リンクを生成することを含むことができる。さらに、埋め込まれた情報に基づいてアプリケーションの実施の詳細を生成することができ、参考文書に対して1つまたは複数のキーワード・リンクを生成することができる。注釈ページを生成することは、注釈ページのキーワード・リンクおよび注釈リンクを強調表示することを含むことができる。更新されたソース・コード・ファイルを受け取ったときに、注釈ページを自動的に更新することができる。

#### 【0025】

注釈リンクに対する1つまたは複数の注釈を構文解析することによって、グローバル目次を自動的に生成することができる。このグローバル目次は、提供されることができ、注釈へのリンクを含むことができる。別法として、グローバル目次は、生成されることができ、アプリケーションに対する注釈ページを含めてウェブ・ページへのリンクを含むことができる。グローバル目次内のローカル・リンクが選択されたときは、ローカル目次を提供することができる。

#### 【0026】

提示された注釈ページは実行されるアプリケーションを記述することができ、その注釈ページは、所定アプリケーションの実行と連携して提示されることができる。

#### 【0027】

注釈マークアップを除去し、アプリケーションのソース・コードは含むが注釈からのテキストは含まないソース・コード・ファイルを生成することができる。除去済みのソース・コード・ファイルを提示することができ、その除去済みのソース・コード・ファイルをユーザが編集することを可能にすることができる。

#### 【0028】

本発明の別の態様によれば、コンピュータ・システムのウェブ・ベース環境で実行される、ユーザにアプリケーションを実施することを教示する方法は、所定の複数アプリケーションを提供することを含む。所定アプリケーションが実行され、その実行されたアプリケーションを記述する注釈ページを、所定アプリケーションの実行と連携して提示することができる。注釈ページは、アプリケーションの実施の詳細と、注釈および参考文書へのリンクを含む。

#### 【0029】

本発明の別の態様によれば、コンピュータ・システムのウェブ・ベース環境で実行される、ユーザにアプリケーションを実施することを教示する方法は、その環境内における内容に基づいてグローバル目次を自動的にアセンブルし、提供することを含む。グローバル目次は、環境内の内容への複数のリンクを含む。ローカル・トピック内でユーザを方向付ける内容へのリンクを含むローカル目次が生成される。ユーザは、ローカル・トピックスにアクセスするために、ローカル目次からリンクを選択することが許可される。

#### 【0030】

本発明のさらなる態様によれば、コンピュータ・システムのウェブ・ベース環境で実行される、ユーザにアプリケーションを実施することを教示する方法は、複数の事前定義さ

れた対話型の例示を提供することを含む。1つまたは複数の事前定義された対話型の例示は、ユーザ選択に応答して実行され、その実行されている対話型の例示を記述する1つまたは複数の注釈は、事前定義された対話型の例示の実施と連携して提示される。ユーザは、実行されている対話型の例示、注釈、またはその両方の異なる態様を選択的に探索することができる。

#### 【0031】

本発明の別の態様によれば、ユーザにアプリケーションを実施することを教示するウェブ・ベースのコンピュータ・システムは、1つまたは複数の事前定義された対話型アプリケーションと、1つまたは複数の注釈を含む注釈ページとを含む。事前定義された対話型アプリケーションは、ウェブ・ベースのコンピュータ・システムのユーザによって選択的に実行可能である。注釈ページは、事前定義された対話型アプリケーションを説明する。注釈ページは、1つまたは複数のリンクと、アプリケーションの実施の詳細とも含む。事前定義された対話型アプリケーションの選択的な実行に応答して、異なる注釈が注釈ページに自動的に提供される。

#### 【0032】

本発明のさらなる態様によれば、ユーザにアプリケーションを実施することを教示するウェブ・ベースのコンピュータ・システムは、内容フレーム、フレームワーク・アプレット、および内容フレームの内容に関するリンクのグローバル目次の階層を表示する目次フレームを含むウェブ・ブラウザ・ウィンドウを含む。システムは、内容フレーム内に表示された1つまたは複数の注釈も含み、そこで、各注釈は、事前定義された対話型アプリケーションを説明し、別の内容へのリンクを含む。このシステムは、表示された注釈においてローカル内容に関するリンクのローカル目次の階層を表示する目次ウィンドウを含む。

#### 【0033】

1つまたは複数の実施形態の詳細を、添付の図面と以下の説明で述べる。この他の機能、目的、および利点は、説明、図面、および特許請求の範囲から明らかになろう。

#### 【0034】

従来のオンライン・ヘルプ・システムによって提供されるヘルプ情報は、ユーザがアプリケーション・プログラムを効果的に活用することを支援する上で有用であることは既に証明済みである。しかし、これらの従来のオンライン・ヘルプ・システムは、基本的に静的テキストまたはグラフィック情報を提供することに限定されているので、それらの有効性は大幅に低下する。従来のオンライン・ヘルプ・システムのユーザは、ヘルプ・システムによって提供されるテキスト情報およびグラフィック情報を読み、注意深くこれを検討し、次いでその教示を解決すべき問題点に適用することによって解決を得る。その結果、少しでも複雑なアプリケーションを使用することを学ぶことは、しばしば面倒で時間のかかるプロセスである。

#### 【0035】

NewEra(商標)byExampleとして知られているInformix(商標)Software社によって開発されたオンライン・ヘルプおよび指示システムは、ユーザが情報を吸収し、新しいアプリケーションの使用法を学習することのできる容易さを劇的に強化する。NewEra(商標)byExampleは、Windows(商標)95/NTオペレーティング・システム下で実行するNewEra(商標)に関するオンライン文書化機能であり、Informix(商標)Software社製のオブジェクト指向アプリケーション開発環境である。NewEra(商標)byExampleのそれ自体の使用法および操作に関するオンライン説明を付録Aとして添付する。

#### 【0036】

NewEra(商標)byExample(またはより一般的には、NewEra(商標)、VisualBasic(商標)およびJava(商標)開発環境に提供されている例示ベースの説明システムをカバーするInformix(商標)byExample)は、所要のタスクをどのように達成するかを実例で示す、動的で対話型の例示がユーザに提供される「例示によるドキュメンテーション」と称されるより一般的な概念の特定の実施である。例示の様々な態様を説明する注釈は、実行中の例示に付随する。例示によるドキュメンテーションは、一部には、ユーザが、単にトピ

ックに関して読むことによるだけでなく、何かを実行する（例えば、ある活動に参加して、その結果を監視したり、その結果に影響を与える）ことによって最もよく学ぶという前提に基づいている。

#### 【0037】

図13の流れ図に示すように、Informix(商標)byExampleのユーザは、様々な異なるトピックから選択すること（ステップ800）、それに対するヘルプが探索されるアプリケーションがアクティブに維持されている間に例示を実行すること（ステップ805）、例示を実行中に同時に、またはその例示とは関係なく、例示に関して読むこと（ステップ810）、異なるエディタ・ユーティリティで例示のソース・コードを検査すること（ステップ815）、およびユーザが例示を理解するのに役立つオンライン・バックグラウンド参考資料にアクセスすること（ステップ820）を含む情報を獲得するためのいくつかの異なるオプションを有し、これらはすべてヘルプ環境を離れずに行われる。ステップ815では、例示のためのソース・コードを、新しいアプリケーションを作成する際にNewEra(商標)開発環境でテンプレートとして使用するために、カット・アンド・ペーストすることができるサンプル・プログラムとして使用することができる。さらに、Informix(商標)byExampleは、試行、例えば、例示またはそれらのパラメータを選択的に変更し、かつその変更が例示の結果にどのように影響を与えるかを監視することによってユーザが学ぶことを可能にする。

#### 【0038】

Informix(商標)byExampleの特定の機能を、Informix(商標)byExampleアプリケーションからとられた例示的画面の一場面である図14～29を参照して詳述する。

#### 【0039】

ユーザが最初にInformix(商標)byExampleを起動するとき、図14に示すデフォルト画面構成が表示される。この初期画面には、現行トピック902下でユーザに対して現在使用可能なサブトピックを示すリスト（または「目次」）ウィンドウ900と、リスト・ウィンドウ900から選択されたトピックまたはサブトピックに対応するヘルプ情報を表示するテキスト・ウィンドウ904との2つの別個の表示ウィンドウが含まれる。リスト・ウィンドウ900に表示された8個のサブトピック906の様々なサブトピックをユーザがクリックするとき、テキスト・ウィンドウ904内の情報は、選択されたサブトピック906に対応するように自動的に更新される。ユーザは、必要に応じて、前進（「>>」）ボタン908または後退（「<<」）ボタン910をクリックすることによって現行トピック内の異なるページに移動することができる。

#### 【0040】

図14の例示では、リスト・ウィンドウ900に示されるサブトピックは、トピック「NewEra(商標)byExample導入トピック(Introductory Topics)」に関する。別のヘルプ・トピックに切替え、それによって、オンライン・ヘルプ資料の異なるサブセットを使用可能にするために、ユーザは、図15に示すように、9つの異なるヘルプ・トピックのリスト916を含む「ヘルプ・トピック(Help Topics)」メニュー914を含んでいるウィンドウを提示する「ヘルプ・トピック」ボタン912をクリックする。Informix(商標)byExampleアプリケーションのどのポイントにおいても、ユーザは、「ヘルプ・トピック」メニュー914を提示させ、所望のトピックをクリックすることによって、オンライン・ヘルプ・システムの別の任意の部分にジャンプすることができる。ユーザは、適切な回数だけ「戻る(Back)」ボタン916を押し下げることによって前のトピックに戻ることができる。

#### 【0041】

図15のリスト916内のヘルプ・トピックのそれぞれは、複数レベルのサブトピックの階層を開示するために拡張することができる。ユーザが例えばトピック918（「NewEra(商標)byExample」）をクリックすると、そのトピックは、図16に示すサブトピックの2つの付加的なレベルを開示するために拡張する。2つの付加的なレベルとはすなわち、サブトピック「導入(Introduction)」「一般的なNewEra(商標)プログラミング技

術 (Common NewEraProgramming Techniques)」および「例 (The Examples)」を含む第1のレベル920と、43個の対話型の例示を含む「例 (The Examples)」サブトピック下の第2のレベル922である。

#### 【0042】

ユーザが例示、例えば「許可及び不許可ボタン (Enabling and Disabling Buttons)」の例示924をクリックすると、選択された例示に関連付けられた注釈セグメント926 (「ボタン2の概観 例 (Overview of Buttons2 Example)」、「図解的オブジェクト要約 (Graphical Object Summary)」、「イベント・ハンドラ要約 (Event Handler Summary)」、「重要なイベント・ハンドラ (Important Event Handlers)」および「追加の機能及びバリエーション (Enhancement and Variations)」)を表示するためにリスト・ウィンドウ900が図17に示すように更新される。注釈セグメント926は、対応する例示を一括して説明し、例示のウィンドウ、そのグラフィック・オブジェクト、およびそのイベント・ハンドラの説明を含む。例示の散文説明に加え、注釈セグメントは、通常、検討中の例示のソース・コードから直接的にインポートされた特定の対象のソース・コード・フラグメントも含む。

#### 【0043】

図17に示すように、注釈セグメントの1つ (「重要なイベント・ハンドラ (Important Event Handlers)」) は13個のトピック928を含むが、これはボタン2 (BUTTONS2)の例示で使用される主イベント・ハンドラのリストである。各イベント・ハンドラのトピック928は、ソース・コード・フラグメントと、トピックが対応するイベント・ハンドラを説明する散文説明を含む。例えば、ユーザがイベント・ハンドラ・トピック930をクリックすると、テキスト・ウィンドウ904は、図18に示すように、対応するイベント・ハンドラ (nextBT::activate())に関するソース・コード・フラグメント932をコードの操作を説明する注釈934と共に表示する。

#### 【0044】

テキスト・ウィンドウは、関係する情報への1つまたは複数のリンク、例えばバックグラウンド参考資料を含むことができ、このバックグラウンド参考資料は、付加的なバックグラウンド情報へのさらなるリンクを含むことができ、階層方式によってこれはさらに累々と続く。階層における各連続的なレベルは、例示に関する情報をさらに詳細に、かつさらに低レベルの抽象性で提供する。この方法でさらに詳細な文書に対するリンクの階層を提供することによって、Informix (商標) by Exampleは、ユーザの経験と知識のレベルが様々であることに関係なく、すべてのユーザに対して文脈に合った情報を有用かつ有効な方法で提供する。ユーザは、ユーザのニーズとバックグラウンドに相応の理解のレベルに到達するまで、説明的情報の階層リンクを選択的に横断して降下することができる。この構成は、新人ユーザに対しては、詳細な説明情報への容易なアクセスを提供し、同時に、特定の点に関するヘルプを探している経験豊富なユーザを多量の不要な情報の中をナビゲートしなければならないことから保護する。

#### 【0045】

図17に階層リンクの一例を示す。ここでは、テキスト・ウィンドウ904は、1つの関数はボタン2 (BUTTONS2)の例示の関数のうちの1つであるMAIN()関数へのリンク936 (緑色の下線を引いたテキストによって表示される)を含む。ユーザがMAIN()関数リンク936をクリックすると、図19に示すようにテキスト・ウィンドウ904はその関数に対するソース・コード940を表示する。ソース・コード940は、キーワードおよびオブジェクトのクラスの説明を含むオンライン言語解説書などの関連情報へのさらなるリンクを含む。ユーザがそれらのリンクの1つ、例えばLETステートメントへのキーワード・リンク942をクリックすると、テキスト・ウィンドウ904は、図20に示すように対応するオンライン言語解説項目を表示するように変わる。同様に、ユーザがオブジェクト・クラス・リンク944をクリックした場合、テキスト・ウィンドウ904はixSQLConnectクラスに関する情報を既に表示しているはずである。図20では、ユーザは、例えば、テキスト・ウィンドウ904を図21のように表示させるようにオブジェクト表現 (Ob

ject Expression) ボックス 946 をクリックすることによって、さらなるバックグラウンド情報へのリンクに従うことができる。その後、または別法として、ユーザは、図 22 に示すように、用語集目次ウィンドウ 950 および用語集テキスト・ウィンドウ 952 の一対のウィンドウ中にオンライン用語集を提示するために用語解説 (Glossary) ボタン 948 をクリックすることができる。用語集目次ウィンドウ 950 の用語をクリックすることによって、その用語の定義が用語集テキスト・ウィンドウ 952 に表示される。

#### 【0046】

例示の注釈、そのソース・コード・フラグメント、対応する言語解説項目、用語集、またはそれらの組み合わせを検討した後、ユーザは、図 15 に示すようにヘルプ・トピック (Help Topic) メニュー 914 を提示させる内容 (Contents) ボタン 954 (または検索可能な索引形式の使用可能なヘルプ・トピックを提示する索引 (Index) ボタン 956) をクリックし、次いで図 15 および 16 に関連して説明した方法で所望のトピックをユーザが選択することによって、ヘルプ・システム内の任意の別の位置に選択的にジャンプすることができる。

#### 【0047】

図 19 のLETステートメント・リンク 942 および ixSQLConnectクラス・リンク 944 などのキーワード・リンクおよびクラス名リンクは、それぞれに、可視的な固有の方法 (例えば、キーワードには青色の上付きテキスト、クラス名には青色の下付きテキスト) で示され、その結果、容易に相互から区別することができ、また、図 17 に示すように MAIN () 関数リンク 936 (緑色の下付きテキスト) などの他の種類のリンクからも容易に区別することができる。異なる種類のリンクに対して異なるスタイルを用いることによって、Informix (商標) byExampleは、使用可能なオンライン情報の性質および例示の異なる構成要素 (注釈、ソース・コード・フラグメント、言語解説など) 間の相互関係に関して直感的に有用な情報をユーザに提供する。ほぼ任意の数の異なるリンクの種類は、システム設計者の好みに応じて異なるスタイルで表示される場合がある。

#### 【0048】

例示の注釈に含まれるソース・コード・フラグメントのそれぞれについて、ユーザは、例示のソース・コードを検査し、編集し、またはコピーするために、Informix (商標) by Exampleから適切な編集ユーティリティを呼び出すことができる。これによって、ユーザは、ソース・コード・フラグメントが取り出されるさらに大きなプログラムに照らしてソース・コード・フラグメントを表示させることができる。

#### 【0049】

Informix (商標) byExampleは、2つの異なる種類のソース・コードに由来するソース・コード・フラグメントを含む。それらは、NewEra (商標) プログラミング言語のテキスト・プログラム・コード (4GLまたは4GHファイル接尾辞によって示される) と、開発中のアプリケーションのエンド・ユーザに対してGUIをどのように表示し、そのエンド・ユーザとどのようにインターフェースするかを定義するウィンドウ・インターフェース定義ファイル (接尾辞WIFを有するファイル) である。どちらか一方の種類のソース・コード・フラグメントを表示するために、ユーザは、コード・フラグメントの隣にあるショートカットの矢印、例えば図 17 ~ 19 に示すショートカット矢印 958 および 960 のうちの1つをクリックする。Informix (商標) byExampleはこれに応答して、検討中のソース・コードの種類に対応するエディタを起動する。ユーザが、図 17 および 19 のショートカット矢印 958 などの4GHまたは4GLファイルの隣のショートカット矢印をクリックすると、Informix (商標) byExampleは、図 23 に示すように、コード・フラグメントがそこから取り出されたソース・コード・ファイルを表示させるために、適切なエディタであるNewEra (商標) Codewriterを自動的に起動する。同様に、ユーザが、図 17 および 18 のショートカット矢印 960 などのWIFファイルの隣のショートカット矢印をクリックすると、Informix (商標) byExampleは、図 24 に示すように、コード・フラグメントがそこから取り出されたWIFファイルを表示させるために、適切なエディタであるNewEra (商標) Window Painter3.0を自動的に起動する。

## 【0050】

この方法で複数のエディタのうちから適切なエディタを選択的に起動することは、NewEra(商標)開発環境の標準的な編集動作を反映する。NewEra(商標)開発環境と、Informix(商標)byExample資料システムの両方とも、同じ方法で同じエディタを利用する。その結果、ユーザは、オンライン・ヘルプ・システム(すなわち、Informix(商標)byExample)との通常の対話によって、ヘルプを捜すアプリケーション(すなわち、NewEra(商標)開発環境)に精通する。

## 【0051】

ユーザが例示に対するソース・コードの開発を完了すると、WIFファイルのビジュアル・オブジェクトまたは4GHまたは4GLファイル内のプログラム・ステートメントがユーザ自身のソース・コード・ファイルに入っているかどうかに関わらず、ユーザは、簡単にそのコードを検討したり、そのコードの一部をカット・アンド・ペーストしたりすることができる。別法として、ユーザは、「...として保存する(Save As...)」操作を実行することができ、それによって、新しいファイル名でその例示に対するソース・コードを保存することができる。次いでユーザは、必要に応じて新しいファイルを編集し、または他の方法で操作することができる。この方法で、Informix(商標)byExampleによって提供された例示は、NewEra(商標)開発環境で新しいアプリケーションを開発する際に使用するためのテンプレートとしての機能を果たすことができる。

## 【0052】

ユーザは、それらの動作方法を直接的に監視するためにInformix(商標)byExampleに提供された43個の対話型の例示のどれかまたはすべてを実行することができる。例示は事前に構築されており、それらの対応するInformix(商標)byExample注釈から直接的に起動されることができる。これを実行するためには、ユーザは、まず図16に示すヘルプ・トピック(Help Topics)ウィンドウ914から関心のある例示を選択し、対応する注釈がテキスト・ウィンドウに表示されると、そのテキスト・ウィンドウ上部付近に表示されている実行(Run)ボタンをクリックする。それに応答して、例示が実行され、また、ユーザから受け取った入力に基づいて、その例示がスタンダード・アプリケーションの場合は様々な画面をユーザに対して表示する。同時に、テキスト・ウィンドウは、ユーザによって現時点で実行中の例示の一部に関連する説明情報を表示させるために自動的に更新する。ユーザが実行中の例示で実行している各連続的な操作によって、ユーザに対して現在例示に何が起こっているかを説明する注釈の対応するセクションを表示することによって、対話型の例示の状況との同期を維持するために、テキスト・ウィンドウが同時に(またはほぼ同時に)更新される。ヘルプ表示を例示の現在の状態と調整することによって、ユーザの関心のある現行トピックに直接的に関連するタイムリーかつ有用な情報(例えば、その例示によって実行中の特定のソース・コード)が一貫してユーザに提供される。その結果、情報を理解し吸収するユーザの能力は劇的に高められる。Informix(商標)byExampleの例示の自動的に調整されたヘルプ表示を図25~29に示す。

## 【0053】

図25は、ユーザがヘルプ・トピック(Help Topics)メニューから「編集メニューを表示(Displaying an EditMenu)」の例示を選択したときに、表示される初期リスト・ウィンドウ900およびテキスト・ウィンドウ904を示す。この例示を実行するために、ユーザは実行(Run)ボタン962をクリックする。この実行(Run)ボタン962は、図26に示すように、編集ウィンドウの基礎を示す例示ウィンドウ964を生成する。同時に、テキスト・ウィンドウ904は、「編集ウィンドウを表示(Displaying an Edit Window)」の例示に対するMAIN()関数に関する情報を表示するために更新される。

## 【0054】

ユーザが例示ウィンドウ964内でGUI制御を選択的に操作すると、対応する方法でテキスト・ウィンドウ904に表示される情報が自動的に更新される。図27では、ユーザは、テキスト・ウィンドウ904にedit1TB::focusIn()に関する情報を表示させるテキスト・ボックス966内を既にクリックしている。同様に、ユーザがテキスト・ボックス

968をクリックすると、テキスト・ウィンドウ904は、図28に示すようなedit2TB::focusIn()に関する情報を表示する。ユーザがチェックボックス(CheckBox)970をクリックすると、テキスト・ウィンドウ904は、図29に示すようにnoneditCB::focusIn()に関する情報を表示する。

#### 【0055】

ユーザは、例示のソース・コードを変更するか、またはそのパラメータを修正し、それらの変更が例示にどのような影響を与えるかを監視することによって、例示を試行することができる。これを実行するために、ユーザは、所望のソース・コード・ファイルを編集し、事前定義された例示を乱さないように、編集したそのソース・コード・ファイルを別個の作業ディレクトリとして保存し、次いでNewEra(商標)開発環境に提供される機構を使用してその例示を再構築する。作成され、実行されることのできるそのような試行の数と種類は、ユーザの想像力によってのみ限定されるものである。

#### 【0056】

例示を実行する際の別のオプションも可能である。例えば、ユーザは、同時に注釈を表示させずに例示を実行することができる。さらに、NewEra(商標)に提供されたデバッガ(Debugger)は、例示を実行する前に例示のソース・コードに区切り点を設定するために使用することができ、それによって、ユーザには例示がどのように機能するかに関するさらなる洞察が与えられる。

#### 【0057】

Informix(商標)byExampleアーキテクチャの説明と、NewEra(商標)開発環境およびInformix(商標)byExampleアプリケーションが構築される方法を、図30～32を参照して提供する。

#### 【0058】

Informix(商標)byExampleは、Windows(商標)95/NTオペレーティング・システムに提供されたオンライン・ヘルプ(OLH)機能を基礎としている。図30に示すように、Informix(商標)byExampleアプリケーション1000は、Informix(商標)byExampleに対して特に作成された資源、ならびにNewEra(商標)開発環境1005に固有の資源を示す。Informix(商標)byExampleアプリケーション1000に特有の構成要素は、対話型例示1007、その例示に対するソース・コード1010、およびその例示を説明する注釈1015を含む。注釈1015は、例示のソース・コードの代表フラグメント1020、例示のソース・コードを表示させるための適切なエディタ(例えば、NewEra(商標)Code WrightまたはNewEra(商標)Window Painter)を起動するショートカット1025、対話型例示へのジャンプ1030、NewEra(商標)オンライン解説1040内に含まれる指定されたキーワードおよびクラス名の説明へのリンク1035を含む複数の異なる副構成要素を有する。

#### 【0059】

図30に示すように、オンライン解説1040、コードライト・エディタ1050およびウィンドウ・ペインタ・エディタ1055は、アプリケーション・ビルダ1060、デバッガ1065およびIPC(Inter Process Communication)ライブラリ1070と共に、開発環境1005の一部として存在し、したがってInformix(商標)byExampleアプリケーション1000とは論理的に分離される。その結果、Informix(商標)byExampleアプリケーション1000のユーザが、キーワードまたはクラス名へのリンク1035をクリックすることによって、またはソース・コードを表示させるためのショートカット1025をクリックすることによって、NewEra(商標)開発環境内に常駐する資源を要求すると、Informix(商標)byExample1000は、要求された資源にアクセスするために、まずインターフェース・ダイナミック・リンク・ライブラリ(DLL)1080を介してNewEra(商標)開発環境1005と対話する必要がある。インターフェースDLL1080は、Informix(商標)byExampleアプリケーション1000に、開発環境の構成要素などの他のアプリケーションと対話することを可能にするルーチンのコンパイルされたライブラリである。Informix(商標)byExample1000は、要求されたオンライン解説情報を表示

するため、または適切なソース・コード・エディタを起動するために、ユーザによって行われた要求の性質に応じて適切なDLLルーチンを呼び出す。

#### 【0060】

より具体的には、Informix(商標)byExampleのユーザが、例示のソース・コード1010の位置へのショートカット1025をクリックすると、Informix(商標)byExampleアプリケーション1000はDLL内の関数を呼び出しが、そのDLL内の関数は、IPCライブラリ1070内の関数を呼び出し、そのIPCライブラリ1070内の関数は適切なエディタを起動する。この関数呼び出し(後述する、Informix(商標)byExampleの構築中にソース・コード・フラグメントを処理することによって自動的に生成される)の一部として、Informix(商標)byExampleアプリケーション1000は、起動されるべきエディタ(コードライト1050またはウィンドウ・ペインタ1055)を指定し、かつ例示のソース・コード1010が指定されたエディタによって開かれるべき行数を識別するパラメータを渡す。Informix(商標)byExampleのユーザがキーワードまたはクラス名へのリンク1025をクリックすると、Informix(商標)byExampleアプリケーション1000はDLL内の関数を呼び出し、その関数は、Windows(商標)OLH機能を使用してオンライン解説1040内の対応する定義を表示させる。

#### 【0061】

インターフェースDLL1080によって提供される他の関数は、対話型例示1007の実行を制御し、リスト・ウィンドウおよびテキスト・ウィンドウの表示が確実に対応を維持するように調整する。インターフェースDLL1080と、Informix(商標)byExampleアプリケーション1000のランタイム動作に関するさらなる詳細を付録Bに記載する。

#### 【0062】

Informix(商標)byExample1000とその構成要素(例えば、例示1007、その例示のソース・コード1010および注釈1015)が生成される方法は、高度のコードの「保全性」を実現する。この保全性とは、アプリケーションを修正することができる効率および容易性の尺度である。高度のコード保全性は、対話型例示と、Informix(商標)byExampleの対応する注釈構成要素を生成するために使用されるすべての情報を、統合された論理エンティティ、すなわち対話型例示自体に対するソース・コードに埋め込むことによって達成される。その結果、情報の唯一の中央ソースだけを維持すればよい。その中央情報に対してなされたいかなる変更または更新も、例示と、その例示に関する資料/説明/ヘルプ機能(Informix(商標)byExample)の両方に自動的に埋め込まれることになる。この自動化された構築手順により、例示と、対応するInformix(商標)byExampleの注釈は、基礎をなすソース・コードに行われた修正の数および回数に関わらず同期して維持されることが保証される。

#### 【0063】

図31に示すように、NewEra(商標)byExampleのソース・コード1100は、物理的には相互依存ファイルの集合で形成されているが、単一の論理エンティティとみなすことができる。ソース・コード1100は、ソース・コード全体で混合されているプログラム説明1105、プログラム・コメント1110、および注釈1115の3つの基本的な種類のテキストを含んでいる。異なるテキストの種類は、プログラミングの慣例により、また、ソース・コード全体を通して様々な異なるマークアップ・シンボル1120を戦略的に配置することによって、相互に区別される。

#### 【0064】

ソース・コード1100内のいくつかのテキストは、複数の目的に役立つことができる。例えば、ソース・コード1100内のプログラム説明1105は、例示の2進数の実行可能ファイル1125内にコンパイルされる。これらのプログラム説明は、例示の実行中の適切なポイントで対応する注釈を表示するためのOLH機能を呼び出すことを含む。エンド・ユーザが例示を実行するとき、これらのOLH呼び出しは、その例示に現時点で何が起こっているかを自動的に説明するためにテキスト・ウィンドウに適切な注釈を表示させる。

### 【 0 0 6 5 】

それらの同じプログラム説明 1 1 0 5 の部分は、また、その例示のソース・コードのクリーン・コピーとしての機能を果たすために抽出され、それは編集環境でユーザに対して表示することができる。同様に、プログラム・コメント 1 1 1 0 としての機能を果たす説明テキスト (Informix (商標) byExample) プロジェクト開発者に方向付けた未処理のプログラミング説明) は、また、注釈 1 1 1 5 (実行時に Informix (商標) byExample のエンド・ユーザに表示されるプログラミング説明) としての機能を果たすことができる。

### 【 0 0 6 6 】

マークアップ・シンボル 1 1 2 0 は、ソース・コード内の様々な種類のテキストを図で示し、対話型例示および Informix (商標) byExample 注釈が構築されるときにそれらがどのように扱われるべきかを指定する。図 3 2 に、「通常の」シンボル 1 2 0 0 および 1 2 0 5 の 2 つのインスタンス、「edit」シンボル 1 2 1 0 および「file」シンボル 1 2 1 5 を含むいくつかのマークアップ・シンボルを含む NewEra (商標) ソース・コードのサンプルを示す。これらのマークアップ・シンボルのそれぞれは、それぞれの引数と共に、それらがコメント・フィールド内に常駐しており、NewEra (商標) プログラム説明として扱われないことを示す一対の括弧 (「{ ... }」) によってバインドされる。NewEra (商標) 以外のプログラミング言語は、コメント・フィールドを図で示すために異なる規則を使用することができる。例えば、Java (商標) プログラミング言語では、コメント・フィールドの開始は「/\*」シンボルによって示され、「\*/」シンボルによって終了する。いずれにせよ、対応するプログラミング言語のコンパイラは、コメント・フィールドに常駐するよう示されたすべてのテキストを無視することになる。

### 【 0 0 6 7 】

「.normal」マークアップ・シンボルは、そのシンボル (例えば、シンボル 1 2 0 0 に続く「Since objects、...」) に続くテキストは説明コメントとして扱われるべきであり、したがって、対応する対話型例示を実行中の適切なポイントで注釈テキストの一部としてテキスト・ウィンドウ内でエンド・ユーザに対して表示されるべきであることを示す。他のマークアップ・シンボルは、出力ファイル名、例示のソース・コードの代表フラグメントとしての機能を果たすべきソース・コードの一部、ジャンプおよびリンクのホットスポットおよび宛先、または表示特性およびオブジェクト (ウィンドウ、ポップアップ、ボタンなど) に関する GUI 関連情報を指定する。マークアップ言語の詳細な説明を付録 C に記載する。

### 【 0 0 6 8 】

一旦ソース・コード 1 1 0 0 が所望通りに修正されると、このコードは、いくつかの異なるステップを通して Informix (商標) byExample アプリケーションの対話型例示および説明的な内容を構築するために使用される。第 1 に、ソース・コード 1 1 0 0 は、PERL スクリプト (テキストを構文解析するためにしばしば使用される汎用インタープリタ型言語である Practical Extraction and Report Language) および WordBasic スクリプトの 2 つの異なるスクリプトによって処理される。スクリプト 1 1 3 0 は、2 つの基本的なタイプの出力を生成する。2 つの基本的なタイプの出力とは、すなわち対話型例示に対するソース・コード・ファイル 1 1 3 5 と、Informix (商標) byExample アプリケーションの説明的かつ視覚的な内容 (例えば、注釈、ソース・コード・フラグメント、ソース・コード・エディタへのショートカット、オンライン解説へのリンク、実行可能な例示へのジャンプ) を表現する RTF ファイル 1 1 4 0 (OLH コンパイラが要求するリッチ・テキスト・フォーマット) である。

### 【 0 0 6 9 】

PERL スクリプトは、マークアップ・シンボルを検索中のソース・コード 1 1 0 0 を構文解析し、遭遇した特定のマークアップ・シンボルに基づいていくつかの RTF ファイル・フラグメントといくつかのソース・コード・ファイル 1 1 3 5 とを作成するが、これらは、ソース・コード 1 1 0 0 全体の様々なサブセットを表現する。次いで WordBasic Script は、RTF ファイル・フラグメントを完全な RTF ファイル 1 1 4 0 にマージし、これは Informix

(商標) byExampleアプリケーションに対する説明的かつ視覚的な内容を含んでいるOLHファイル1150を作成するためにWindows(商標)OLHコンパイラ1145によって処理される。同時に、例示のソース・コード1135は、対話型例示1125に対応する実行可能な2進数を生成するために、NewEra(商標)コンパイラ1155によってコンパイルされる。

#### 【0070】

PERLスクリプトによって生成されたRTFファイル・フラグメントは、ソース・コード1100内に表示される注釈1115の他にいくつかの異なる構成要素を含む。PERLスクリプトは、例示のために抽出されたソース・コードに表示されるキーワードまたはクラス名の各インスタンスを識別する。検出された各キーワードおよびクラス名に関して、PERLスクリプトは、オンライン参考資料内の対応する項目へのリンクをRTFファイル内に作成する。

#### 【0071】

PERLスクリプトは、また、説明的なコメントと共に表示されるテキストとしてRTFファイルに含むために、ソース・コードを表現するフラグメントも抽出する。ソース・コード・フラグメントは、先導し、また後に続く空白行によって図で示されるモノスペースのラップされていないテキストとしてフォーマットされ、説明的なコメントは比例的にスペースの与えられたラップ・テキスト(wrapped text)としてフォーマットされる。RTFファイルに含まれるソース・コード・フラグメントごとに、PERLスクリプトは、また、RTFファイル内に、対応するショートカット・ボタンを挿入する。このショートカット・ボタンによって、エンド・ユーザは、ソース・コード・エディタを起動し、フラグメントが開始する行でソース・コードを表示させることができる。PERLスクリプトは、また、その例示のために抽出されたソース・コードからすべてのマークアップ・シンボル1120を除去する。これにより、関連付けられたエディタで表示させるためのソース・コードのクリーン・バージョンがエンド・ユーザに提供される。

#### 【0072】

PERLスクリプトによって実行される他の関数は、注釈トピックに対する識別子が、対話型例示においてもWindows(商標)OLH機能に含まれるのと同じであることを自動的に保証することを含む。すなわち、PERLスクリプトは、Windows(商標)OLH機能のためのヘルプ・トピック識別子を読み込み、対応するNewEra(商標)定数を生成する。PERLスクリプトは、また、例示を構築するために使用されるNewEra(商標)マークファイル(アプリケーションを構築するためのコンピュータ可読命令を含むファイル)の修正されたバージョンも生成する。PERLスクリプトとその操作のさらなる詳細を付録Bに記載する。

#### 【0073】

上述のPERLおよびWordBasicスクリプトはNewEra(商標)プログラミング言語で書かれたソース・コードで動作するが、他の種類のソース・コード、例えばJava(商標)またはVisualBasic(商標)などを構文解析するために異なるスクリプトを使用することも可能である。一般に、提供されるほぼいかなる種類のプログラミング言語でも処理するために、適切なPERLおよびWordBasicスクリプトを書くことができ、そのプログラミング言語は、ASCIIソース・コード(PERLによって要求される)を使用し、ある種のソース・コードのコメント機構を提供する。Informix(商標)byExample技術の使用を容易にする他のプログラミング言語の属性は、トピック識別子を有するWindows(商標)OLH機能を呼び出すための機構(例示がその注釈を表示することができるための)、開発環境の編集関数を呼び出すための機構(検討中のプログラミング言語が開発環境を提供または要求すると仮定して、注釈がソース・コード・ファイルを開くことができるための)、Windows(商標)OLHフォーマットでのオンライン解説(ソース・コードのキーワードがオンライン解説へのジャンプを有することができるための)を含む。基礎をなすプログラミング言語に1つまたは複数のそれらの他の属性が欠如しても、上述のInformix(商標)byExample機能の多くは実施可能である。

#### 【0074】

PERLスクリプトは、RTF以外のフォーマットでファイルを出力するために修正されることができる。例えば、修正されたPERLスクリプトは、いかなる使用可能なウェブ・ブラウザ（例えば、Netscape Navigator）を使用して表示させることができるハイパーテキスト・マークアップ言語（HTML）を出力することができる。

#### 【0075】

例示によるドキュメンテーションの他の変形形態も可能である。例えば、対話型例示を説明する注釈は、テキスト以外の方法で提示することもできる。必要に応じて、音声、グラフィック・シンボル、ピクチャー、ムービー、または他のいかなる伝達手段をも使用することができる。さらに、実行すべき対話型例示の選択は、ユーザによる指定以外の要因に基づくか、またはユーザによる指定に付加的な要因に基づくことができる。例えば、対話型例示は、基礎となるアプリケーションの実行中のあるポイントで、またはヘルプ・システムのある実行ポイントで自動的に起動することができる。ユーザがキーワード、クラス名、または他のリンクをクリックすると、リンクによって指示されたテキスト解説情報を表示することに加え、またはこれの代わりに例示を自動的に起動することができる。

#### 【0076】

上述の例示によるドキュメンテーションの方法および技術は、ソフトウェア開発システムのユーザを支援することにのみ限定されるものではなく、アプリケーションを任意のコンピュータ・ベースのアプリケーションまたはユーティリティのための一般的なトレーニングおよび教育ツールとみなすことができる。さらに、本明細書に記載の技術は、ハードウェアまたはソフトウェアでも、またはこれらの組み合わせによって実施することができる。この技術は、それぞれがプロセッサ、（揮発性および不揮発性メモリ、または記憶素子、あるいはその両方を含めて）そのプロセッサによって可読の記憶媒体、および適切な入出力装置を含むプログラミング可能なコンピュータ上で実行中のコンピュータ・プログラムによって実施されることが好ましい。プログラム・コードは、記述された関数を実行し、出力情報を生成するために、入力装置を使用して入力されたデータに適用される。この出力情報は、1つまたは複数の出力装置に適用される。

#### 【0077】

各プログラムは、好適には、コンピュータ・システムと対話するために、高レベル手続き型のまたはオブジェクト指向プログラミング言語によって実施される。しかし、プログラムは所望ならばアセンブリまたは機械言語でも実施することができる。いずれにせよ、言語はコンパイル型言語またはインタープリタ型言語のどちらかであってよい。

#### 【0078】

そのような各コンピュータ・プログラムは、記憶媒体または記憶装置（例えば、CD-ROM、ハードディスクまたは磁気ディスクケット）に記憶することが好ましい。尚、これら記憶媒体または記憶装置は、これらが上述の手続きを実行するためにコンピュータによって読み込まれたときに、コンピュータを構成し、また、操作するために、汎用または専用のプログラム可能コンピュータによって可読である。このシステムは、コンピュータ・プログラムによって構成されたコンピュータ可読記憶媒体としても実施することができるが、この場合、そのように構成された記憶媒体はコンピュータを特定の事前定義した方法で動作させる。

#### 【0079】

他の実施形態は頭記の特許請求の範囲に含まれる。

#### 【0080】

例えば、例示によるドキュメンテーションの方法および技術は、ローカル・エリア・ネットワーク（LAN）、インターネット、またはインターネットなどのネットワークまたはウェブ・ベース環境に適用することができる。Informix（商標）byExampleとして知られているInformix（商標）Software社製のウェブ・ベースの説明システムは、開発者が有効かつ効率的にデータベース・アプリケーションを実施することができる容易性を劇的に高める。このウェブ・ベースの説明システムは、ユーザにアプリケーションの実施方法を教示するウェブ・ページ形式の例示および説明を提供する。アプリケーション自体は、ウェ

ブ・ベースのアプリケーションであっても、そうでなくともよい。このウェブ・ベースの説明システムによって、ユーザは、アプリケーション・プログラムの実施の詳細にアクセスするために、フォーマットされたソース・コードを読み、プログラム構造（例えば、ファイル、クラス、関数の階層）をナビゲートすることができる。このウェブ・ベース・システムは、ユーザが言語キーワード（コード・フラグメント形式として示すことができる）から、そのキーワードに関係する完全な参考文書にジャンプすることも可能にする。ユーザは、エディタでソース・コードを開き、ツールバーの実行ボタンのクリックでウェブ・ブラウザから直接的にプログラムを実行することができる。例示が実行されると、現行方法またはプログラム内の関数に関する注釈ページが表示される。したがって、ユーザは、そのようなアプリケーション・プログラムを実施する方法をより迅速に理解するために方法または関数が実行されるときに、注釈を表示させることができる。

#### 【0081】

これは、単純にユーザにアプリケーションの使用方法を示す従来のオンライン・ヘルプ・システムとは対照的である。一部の実施はアプリケーションの動作を変更しないが、すべての実施態様の変更は、定義により、実施を左右するので、これは実施資料の維持をより重要なものにする。ウェブ・ベースの説明とヘルプ・システムは、従来のオンライン・ヘルプ・システムのこの問題点を、ソース・コードに埋め込まれたコメントとして注釈を維持し、ソース・コード・ファイルが変更するときはいつでも注釈ページを自動的に生成することによって解決する。

#### 【0082】

ウェブ・ベースのシステムは、また、ユーザに対して特定タスクを達成するためのハウツー資料を提供する。このハウツー資料は、明瞭なステップ・バイ・ステップの説明を提供し、エキスパートおよび「初心者」に対して柔軟なインターフェースを提供する。このハウツー資料は有用なグラフィックを含み、必須のソフトウェアおよび関連するデモおよび他の技術情報へのリンクを提供する。さらに、このハウツー資料は、目的を達成するためにどの製品を使用すべきかをユーザが決定するために役立つ。

#### 【0083】

図33に示すように、ユーザが最初に、ウェブ・ベースの説明システムを提供するウェブ・ブラウザにアクセスしたとき、ディフォルト・ブラウザ構成1300が表示される。初期ブラウザは、3つの別個の表示フレームのフレームワークを含む。そのフレームワークとは、すなわち、典型的にはウェブ・ページを表示するが、注釈ページ（特定タイプのウェブ・ページ）などの他の関連ページも表示する内容フレーム1305と、他のサービスを提供することに加えてツールバーまたはナビゲーション・バーを表示するフレームワーク・アプレット（例えばJava（商標）アプレット）を含むトップ・フレーム1310と、例えばJava（商標）アプレットを使用して実施することができる目次（TOC）フレーム1315である。

#### 【0084】

ブラウザがbyExample（商標）HTMLページを表示するとき、そのページはJavaScriptを使用して自動的にフレームワークの存在をチェックする。フレームワークが存在する場合、JavaScriptは、トップ・フレーム1310のフレームワーク・アプレットに対して、新しいページの名称およびそれが提供する内容の種類に関して通知する。ページにフレームワークが無い場合、JavaScriptは、そのページに、初期ページ（すなわち、通知が行われるページ）としてフレームワークを開く。

#### 【0085】

図34および35を参照すると、ページ1400の名称に基づいて、目次1315は対応するタイトル1405を自動的に選択する。目次1315は、クライアントで使用可能なJava（商標）ライブラリをテストし、使用可能ならばツリー制御を有する目次階層を自動的に表示する。目次フレーム1315は、トップ・フレーム1310の内容ボタン1410を使用してユーザによってサイズ変更可能かつ除去可能である。例えば、図34では、目次フレーム1315は、ブラウザの約40%を占有しているが、図35では、目次フ

レーム1315は除去されており、したがって不可視である。内容ボタン1410は、除去されたアイコン（例えば、図35）または開かれたアイコン（例えば、図34）を表示することによってTOCの状況を自動的に示す。

#### 【0086】

ユーザは、TOCツリーの対応するサブトピック、例えばサブトピック「データベース・サーバを拡張する(Extending Database Servers)」1415をクリックすることによって、TOCフレーム1315から内容フレーム1305内の他のリンクにアクセスすることができる。それに応答して、図36に示すように、内容フレーム1305は、そのサブトピックに対応する内容を表示し、TOCフレーム1315はその選択されたサブトピック1415を強調表示する。トップ・フレーム1310が、ブラウザがホームページを表示しているかどうかによってその外観（内容ボタンのように）を自動的に変更するホーム・ボタン1420を含むということは、この特定の実施例でも明らかである。

#### 【0087】

別法として、ユーザは、その内容フレーム内の対応するリンク上、例えば、図35に示すサブトピック・リンク「Java Database Connectivity (JDBC)」1425をクリックすることによって、内容フレーム1305内のリンクにアクセスすることができる。これに応答して、図37に示すように、内容フレーム1305は、そのサブトピックを表現するタイトル1430を含めてそのサブトピックに対応する内容を表示する。さらに、TOCフレーム1315は、ユーザに対して内容フレーム1305内にどの内容が表示されるかを示すために、TOCツリー内の選択されたサブトピック1435を自動的に強調表示する。TOCと内容の間のこのような同期は、ユーザがアプリケーションの多くのサブトピックおよびトピックを効果的にナビゲートするために役立つ。

#### 【0088】

再び図36を参照して、ユーザが内容フレーム1305に例示アイコンまたはリンク1440、またはTOCフレーム1315内のサブトピック・アイコンまたはリンク1445のどちらかを使用して例示を選択するとき、ブラウザは、内容フレーム1305内に、図38に示す対応する例示を説明する注釈ページ1500を表示する。さらに図39を参照すると、注釈ページ1500は、例示（散文とも呼ばれる）を説明する注釈1505と、別の注釈ページ1510への少なくとも1つのリンクとを含む。

#### 【0089】

例えば、ユーザがソース・コード・ファイル「querydb.html」へのリンクをクリックするとき、内容フレーム1305は、図40のブラウザに示すようにソース・コード・ファイル注釈ページ「querydb.html」1525を表示する。内容フレームは、図41に示すように、対応するソース・コード・ファイルquerydb.htmlに関するソース・コード・フラグメント1530を表示する。ソース・コード・フラグメント1530は、検討中の例示のソース・コード・ファイルから直接的にインポートされたものである。内容フレーム内のソース・コード・ファイル注釈ページは、トピックが対応するソース・コード・ファイルを説明する注釈1535も表示するが、このような注釈1535は散文と称される場合がある。

#### 【0090】

ソース・コードは、標準HTMLでマークアップされている。注釈コメントがソース・コード・ファイルのソース・コード・コメント内に埋め込まれているので、注釈1535は維持することが容易である。例えば、図42を参照すると、ソース・コード・ファイル1560は、quewrydb.htmlソース・コード1525に関して示される。ソース・コード・フラグメントを含んでいる注釈ページは、マークアップされたソース・コード・ファイルから自動的に生成される。

#### 【0091】

図43の流れ図1600を参照すると、例示の作者は、ソース・コード・コメントに注釈を埋め込むことによってソース・コード・ファイルを注釈する（ステップ1605）。この作者は、また、HTMLタグおよびウェブ・ベース・システムの1つまたは複数の特別な

説明タグ付きの埋め込まれた注釈もマークアップする(ステップ1610)。図44の流れ図1620に示すように、一旦、ソース・コード・ファイルがマークアップされると、PERLなどのスクリプト・プログラミング言語は、プログラミング言語からプログラム構造を判定するためにソース・コード・ファイルを構文解析し、注釈と特別な説明タグとを読み込む(ステップ1625)。PERLスクリプトは、ソース・コード・ファイルのプログラム構造を反映する注釈ページを生成する(ステップ1630)。さらに、PERLスクリプトは、プログラム構造をナビゲートするための注釈ページ間にリンクを提供し(ステップ1635)、フレームワークを通知するためにJavaScriptを使用する(ステップ1645)一方で、注釈コメントを組み込み(ステップ1640)、関連情報へのリンクとしてソース・コード内の言語キーワードを強調表示する(ステップ1650)。したがって、注釈は容易に維持される。図43を再度参照すると、例示の作者は、ソース・コードと注釈を同時に同じ場所で調整または編集することができる(ステップ1615)。したがって、注釈は自動的に再生成される。

#### 【0092】

内容フレーム内の注釈ページは、また、関連情報へのリンク、またはある注釈ページから別の注釈ページへの1つまたは複数のリンクを含むことができる。このような柔軟性は、PERLスクリプトがプログラム構造を決定し、プログラムの別の部分への参照をリンクとしてフォーマットするので、上記の注釈ページの自動生成によるものである。例えば、ユーザがquerydb.htmlファイル1525内の「sportsHeader.tag」リンクを選択するとき、内容フレーム1305は、図46に示すように、sportsHeader.tagソース・コード注釈ページ1540を表示し、TOCフレーム1315は対応するサブトピック・リンク1700を自動的に強調表示する。

#### 【0093】

内容フレーム注釈ページの選択可能なリンクは、バックグラウンド参考資料に対応する。例えば、「MIVAR」リンクが選択されるとき、図47に示すように、MIVARタグに関する情報を表示する別個のブラウザ1705が作成される。この場合、PERLスクリプトは、ソース・コード内のキーワードを発見し、それらを参考文書へのリンクとしてフォーマットする。ユーザが内容ウィンドウ内の「ウェブ・ホーム(WEB HOME)」リンクを選択すると、図48に示すように、用語集を含むブラウザ1710が作成される。

#### 【0094】

再度図40を参照すると、ユーザは、図49に示すように、内容フレーム1305内の注釈ページに対応するソース・ファイルをブラウザ・ウィンドウ1715内の平文テキストとして開くために、トップ・フレーム1310内の編集ボタン1562を選択することができる。この場合、PERLスクリプトはソース・コード・ファイルから注釈コメントを自動的に除去し、その結果、ユーザは長文説明が挿入されていないソース・コード・ファイルを編集することができる。次いでユーザは、このウィンドウと、ユーザ固有のプログラムを表示する別のテキスト・ウィンドウとの間でテキストをカット・アンド・ペーストすることができる。

#### 【0095】

再度図38を参照すると、Java(商標)例示アプレットまたはアプリケーションを起動したり、Java(商標)アプレットまたは他の埋め込まれたプログラム・オブジェクトを含んでいるHTMLページをダウンロードしたり、出力をブラウザ・ウィンドウに宛先変更して非対話型プログラムを実行するように、ウェブ・サーバに共通ゲートウェイ・インターフェース(CGI)要求を送信するために、ユーザはフレームワーク・アプレットを使用して例示を実行することができる。Java(商標)例示の場合、ウェブ・ブラウザに注釈ページを表示するよう要求するために、実行中の例示はJava(商標)アプリケーション・プログラミング・インターフェース(API)を呼び出す。CGIプログラムの場合、ウェブ・ブラウザに注釈ページを表示するよう要求するために、例示プログラムは、JavaScriptを含んでいるHTMLページを戻す。したがって、例示は、サーバ上、またはウェブ・ブラウザの制限付きサンドボックス内で実行される。ユーザが例示をローカルに実行する許可を要求する

ためにフレームワークがブラウザのセキュリティ機能を使用する場合、例示は、ローカル・クライアント上で実行することができる。

#### 【0096】

図38において、ユーザは実行( Run )ボタン1570を選択することによって例示を実行することができる。図45の流れ図1655を参照すると、実行( Run )ボタン1570を選択すると、図50に示すように、上記の任意の方法を使用して、ブラウザ・ウィンドウ1800内で例示が起動する(ステップ1660)。TOCフレーム1315は、例示を実施するソース・コードに対する現行注釈ページを強調表示することによって、同時に自動的に実行中の例示と同期する。PERLスクリプトは、注釈ページの自動生成によく似たように、階層構造を示すリンクに関してHTMLページを構文解析することによってTOCを自動的に生成する。

#### 【0097】

さらに、例示は、現行注釈ページ、この例では上記のquerydb.htmlファイル1525を表示することによって、内容フレーム1305を実行中の例示と同時に自動的に同期する。次いでユーザは、例示が実行されている間に、注釈付きのソース・コードを表示させるか、それともそのソース・コード・ファイルを編集することができ、上記のようにキーワードからそのキーワードに対する参考文書にジャンプすることができる。

#### 【0098】

ユーザがブラウザ・ウィンドウ1800内の例示と対話するとき(ステップ1665)、(例えば、顧客番号を入力し、ブラウザ・ウィンドウ1800内の「サブミット(Submit)」ボタンを選択することによって)、ユーザ・インターフェースは、図51に示すようにブラウザ・ウィンドウ1800内で変化する(例えば、入力された顧客番号に対応するカスタマー・レポート)(ステップ1670)。ユーザ・インターフェースは、実行中の例示で使用される次の注釈ページcust\_db.htmlファイル1586に対応して変化する。実行中の例示は、注釈を表示すべきプログラム・ユニットの例示、ファイル、クラスおよび関数の名称を有する注釈要求モジュールを自動的かつ同時に呼び出す(ステップ1675)。注釈要求モジュールは、その要求を注釈ページにマップし、内容フレーム内に注釈ページを表示するようブラウザに伝える(ステップ1680)。例えば、図51では、次の注釈ページはcust\_db.html1586である。Cust\_db.htmlサブトピック1585は、TOCフレーム1315で強調表示され、図51に示すように、内容フレーム1305はcust\_db.html注釈ページ1586を表示する。埋め込まれた注釈マークアップ1810は、図52に表示されるcut\_db.htmlソース・コード・ファイル内に示される。埋め込まれたマークアップ1810は、ソース・コード・ファイルから注釈ページを生成するときに使用される。

#### 【0099】

再度図36を参照すると、例示を選択することに加え、ユーザは、アイコンまたはリンク1450を内容フレーム1305内に文書化する方法、またはサブトピック・リンク1455をTOCフレーム1315内に文書化する方法を選択することもできる。この場合、図53に示すように、文書化する方法1900が内容フレーム1305に表示される。図を見やすくするために、完全な文書1900は図54に示す。同期において、TOCフレーム1315は、文書化する方法1900に対応する表示されたサブトピック1902を強調表示する。

#### 【0100】

さらに同期において、選択された文書化する方法1900を反映するために、トップ・フレーム1310は動的に変化する。例えば、トップ・フレームには、現時点でアップ・ボタン1905、第1のページ・ボタン1910、前ページ・ボタン1915、次ページ・ボタン1920、最終ページ・ボタン1925、およびリスト・ボタン1930が表示されている。アップ・ボタン1905は、選択されると、文書化する方法1900を列挙するカテゴリまたはサブトピックにジャンプする。例えば、サブトピック「データベース・サーバを拡張する(Extending a Database Server)」は、「ウェブアプリケーション

を生成する (Create Web application)」の文書化する方法 1900 を列挙し、アップ・ボタン 1905 を選択すると、図 36 に示すように、ブラウザは「Extending a Database Server」サブトピックに関するすべての情報を表示する。一般に、トップ・フレームは、新しい内容のページが表示された旨をフレームワークに通知する JavaScript 呼び出しで指定される内容の種類と一致するように変化する。

#### 【0101】

文書 1900 は他の文書またはページへのポインタを含んでおり、このポインタは複数の方法の 1 つによってアクセスされる。ポインタは、例えば異なる色、フォント、またはスタイルによって示される内容フレーム内のリンク上をクリックすることによって、内容フレーム 1305 から直接的に選択することができる。例えば、ユーザは図 54 に示すように内容フレーム 1305 内のアプリケーション・ページ (Application Page)リンク 1930 またはデータベースを準備する (Prepare Database)リンク 1935 上をクリックすることができる。ウェブアプリケーションを生成する (Create Web application)文書 1902 のサブトピックを選択することによって、TOC フレーム 1315 からポインタを選択することができる。例えば、サブトピックデータベースを用意する (Prepare Database)1940 はリンクデータベースを用意する (Prepare Database)1935 に対応し、それらのポインタのどちらを選択してもユーザは同じ文書に移動される。ポインタを選択する際、そのポインタに対応する文書またはページが次いでユーザに表示される。

#### 【0102】

さらに図 55 を参照すると、ユーザがトップ・フレーム内のリスト・ボタン 1930 を選択すると、ローカル目次ウィンドウ 1320 が表示される。例えば、ユーザが図 53 のリスト・ボタン 1930 を選択すると、図 55 に示すようにウィンドウ 1320 が作成される。文書をナビゲートする従来型モデルは階層型目次付きである。TOC モデルは、常時ユーザを方向付けるが、内容を制限する。これは階層には適合しない場合がある。文書をナビゲートするより新しいモデルがハイパーテキスト・ウェブである。このウェブ・モデルは文書内のいかなるトピックでも、同じトピック上で拡張する、より詳細な文書にリンクする。このウェブ・モデルは人為的な制限を除去するが、あっという間にユーザを混乱させる。

#### 【0103】

これとは対照的に、Example (商標) のウェブ・バージョンは、リンクの非構造化ウェブ・ページのローカル・コーナーで構造化された階層図を提供するローカル目次モデルを使用する。このモデルでは、文書内のいくつかのページは、ローカル TOC へのルート・ページである。例えば、ウェブアプリケーションを生成する (Create Web application)文書 1900 に対応するページは、ローカル TOC へのルート・ページである。ユーザが、ルート・ページ (固有である) などのローカル TOC 固有の任意のページにナビゲートするとき、フレームワーク・アプレットはそのルート・ページに対するローカル TOC を読み込む。図 56 に示すように、ユーザがローカル TOC 1320 の紹介 (Introduction) ページにナビゲートするとき、フレームワーク・アプレットはそのルート・ページに対するローカル TOC を読み込み、対応する情報を TOC フレーム 1315 および内容フレーム 1305 に表示する。ローカル TOC の各ページはローカル TOC 内では固有である。ユーザはローカル TOC 内のページを連続的に一歩ずつ進むことができ、ローカル TOC の順番で複数のページを参照したり、選択した現行ページでローカル TOC 階層を表示させたりすることができる。したがって、ユーザは、ローカル・トピック内で方向付けられる。

#### 【0104】

ローカル TOC は、ローカル TOC 内のページのリンクを制約しない。すなわち、ローカル TOC 内のページは、ローカル TOC 1320 には存在しないページへのリンクを有することができる。例えば、ユーザが紹介 (Introduction) 文書内のInformix ウェブ統合オプション (Informix Web Integration Option)リンク 1942 を選択するとき、ブラウザは、図 57 に示すように、ローカル TOC 1320 には存在しないInformix ウェブ統合オプション (Informix Web Integration Option)ページ 1945 を開く。

## 【0105】

さらに、ページは複数のローカルTOC内に表示させることができる。例えば、図56のローカルTOC内にある「サブスペースを作成する(Create subspace)」という題名のページ1950は、別の文書に関する別のローカルTOC内に表示することができる。ルート・ページ（この例では、「ウェブアプリケーションを作成する(Create WEB application)」ページ1900）は單一ローカルTOCに制約される。

## 【0106】

ユーザは、ローカルTOC内の関心のあるトピックを選択することによって、ローカルTOC ウィンドウから直接的にローカルTOC全体をナビゲートすることができる。さらに、ユーザは、第1のページ・ボタン1910、前ページ・ボタン1915、次ページ・ボタン1920、または最終ページ・ボタン1925を使用して、トップ・フレーム1310を介してローカルTOC全体をナビゲートすることができる。これらのボタンは、基本的に、ローカルTOC中を移動することによって、ユーザのために文書内のページをめくる。例えば、ウェブアプリケーションを作成する(Create Web application)文書1900内にあるときにユーザが次ページ・ボタン1920を選択するとき、ブラウザは、ウェブアプリケーションを作成する(Create Web application)文書内の次の文書を表示する。これは、図58に示すように「目次(Table of Contents)」文書1955である。

## 【0107】

図59を参照すると、ローカルTOC2005が構築される方法のモデル2000が示される。ローカルTOCの構築は、ページ2010の非構築（またはランダムな）ウェブ・上に順序付き表示を強制し、このようにして、ウェブ・のページ2010を表示することを容易にする。

## 【0108】

他の実施形態は、頭記の特許請求の範囲に含まれる。

## 【0109】

ウェブ・ベースの説明システムは、Java（商標）、VisualBasic（商標）、C、C++、HTML、Perl、JavaScript、SQL、Informix Stored Procedure Language（SPL）、ESQL/C（Embedded SQL for C）、SQLJ、JSP、ASP、およびInformix Web Data Blade Module言語をサポートすることができる。

## 【図面の簡単な説明】

## 【図1】

従来技術のコンピュータ・システムのブロック図である。

## 【図2】

図1のコンピュータ・システムで使用されるグラフィカル・ユーザ・インターフェース内の表示領域を示す。

## 【図3】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図4】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図5】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図6】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図7】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図8】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図9】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

## 【図10】

従来技術のオンライン・ヘルプおよび文書システムの画面の一場面である。

【図11】

ブラウザ・アプリケーションの画面の一場面である。

【図12】

ブラウザ・アプリケーションの表示領域を示す。

【図13】

Informix(商標) byExampleアプリケーションのユーザに使用可能なオプションを示す流れ図である。

【図14】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図15】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図16】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図17】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図18】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図19】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図20】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図21】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図22】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図23】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図24】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図25】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図26】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

【図27】

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

**【図28】**

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

**【図29】**

Informix(商標) byExampleアプリケーションおよびNewEra(商標)開発環境からの画面の一場面である。

**【図30】**

NewEra(商標)アーキテクチャのブロック図である。

**【図31】**

Informix(商標) byExampleアプリケーションがどのように構築されるかを示すブロック図である。

**【図32】**

NewEra(商標)ソース・コードのサンプルである。

**【図33】**

ウェブ・ベースの指示システムにおける表示領域である。

**【図34】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図35】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図36】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図37】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図38】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図39】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションの表示領域内の内容を示す図である。

**【図40】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

**【図41】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションの表示領域内の内容を示す図である。

**【図42】**

注釈を埋め込んだソース・コード・ファイルである。

**【図43】**

注釈ページの作成者がとるステップを示す流れ図である。

**【図44】**

注釈ページでリンクを自動的に生成するためのスクリプト・プログラムがとるステップを示す流れ図である。

**【図45】**

例示的プログラムがとるステップを示す流れ図である。

**【図46】**

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面

である。

【図47】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図48】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図49】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図50】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図51】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図52】

注釈を埋め込んだソース・コード・ファイルである。

【図53】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図54】

Informix(商標) byExampleのウェブ・ベースのアプリケーションの表示領域内の内容を示す図である。

【図55】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図56】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図57】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図58】

Informix(商標) byExampleのウェブ・ベースのアプリケーションからの画面の一場面である。

【図59】

ウェブ・ベースのアプリケーションで使用されるローカル目次モデルを示すブロック図である。

【誤訳訂正2】

【訂正対象書類名】図面

【訂正対象項目名】図3

【訂正方法】変更

【訂正の内容】

## 【図3】

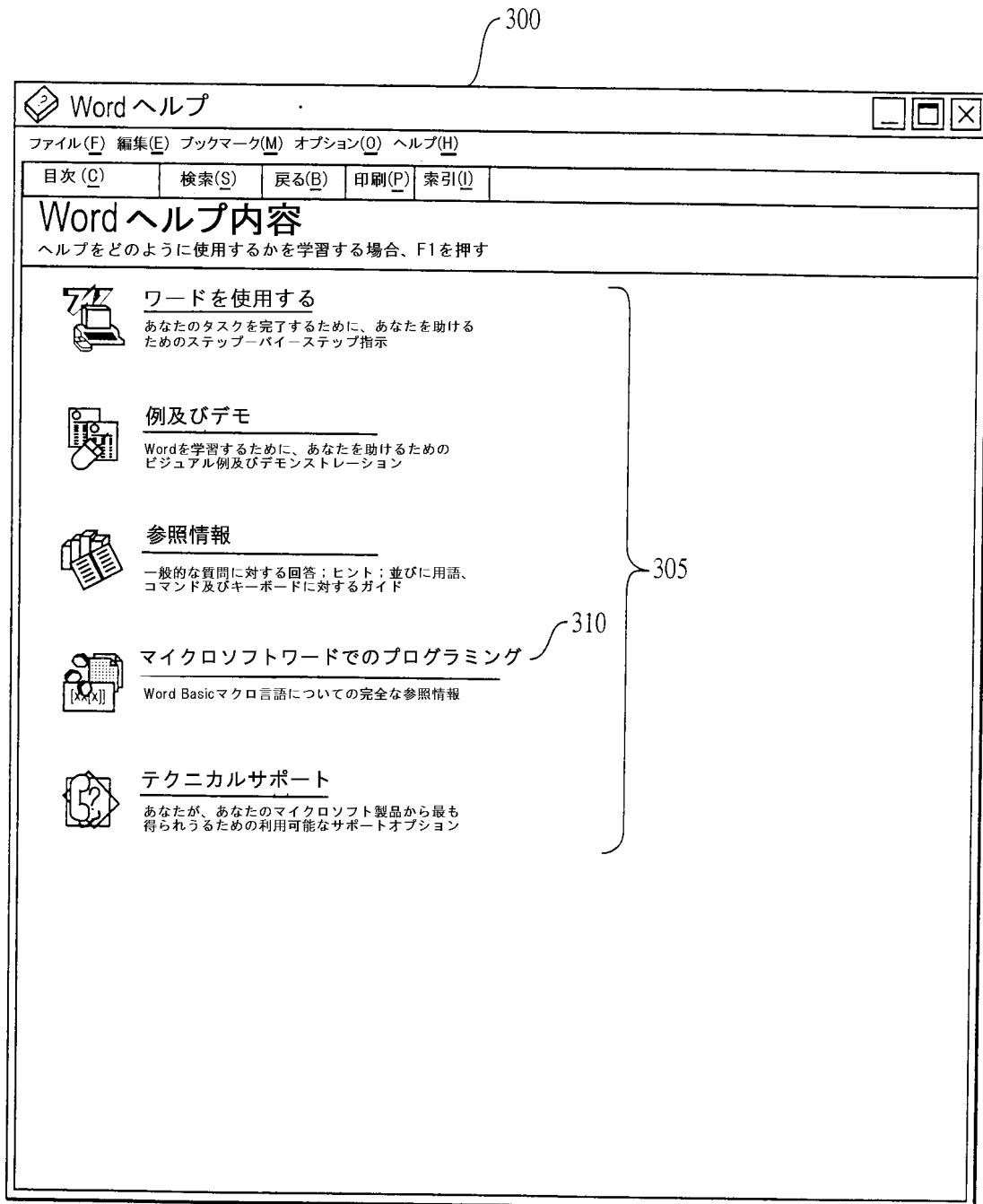


図3  
従来技術

## 【誤訳訂正3】

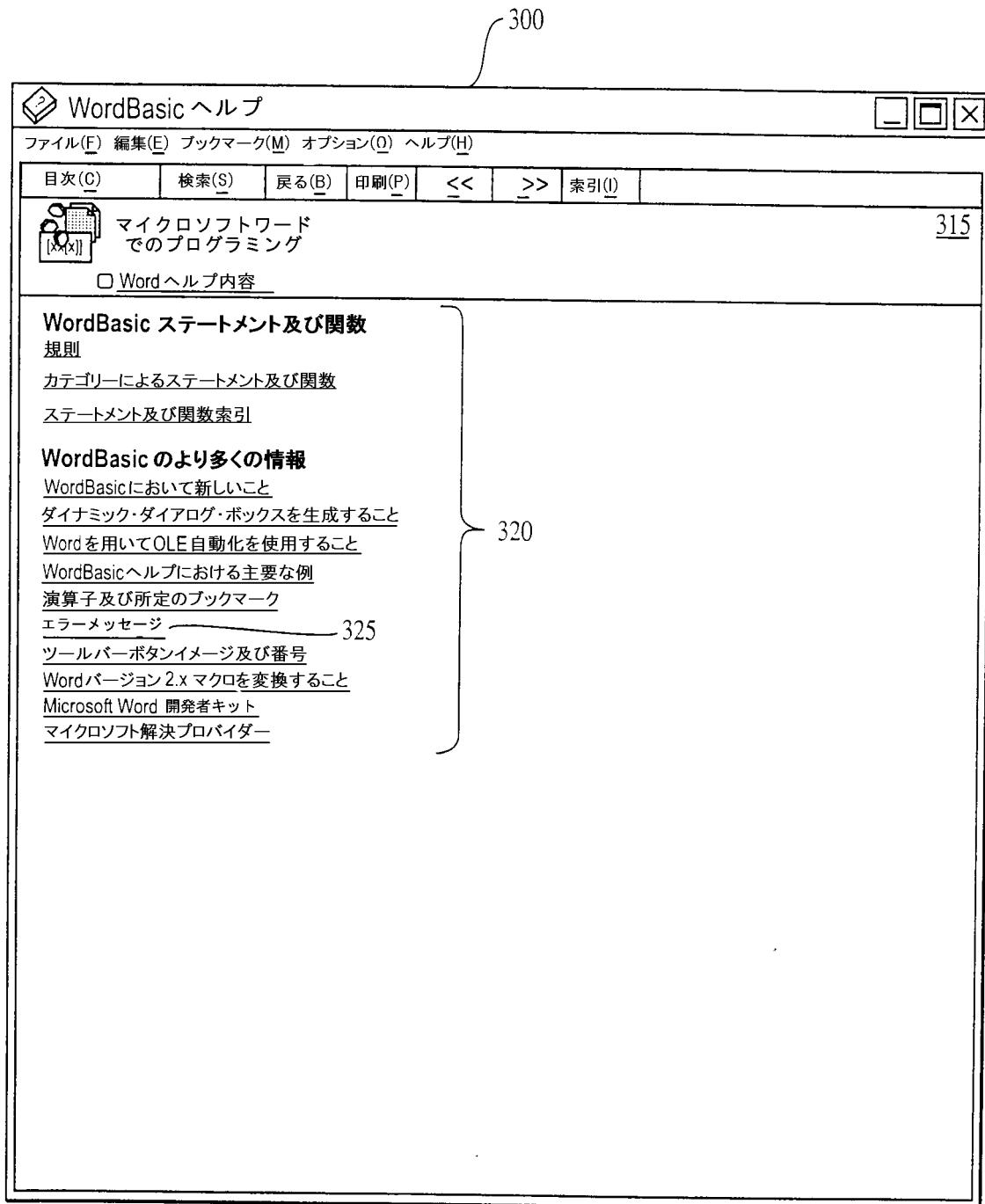
【訂正対象書類名】図面

【訂正対象項目名】図4

【訂正方法】変更

【訂正の内容】

【図4】

図4  
従来技術

【誤訳訂正4】

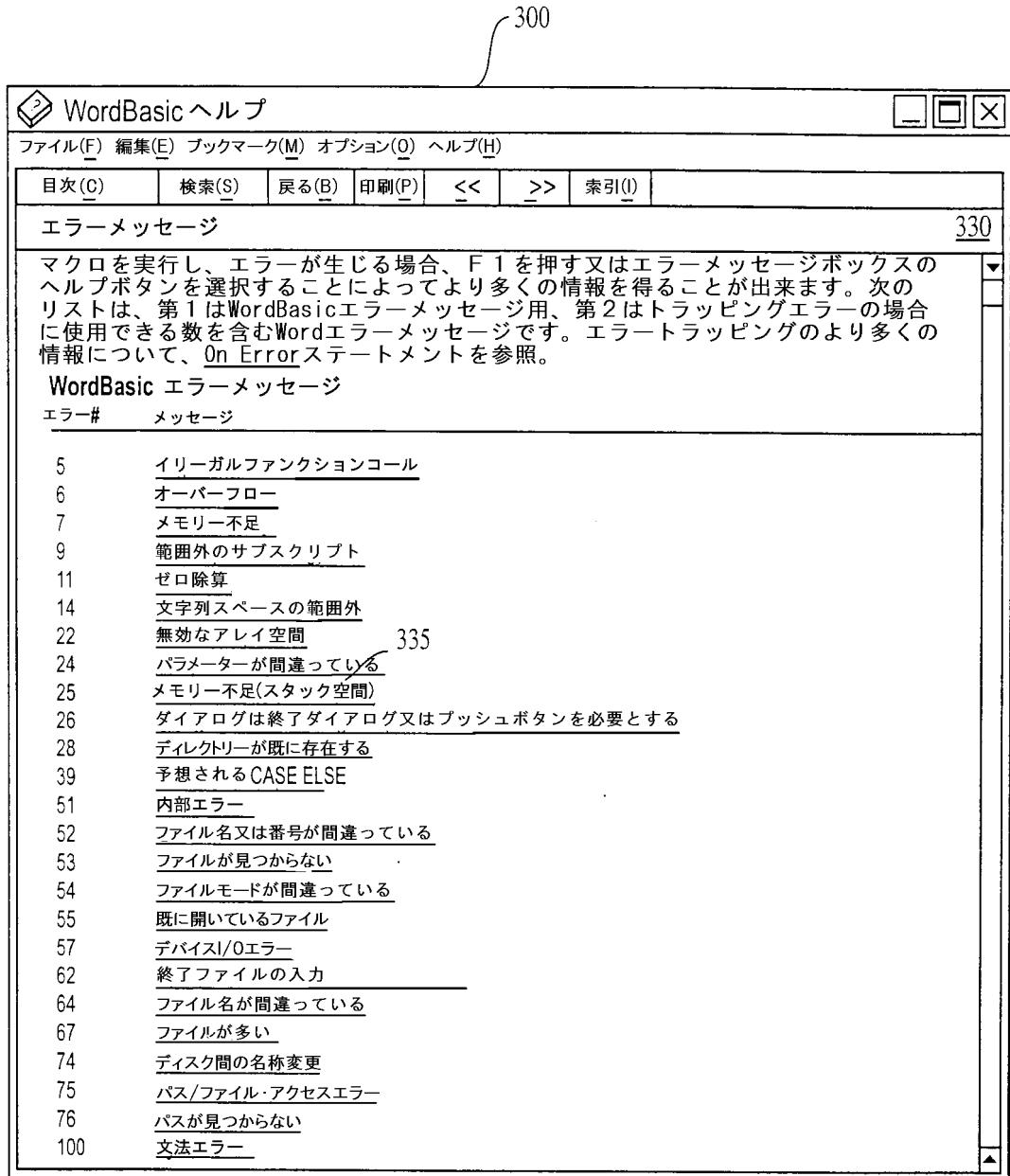
【訂正対象書類名】図面

【訂正対象項目名】図5

【訂正方法】変更

【訂正の内容】

【図5】

図5  
従来技術

【誤訛訂正5】

【訂正対象書類名】図面

【訂正対象項目名】図6

【訂正方法】変更

【訂正の内容】

【図6】

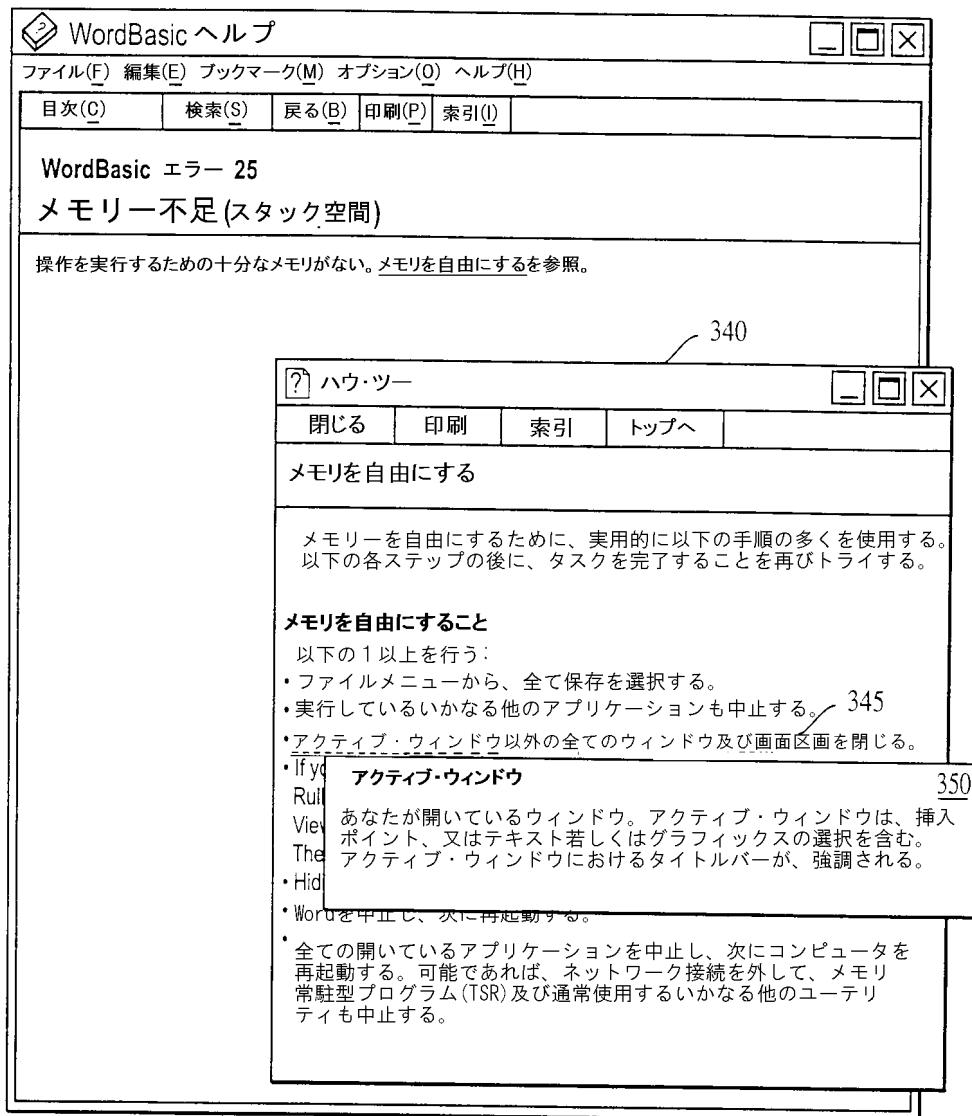


図6  
従来技術

【誤訳訂正6】

【訂正対象書類名】図面

【訂正対象項目名】図7

【訂正方法】変更

【訂正の内容】

【図7】

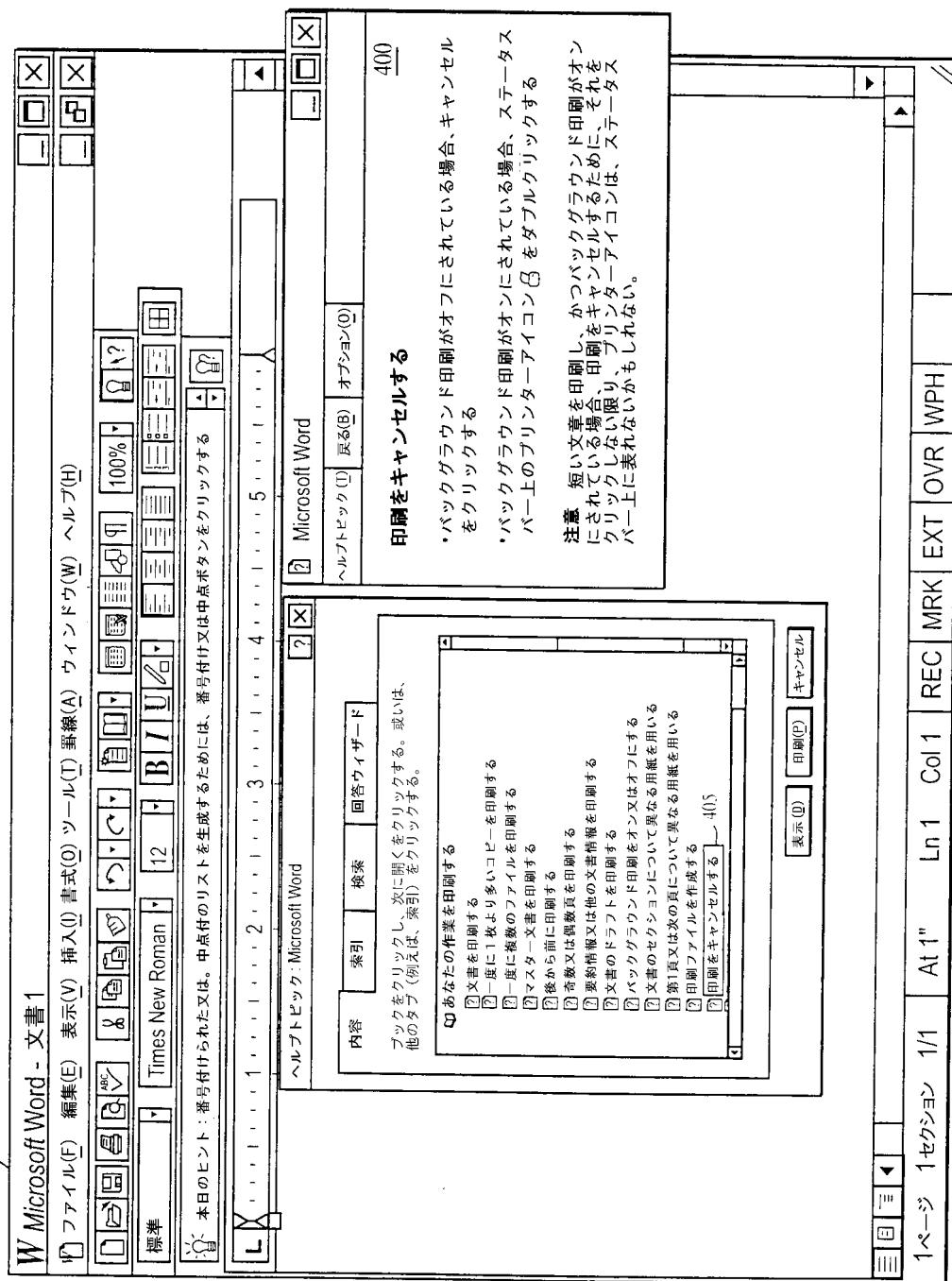


図7 従来技術

【誤訳訂正7】

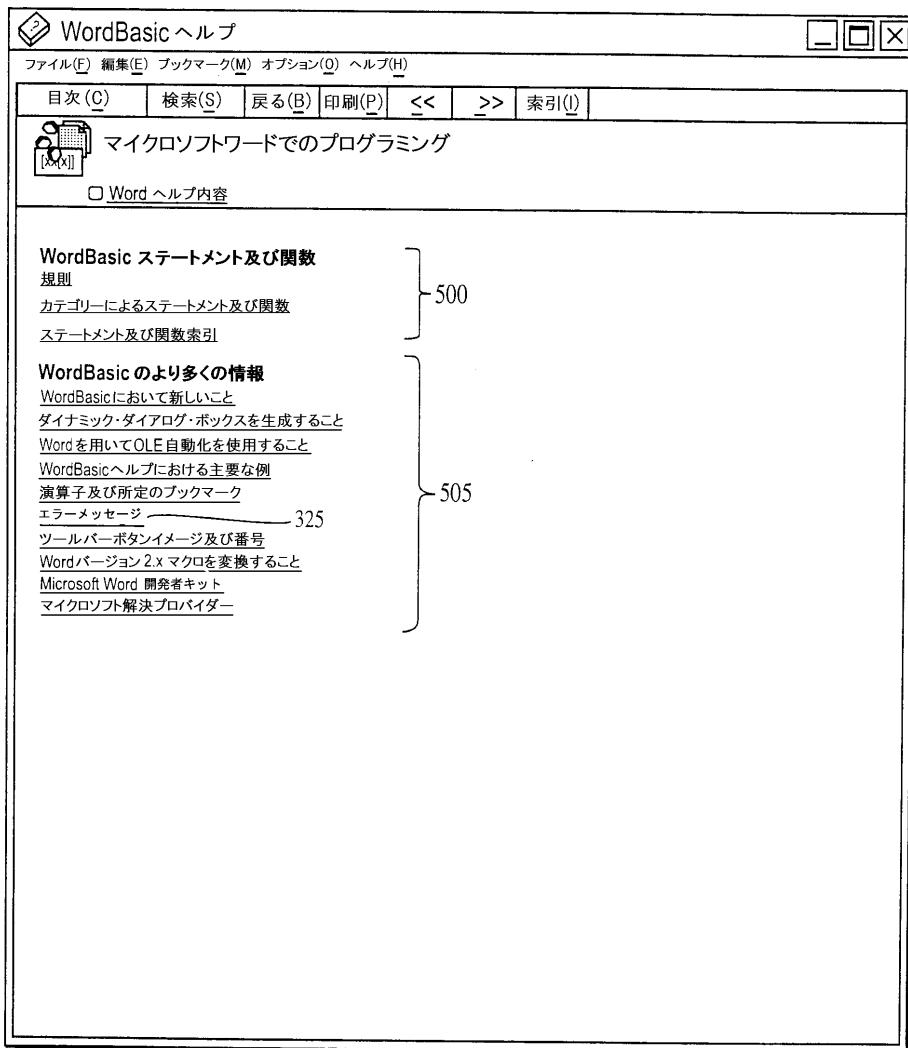
【訂正対象書類名】図面

【訂正対象項目名】図8

【訂正方法】変更

【訂正の内容】

【図8】

図8  
従来技術

【誤訳訂正8】

【訂正対象書類名】図面

【訂正対象項目名】図9

【訂正方法】変更

【訂正の内容】

## 【図9】

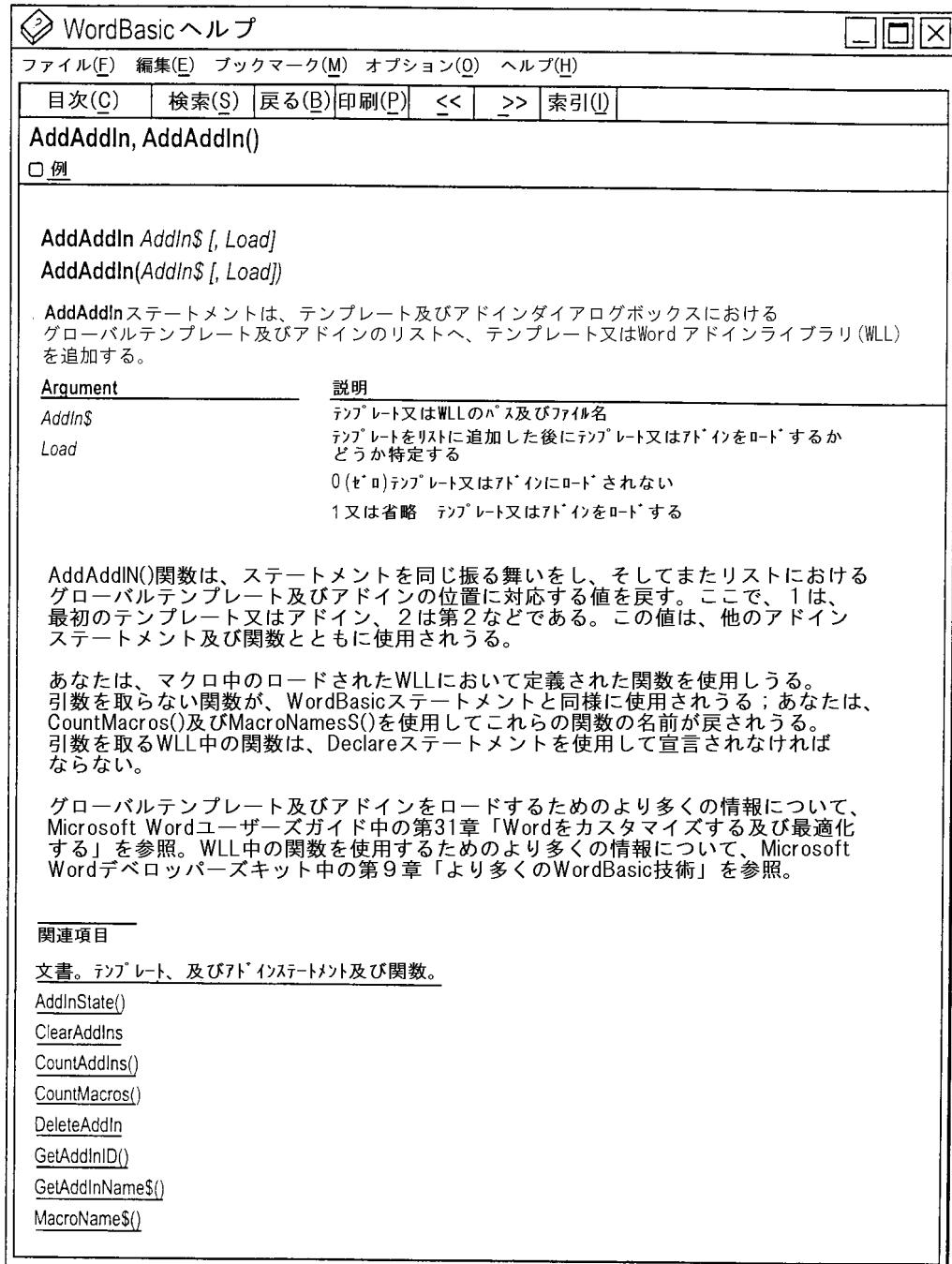


図9  
従来技術

## 【誤訳訂正9】

【訂正対象書類名】図面

【訂正対象項目名】図10

【訂正方法】変更

【訂正の内容】

## 【図 10】

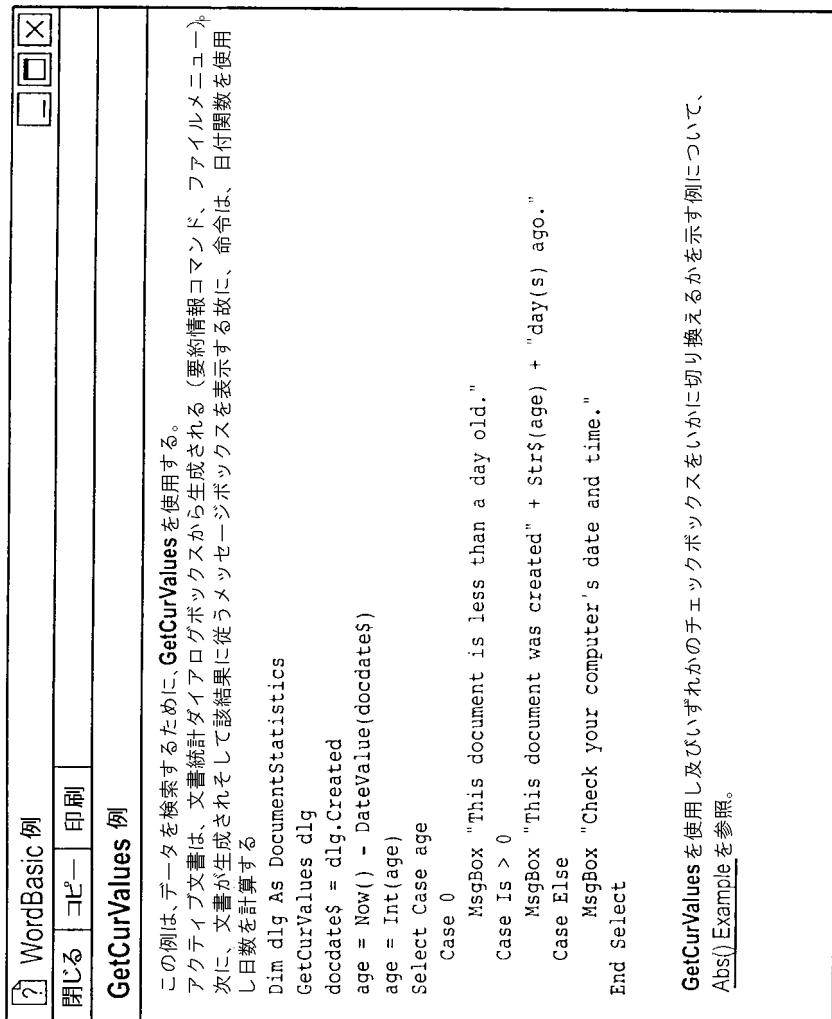


図10  
従来技術

## 【誤訳訂正 10】

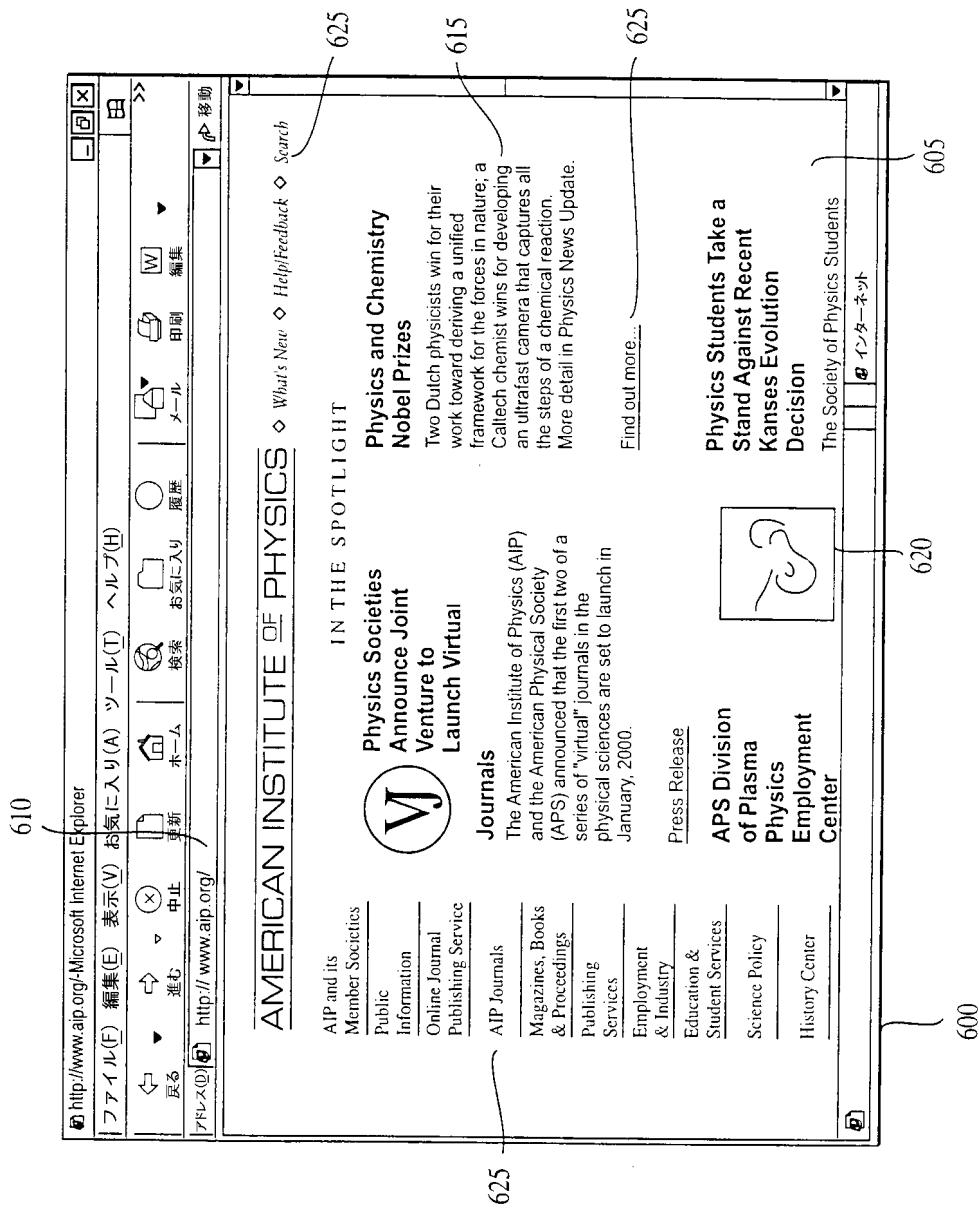
【訂正対象書類名】図面

【訂正対象項目名】図 1 1

【訂正方法】変更

【訂正の内容】

【 図 1 1 】



## 【誤訳訂正 1 1】

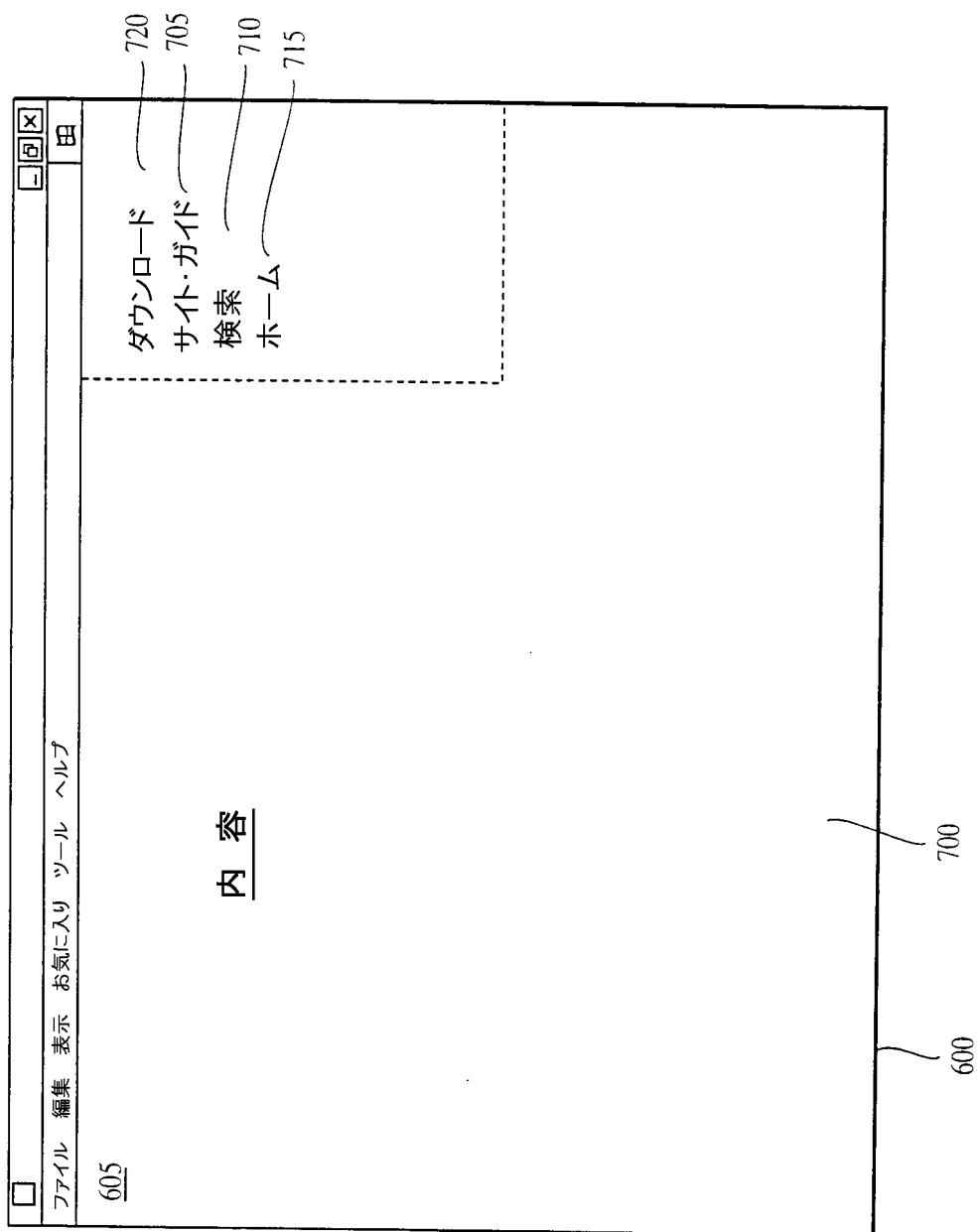
### 【訂正対象書類名】図面

## 【訂正対象項目名】図 1 2

## 【訂正方法】変更

### 【訂正の内容】

【図 1 2】



【誤訳訂正 1 2】

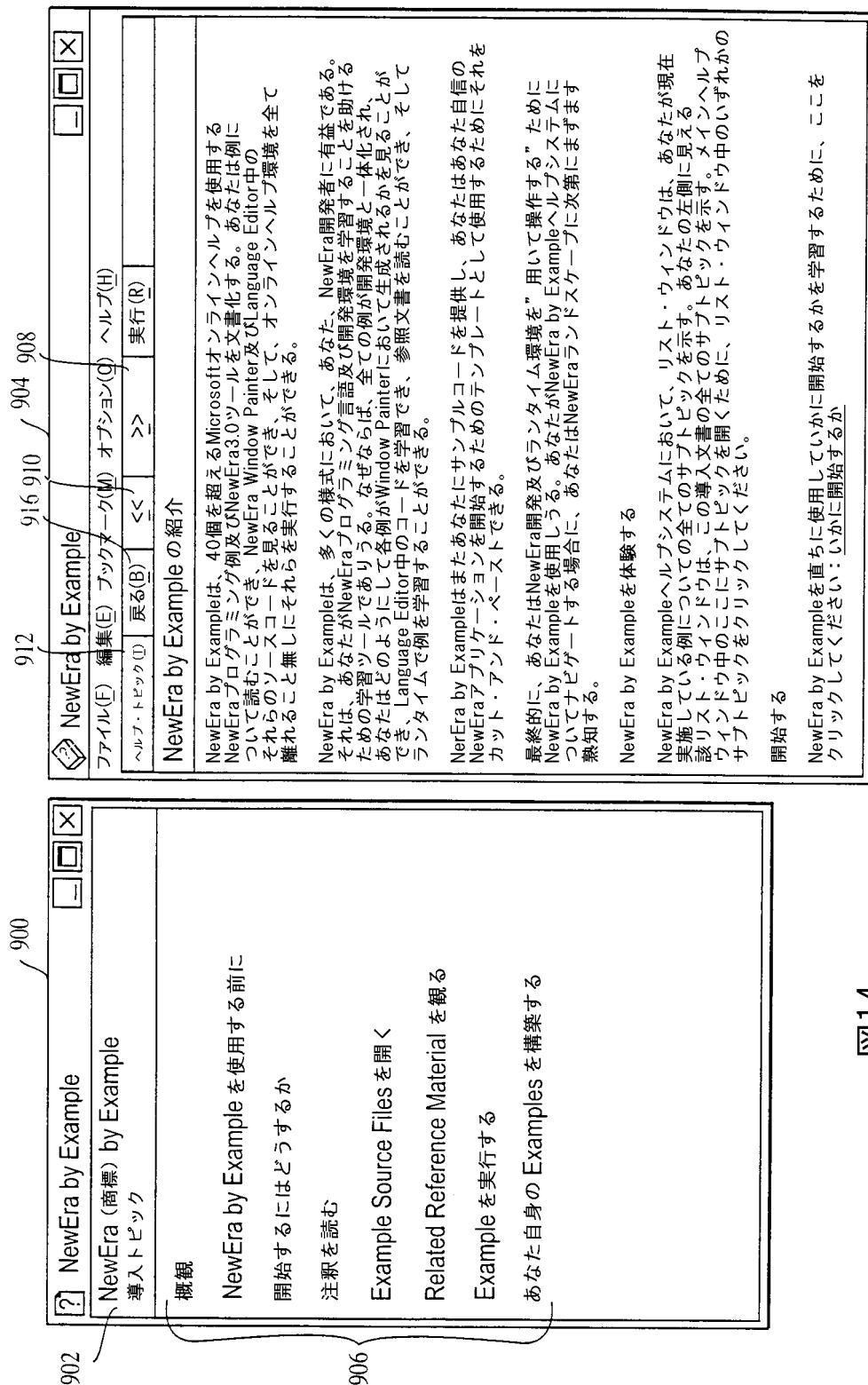
【訂正対象書類名】図面

【訂正対象項目名】図 1 4

【訂正方法】変更

【訂正の内容】

【 図 1 4 】



### 【誤訳訂正 1 3】

### 【訂正対象書類名】 図面

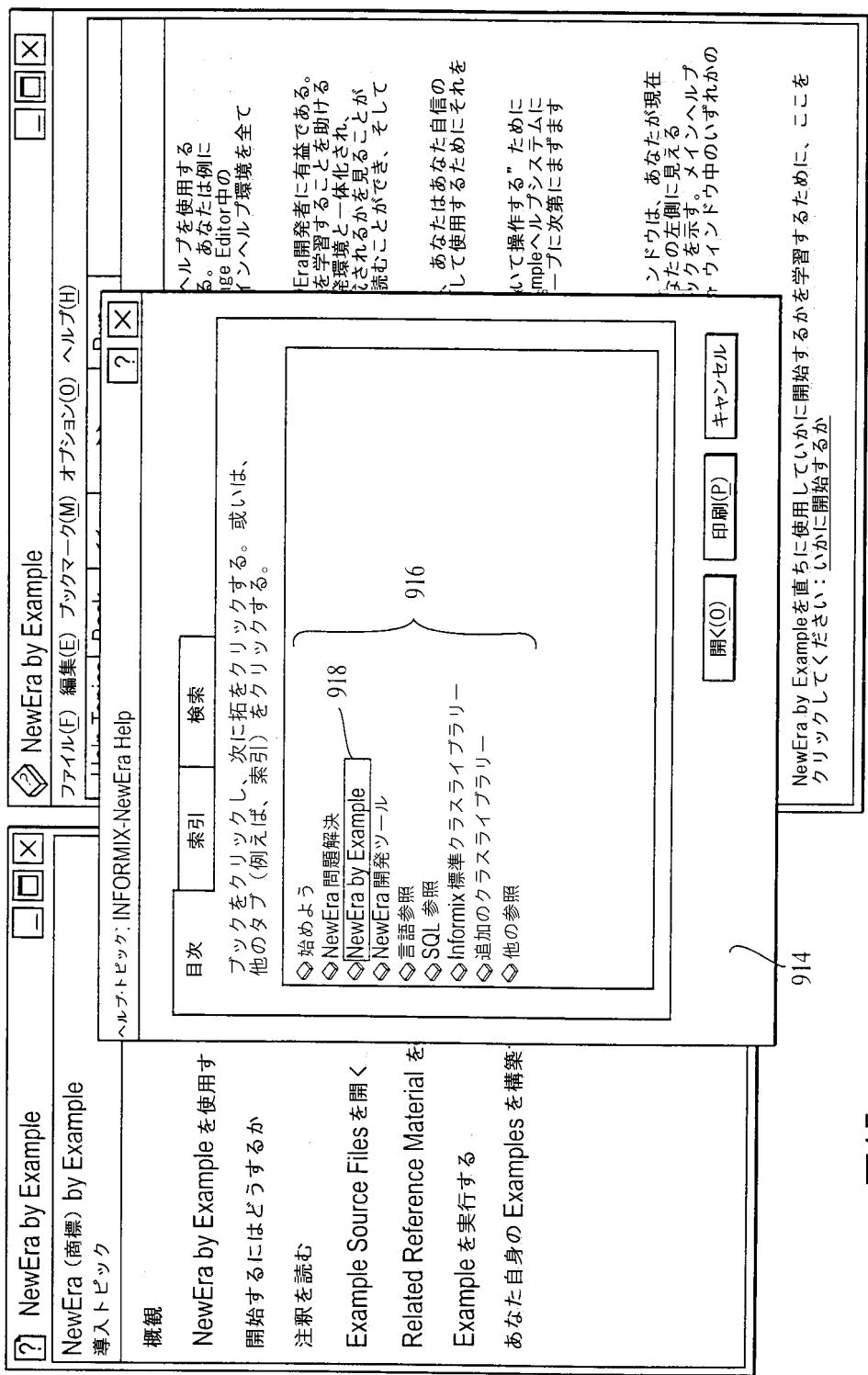
### 【訂正対象項目名】図 1 5

### 【訂正方法】変更

## 【訂正の内容】

14

【 図 1 5 】



15

## 【誤訳訂正 1 4】

### 【訂正対象書類名】図面

## 【訂正対象項目名】図 1 6

## 【訂正方法】変更

## 【訂正の内容】

【図16】

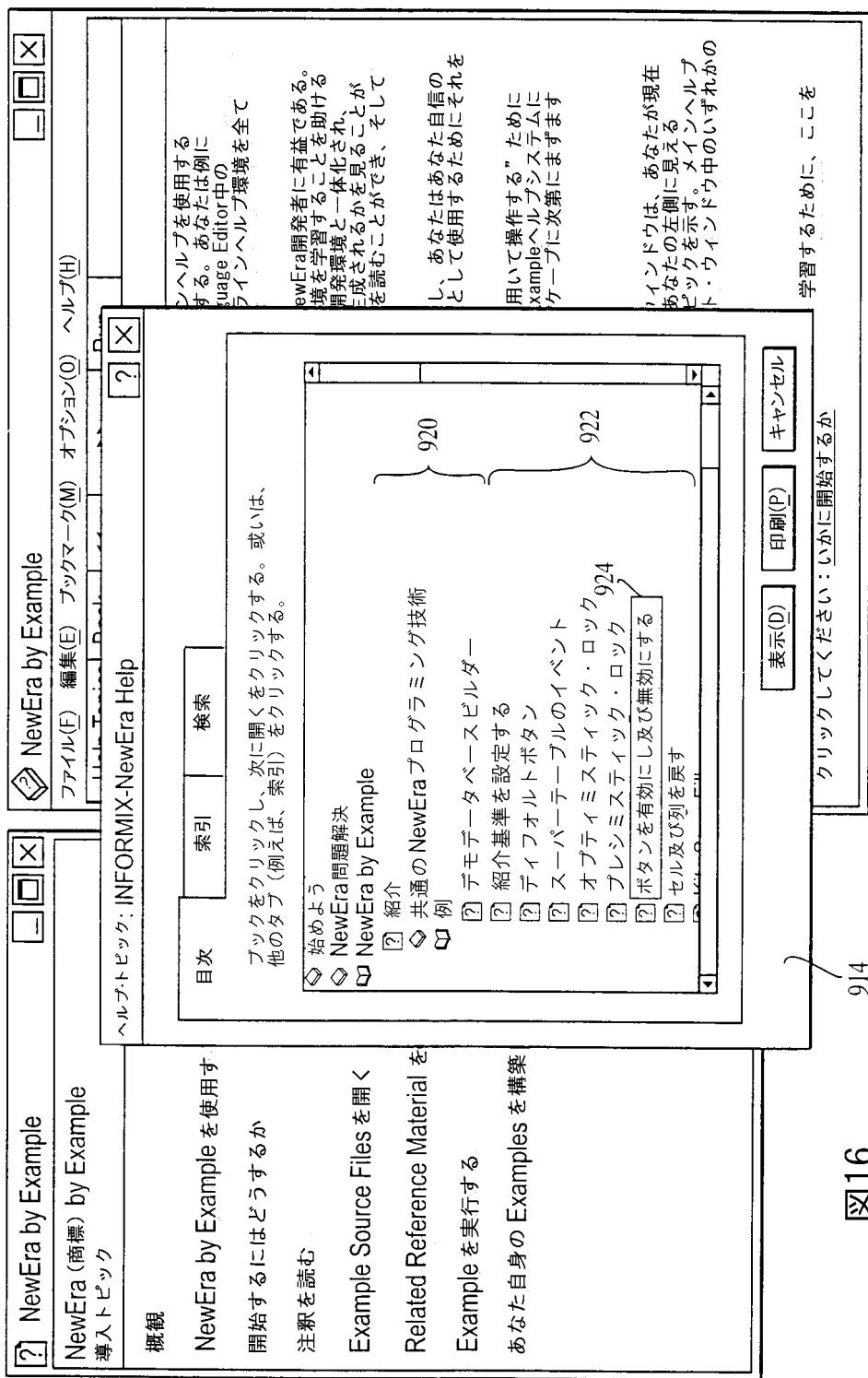


図16

【図17】

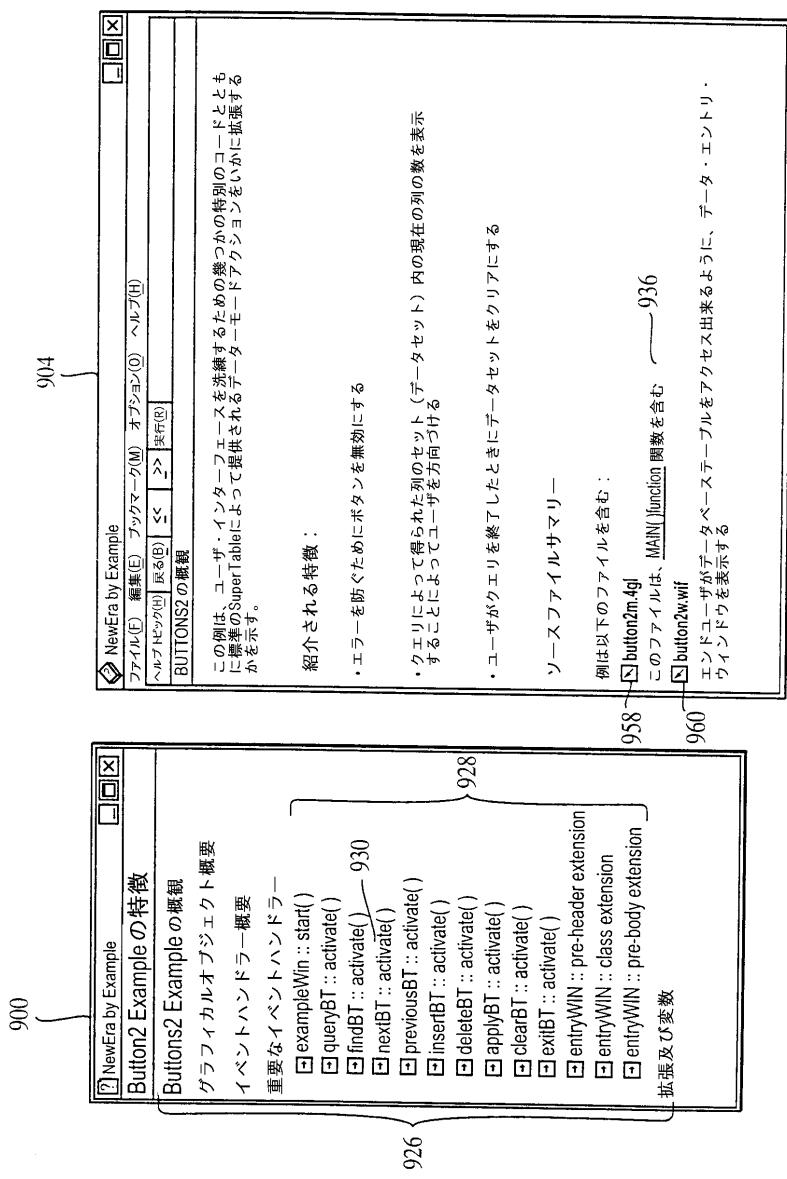


図17

## 【誤訳訂正16】

【訂正対象書類名】図面

【訂正対象項目名】図18

【訂正方法】変更

【訂正の内容】

【図18】

904

```

nextBT :: activate()
960 次のボタンのためのアクティベートハンドラー 934
  button2w.wif - in nextBT handler for ixButton::activate event:
  {
    VARIABLE ok BOOLEAN
    VARIABLE SuperTable ixSuperTable
    VARIABLE rowPosition INTEGER
    LET SuperTable = (getVisualContainer() CAST ixSuperTable)
    LET rowPosition = SuperTable.getCurRowNum() +1
  } 932
  現在の表示モードのための列数を得る:
  IF rowPosition > SuperTable.getNumStoredRows(NULL) THEN} 932
  LET rowPosition = ixSuperTable::lastRow
  END IF
  リターン状態について何もしない:
  LET ok = SuperTable.setCurrentCell(rowPosition, ixSuperTable::currentColumn)} 932
  ボタン状態をセットする:
  CALL (getWindow() CAST exampleWin).resetSuperTableButtons( )} 932
  現在の列位置を示す:
  CALL (getWindow() CAST exampleWin).showRowInfo()} 932

```

図18

【誤訳訂正17】

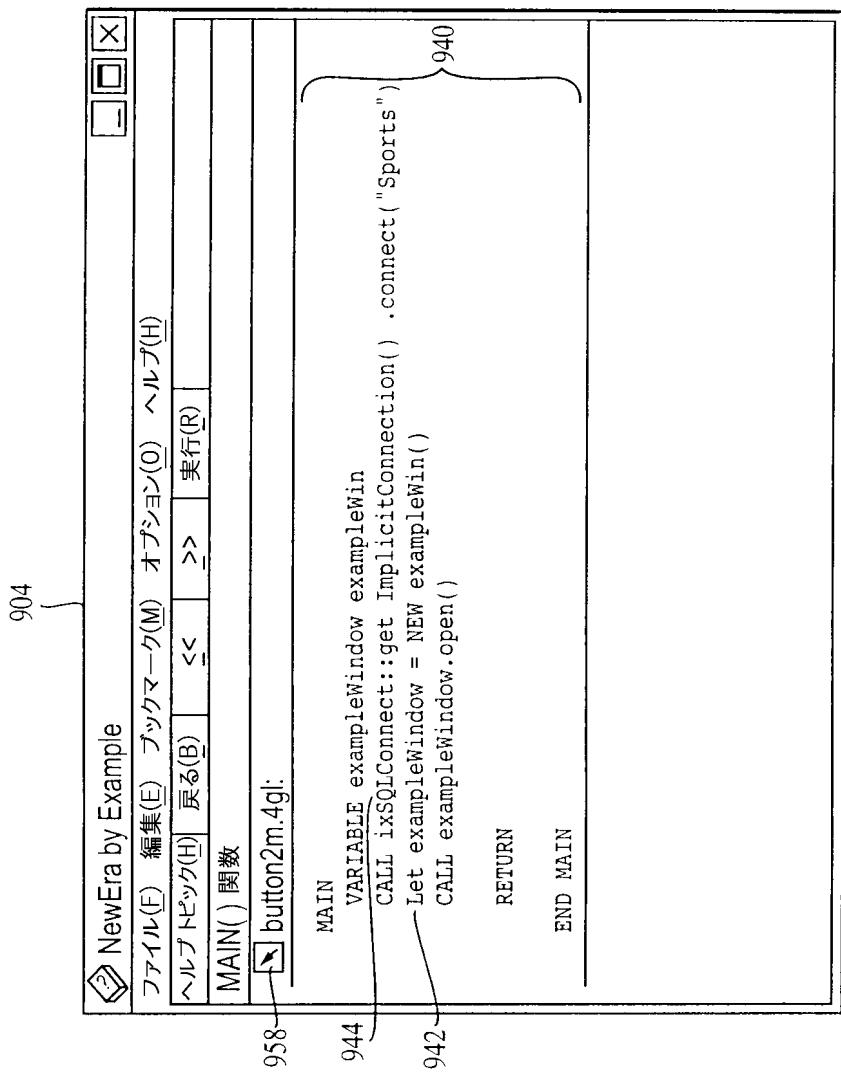
【訂正対象書類名】図面

【訂正対象項目名】図19

【訂正方法】変更

【訂正の内容】

【 図 1 9 】



19

### 【誤訳訂正 18】

【訂正対象書類名】図面

### 【訂正対象項目名】図 2 0

## 【訂正方法】変更

## 【訂正の内容】

【図20】

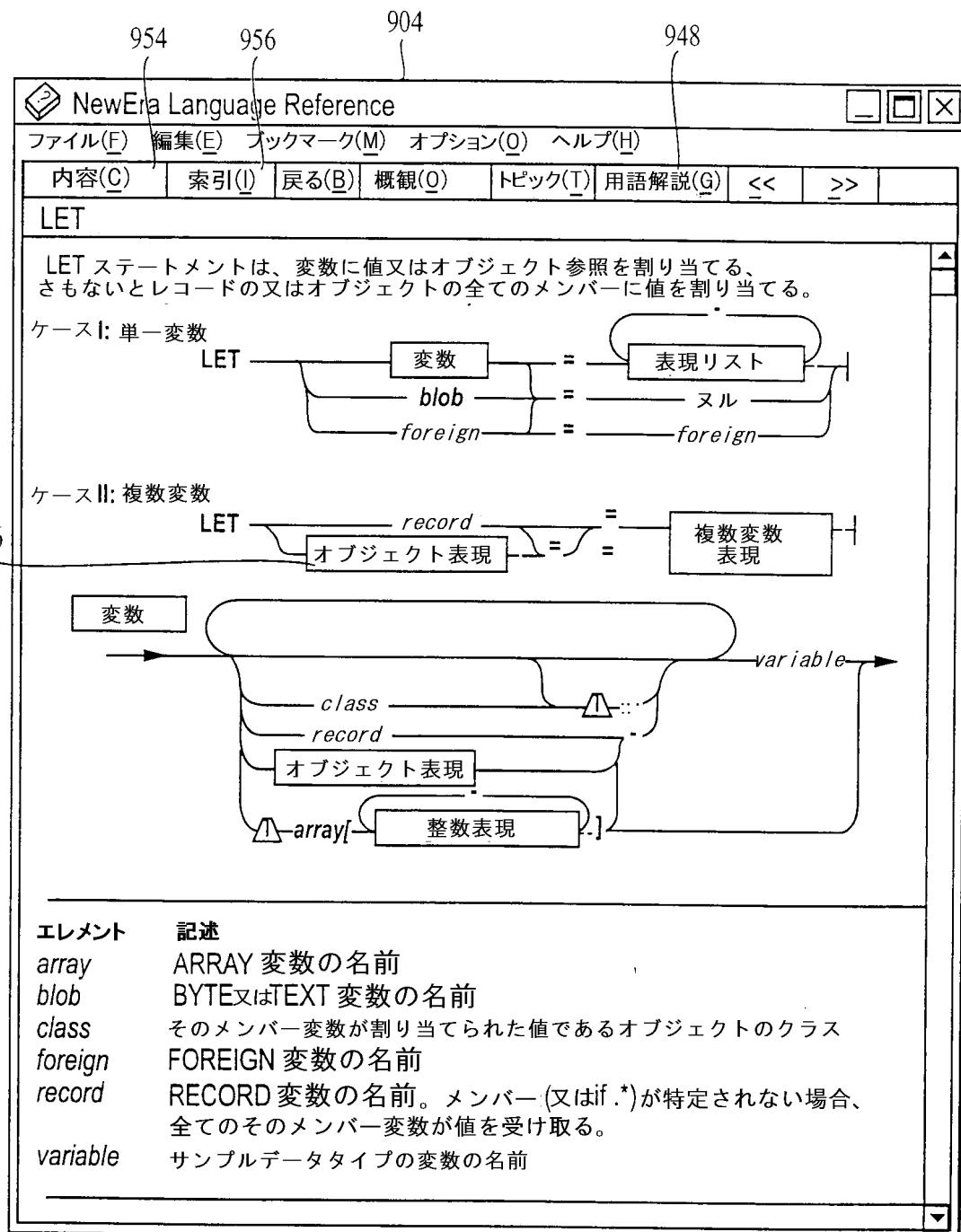


図20

【誤訳訂正19】

【訂正対象書類名】図面

【訂正対象項目名】図21

【訂正方法】変更

【訂正の内容】

【図21】

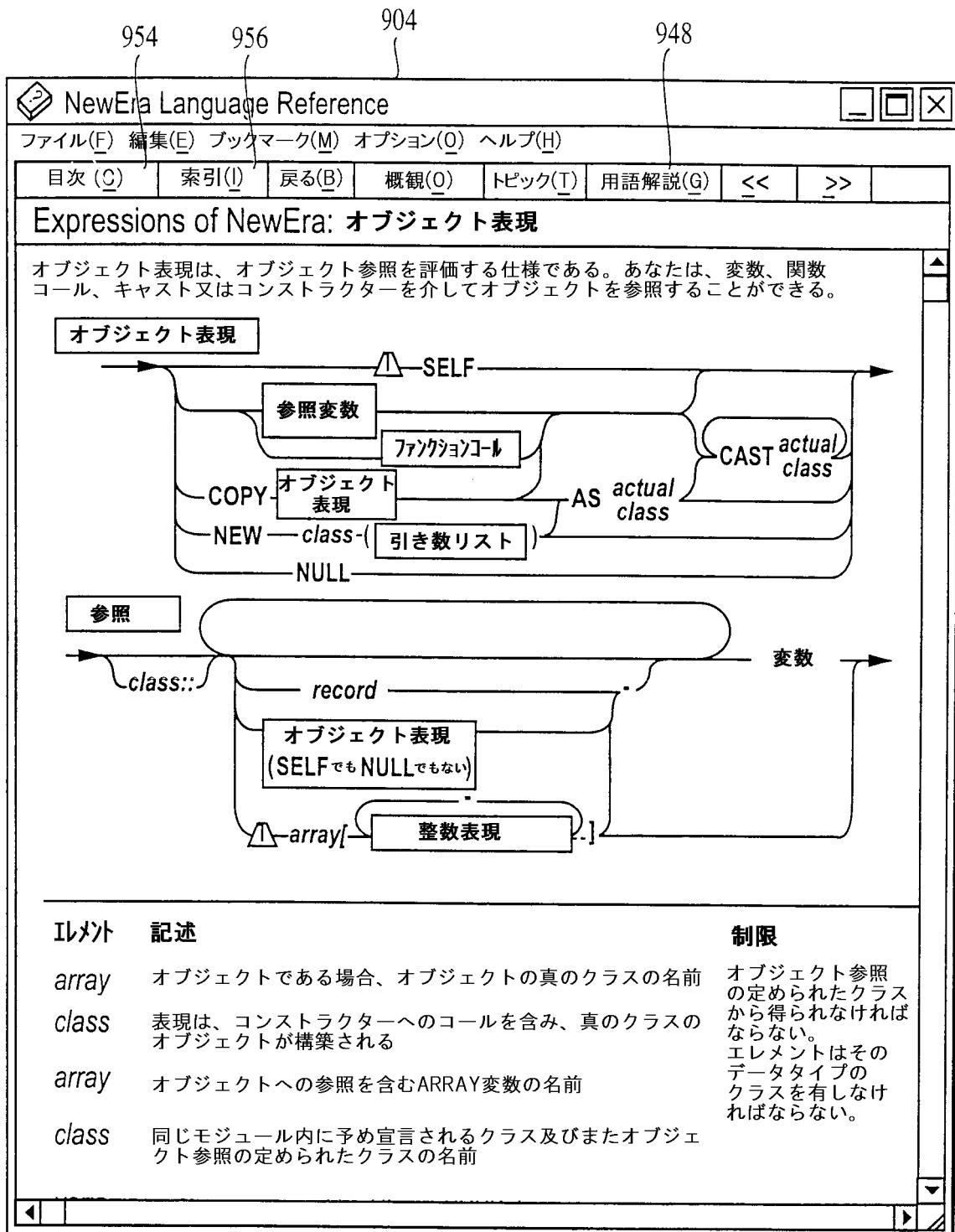


図21

【誤訳訂正20】

【訂正対象書類名】図面

【訂正対象項目名】図22

【訂正方法】変更

【訂正の内容】

【図22】

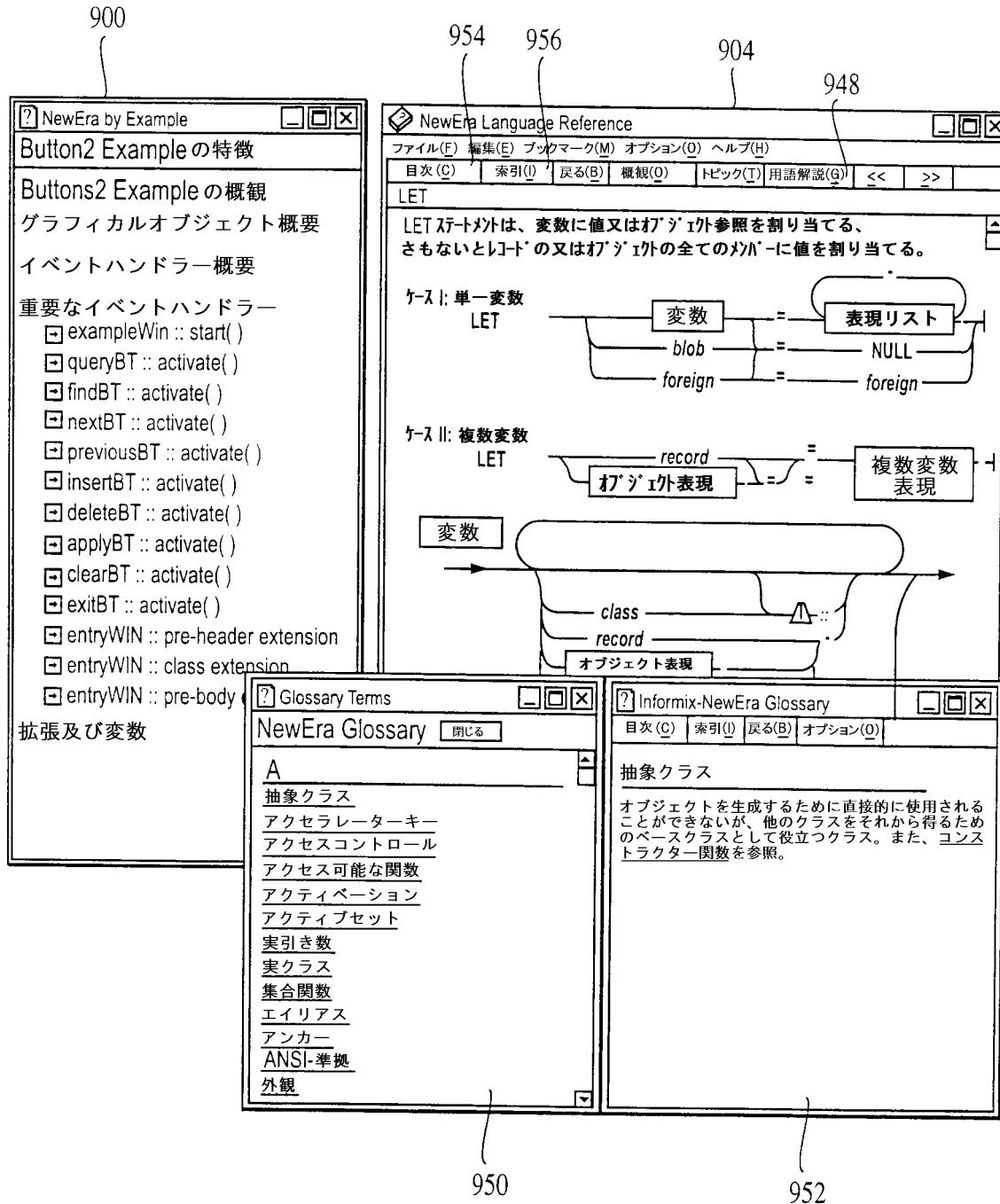


図22

【誤訳訂正21】

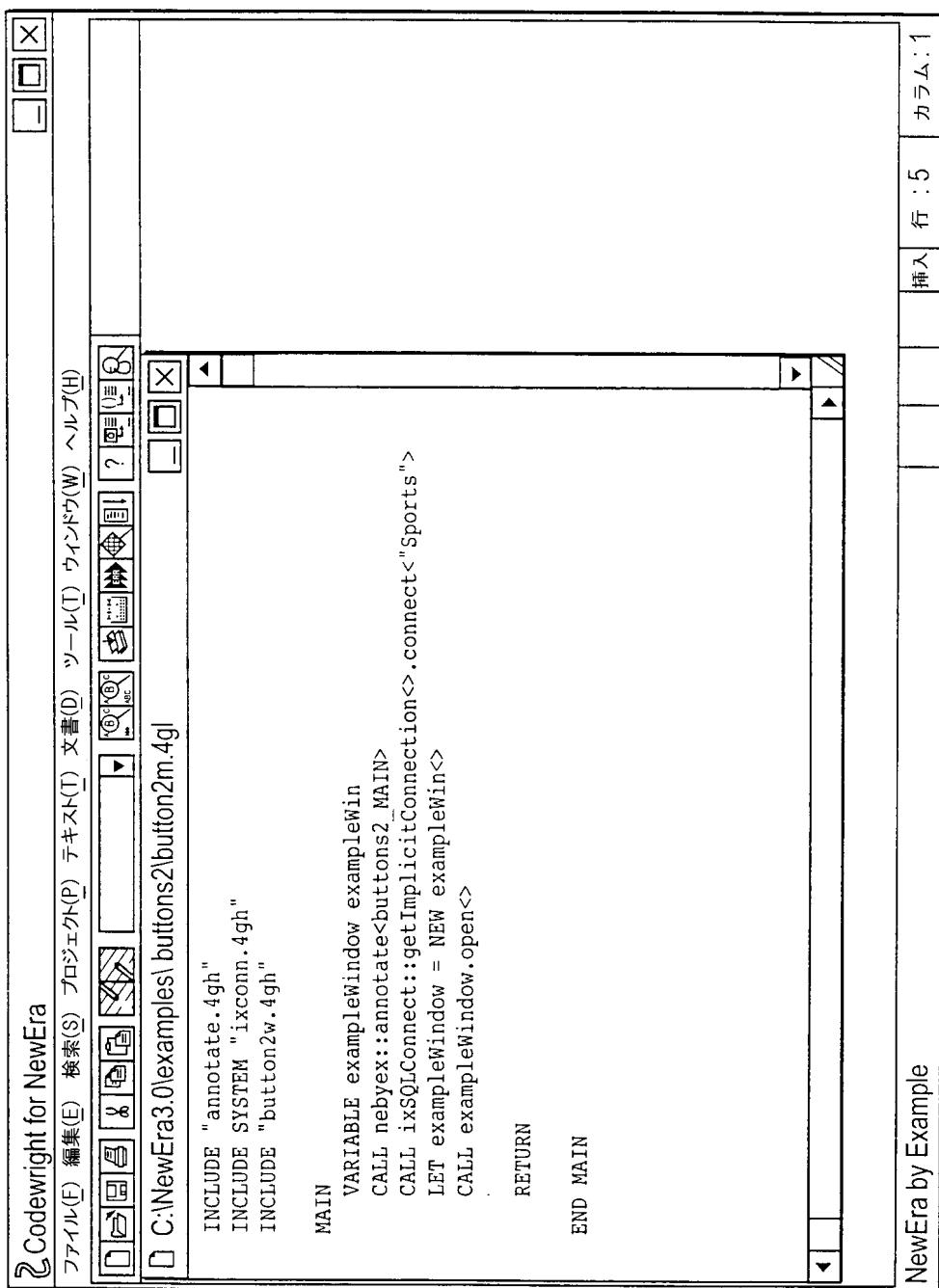
【訂正対象書類名】図面

【訂正対象項目名】図23

【訂正方法】変更

【訂正の内容】

【図23】



The screenshot shows the Codewright for NewEra interface. The menu bar includes ファイル(F), 検索(S), プロジェクト(P), テキスト(T), 文書(D), ツール(T), ウィンドウ(W), ヘルプ(H). The toolbar contains various icons for file operations. The code editor window displays the following C code:

```

Codewright for NewEra
C:\NewEra3.0\examples\buttons2\button2m.4gl

INCLUDE "annotate.4gh"
INCLUDE SYSTEM "ixconn.4gh"
INCLUDE "button2w.4gh"

MAIN
  VARIABLE exampleWindow exampleWin
  CALL nebyex::annotate<buttons2_MAIN>
  CALL ixSQLConnect::getImplicitConnection>.connect<"Sports">
  LET exampleWindow = NEW exampleWin<>
  CALL exampleWindow.open<>

  RETURN

END MAIN

```

The status bar at the bottom right shows "NewEra by Example" and "挿入 行: 5 カラム: 1".

図23

【誤訳訂正22】

【訂正対象書類名】図面

【訂正対象項目名】図24

【訂正方法】変更

【訂正の内容】

【図24】

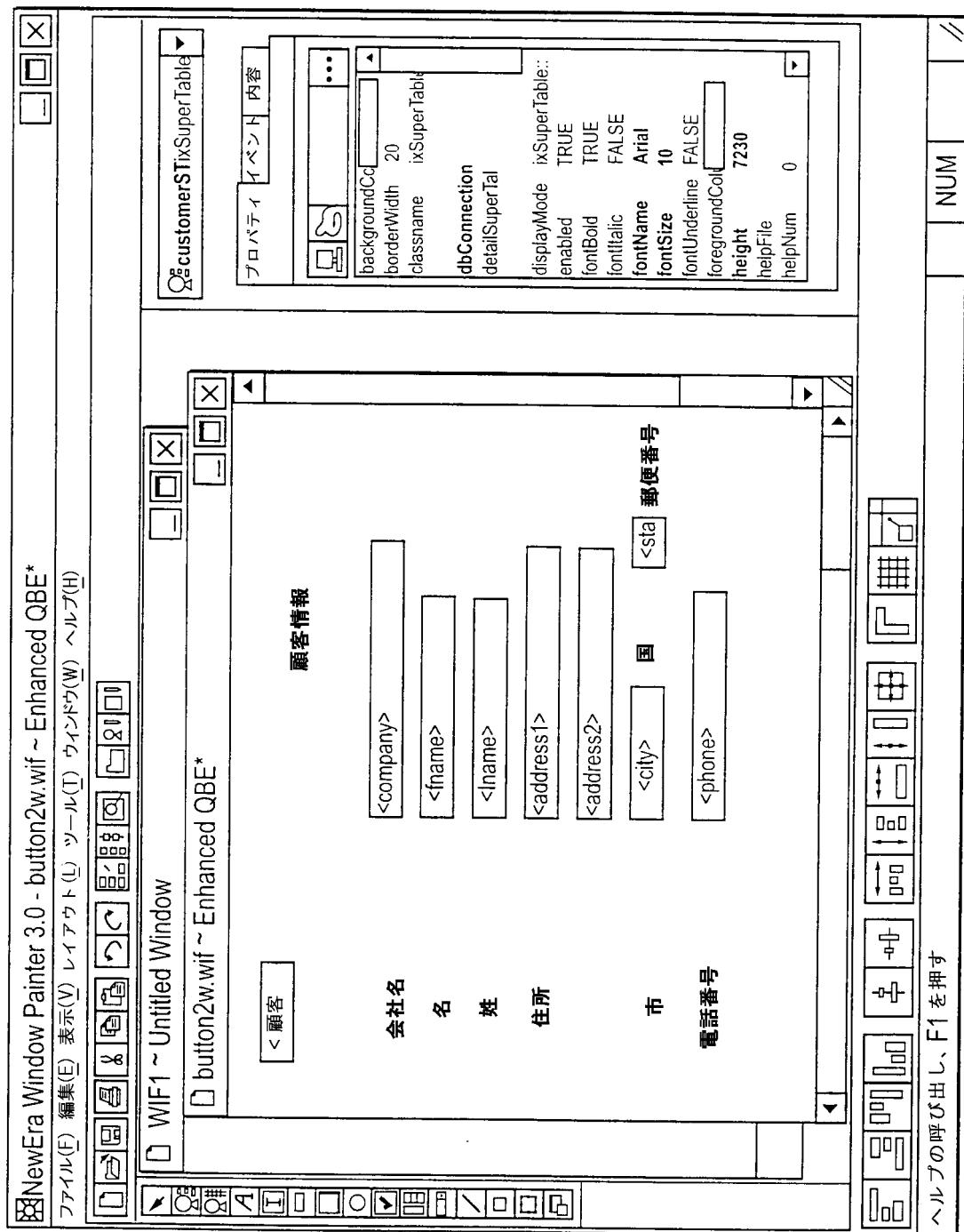


図24

【誤訳訂正23】

【訂正対象書類名】図面

【訂正対象項目名】図25

【訂正方法】変更

【訂正の内容】

【図25】

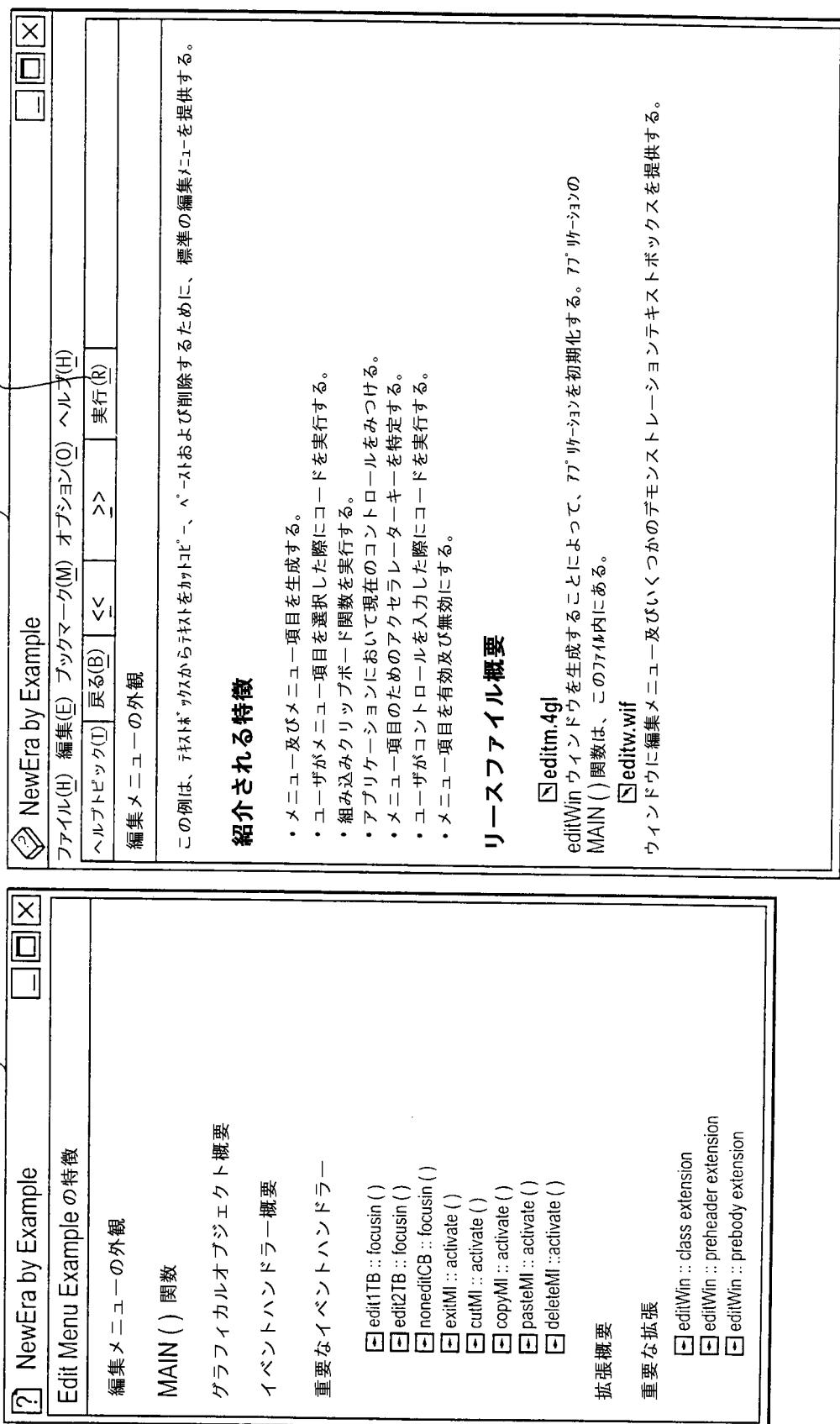
【誤訳訂正24】  
【訂正対象書類】画面

図25

【訂正対象項目名】図26

【訂正方法】変更

【訂正の内容】

【 図 2 6 】

## 【誤訳訂正 25】

### 【訂正対象書類名】団面

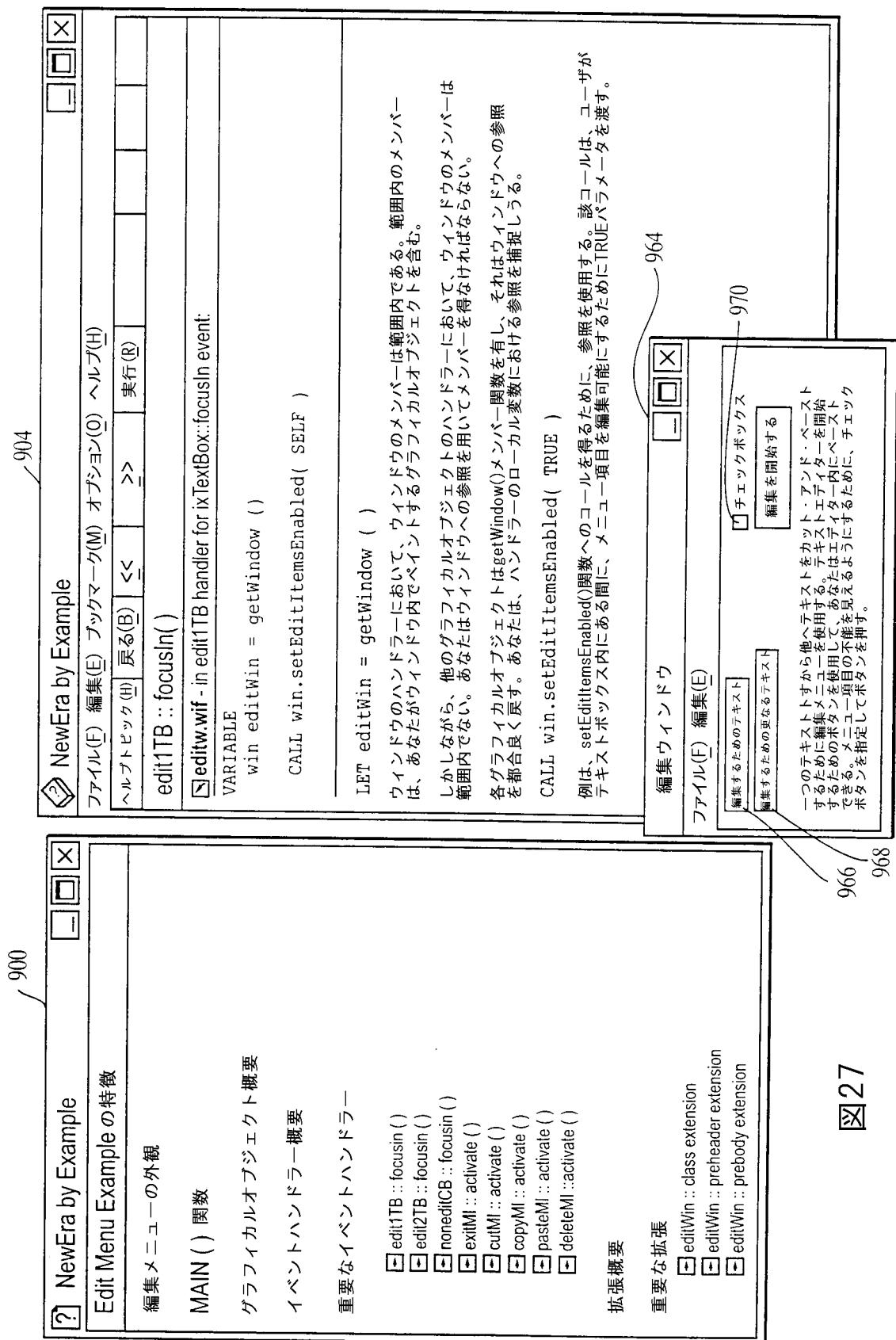
26

【訂正対象項目名】図27

【訂正方法】変更

【訂正の内容】

【 図 27 】



## 【誤訳訂正 2 6 】

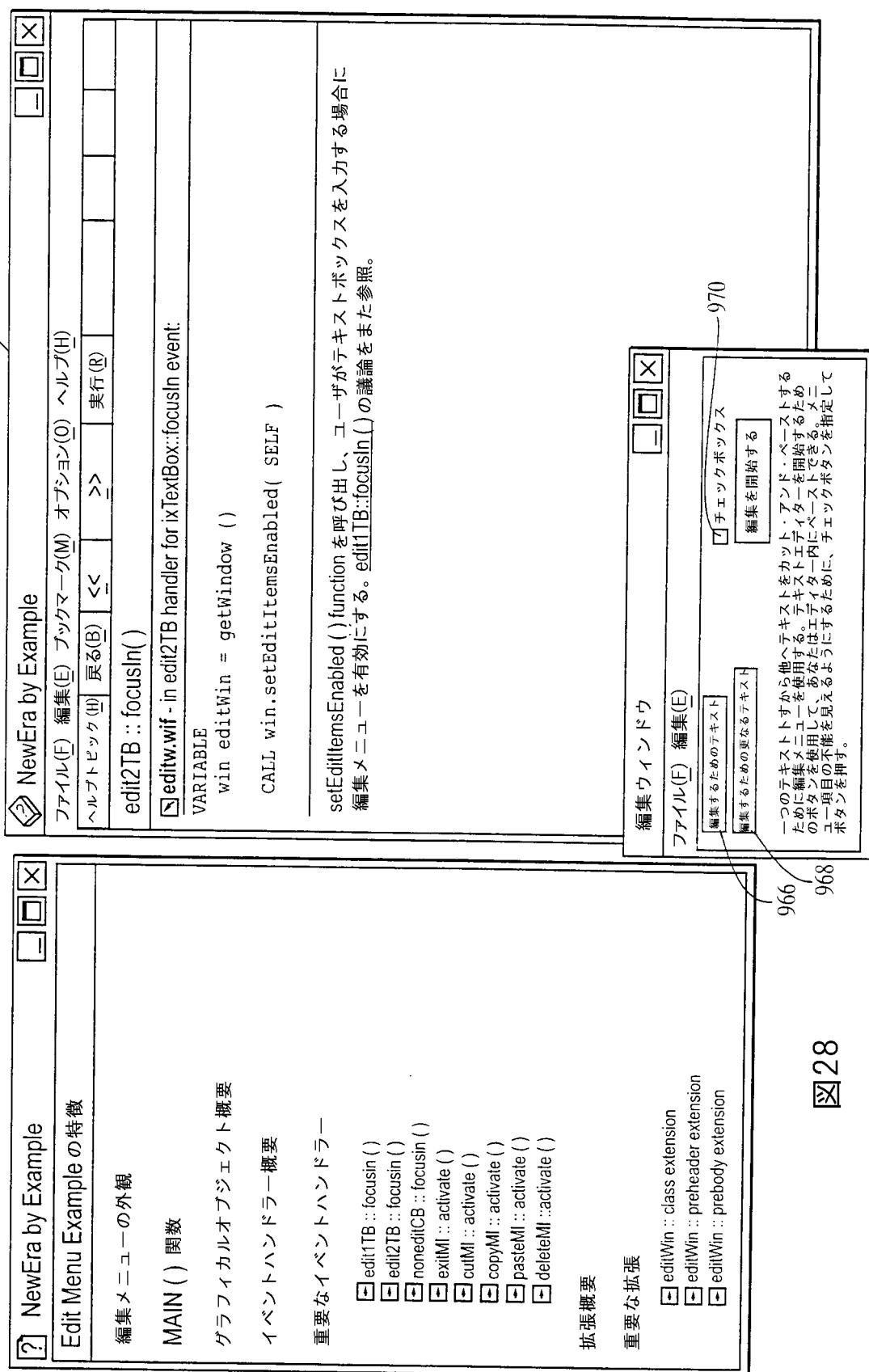
【訂正対象書類名】 図面

【訂正対象項目名】図28

【訂正方法】変更

【訂正の内容】

【図28】



【訂正対象項目名】図 2 9

【訂正方法】変更

【訂正の内容】

【図29】

904

【?】 NewEra by Example

□□×

Edit Menu Example の特徴

編集メニューの外観

MAIN () 関数

グラフィカルオブジェクト概要

イベントハンドラー概要

重要なイベントハンドラー

拡張概要

重要な拡張

ファイル(F) 編集(E) ブックマーク(M) オプション(O) ヘルプ(H)

ヘルプトピック(H) 戻る(B) << >> 実行(R)

noneditCB :: focusIn()

editw.wif - in noneditCB handler for ixCheckBox::focusIn event:

```

VARIABLE
  win editWin = getWindow()
  CALL win.setEditItemsEnabled( SELF )

```

Edit1 TB::focusIn()と同じように働くが、ユーザがテキストボックス内にある間に、  
メニュー項目を編集不能にするためにFALSEパラメータを渡す。

SetEditItemsEnabled () functionを呼び出し、ユーザがテキストボックスを入力する場合に  
編集メニューを有効にする。ユーザは、チェックボックス内にペーストできない。

編集ウインドウ

ファイル(F) 編集(E)

編集するためのテキスト

編集するためのテキスト

チェックボックス

編集を開始する

一つのテキストトスから他へテキストをカット・ペーストする  
ために編集メニューを使用する。テキストエディター内にペーストできる。メニ  
ューボタンを使用して、あるようにするため、チェックボタンを指定して  
ユーザの不満を見えるようにするため、チェックボタンを押す。

【誤訳訂正28】

【訂正対象書類】図面

図29

【訂正対象項目名】図32

【訂正方法】変更

【訂正の内容】

【図32】

```

FUNCTION driveStockRpt( destType SMALLINT, destName CHAR(*) )
1200 RETURNING VOID
  ↘{.normal

    オブジェクト、特に ixRow オブジェクトは、レポートフォーマッターに引数を渡されること
    ができない数に、フェッチされたデータの列は、フェッチされた列におけるエレメントの
    データタイプ及び長さにマッチするレコード内にアンパックされるだろう。

  }

  VARIABLE
    stockRec RECORD
      mn CHAR(15),      -- manufact.manu_name
      sn SMALLINT,      -- stock.stock_num
      sd CHAR(15),      -- stock.description
      sp MONEY(6,2),    -- stock.unit_price
      su CHAR(4)        -- stock.unit
    END RECORD

    stockStmt ixSQLStmt,
    stmtString CHAR(*),
    stockRow ixRow,

    errorCode INTEGER,
    logFile ixErrorLog

1205
  ↘{.normal

    暗黙接続オブジェクトを使用して、SQL ステートメントオブジェクトを生成する。
    該接続オブジェクトは、データベースにすでに接続されなければならない。
    prepare( ) コールの状態をチェックすることは、これを承認する。

  }

1210
  ↘{.[edit stmt]
    LET stockStmt =
      ixSQLConnect:::getImplicitConnection().createStmtObject()
  ↗{.]file stmt
1215

```

## 図32

【誤訳訂正29】

【訂正対象書類名】図面

【訂正対象項目名】図33

【訂正方法】変更

【訂正の内容】

【図33】

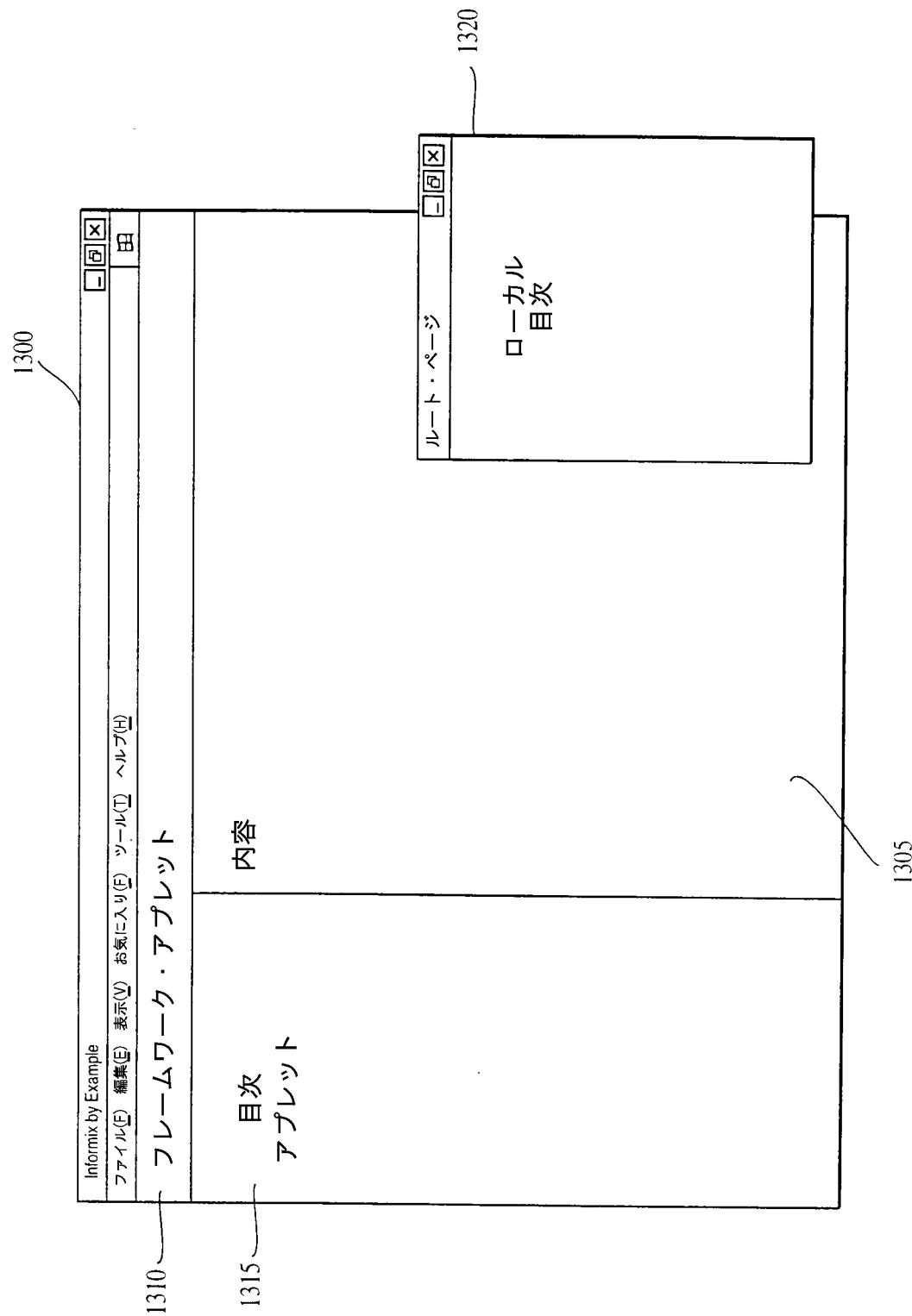


図33

【誤訳訂正30】

【訂正対象書類名】図面

【訂正対象項目名】図34

【訂正方法】変更

【訂正の内容】

【図34】

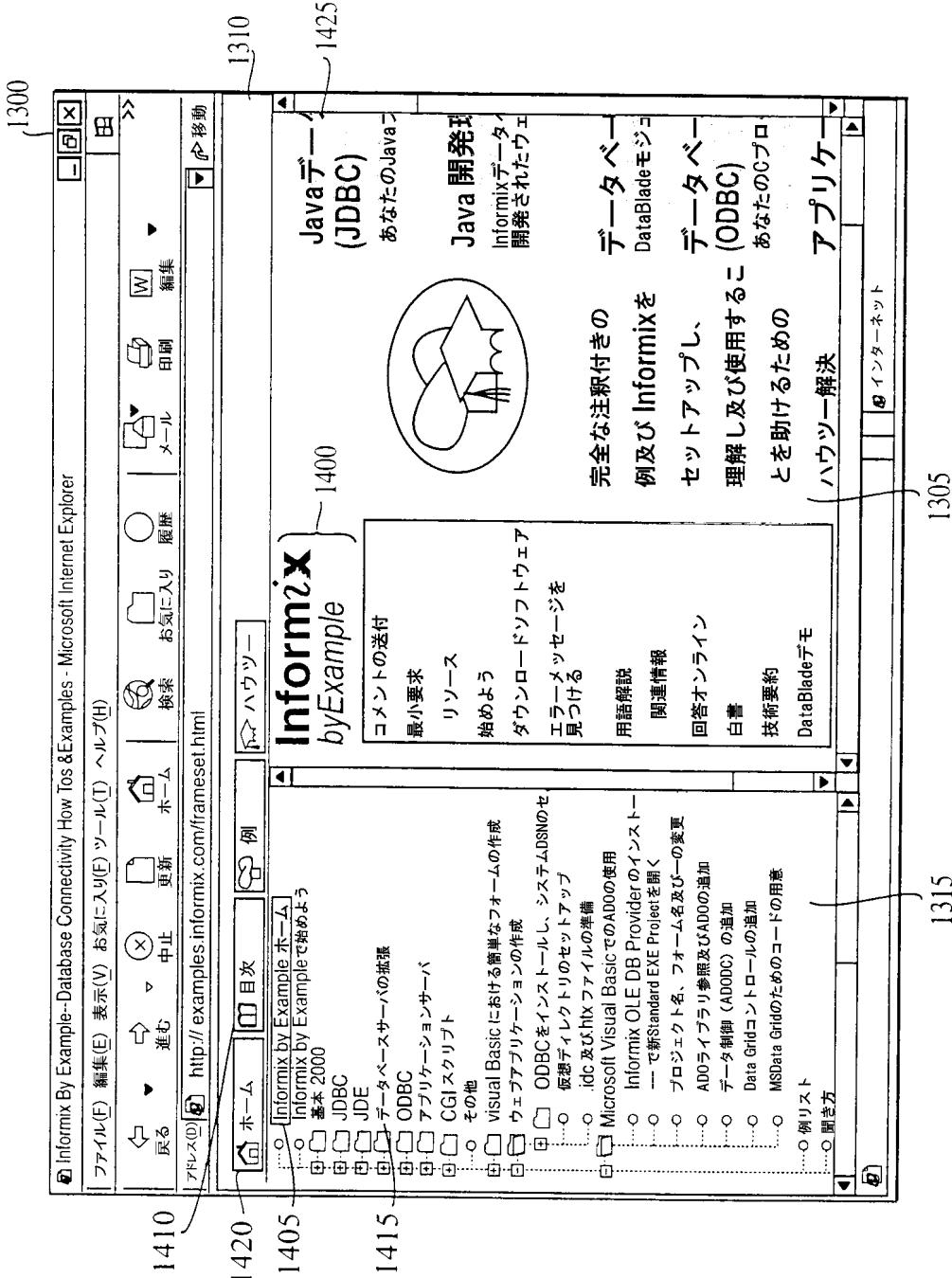


図34

【誤訳訂正31】

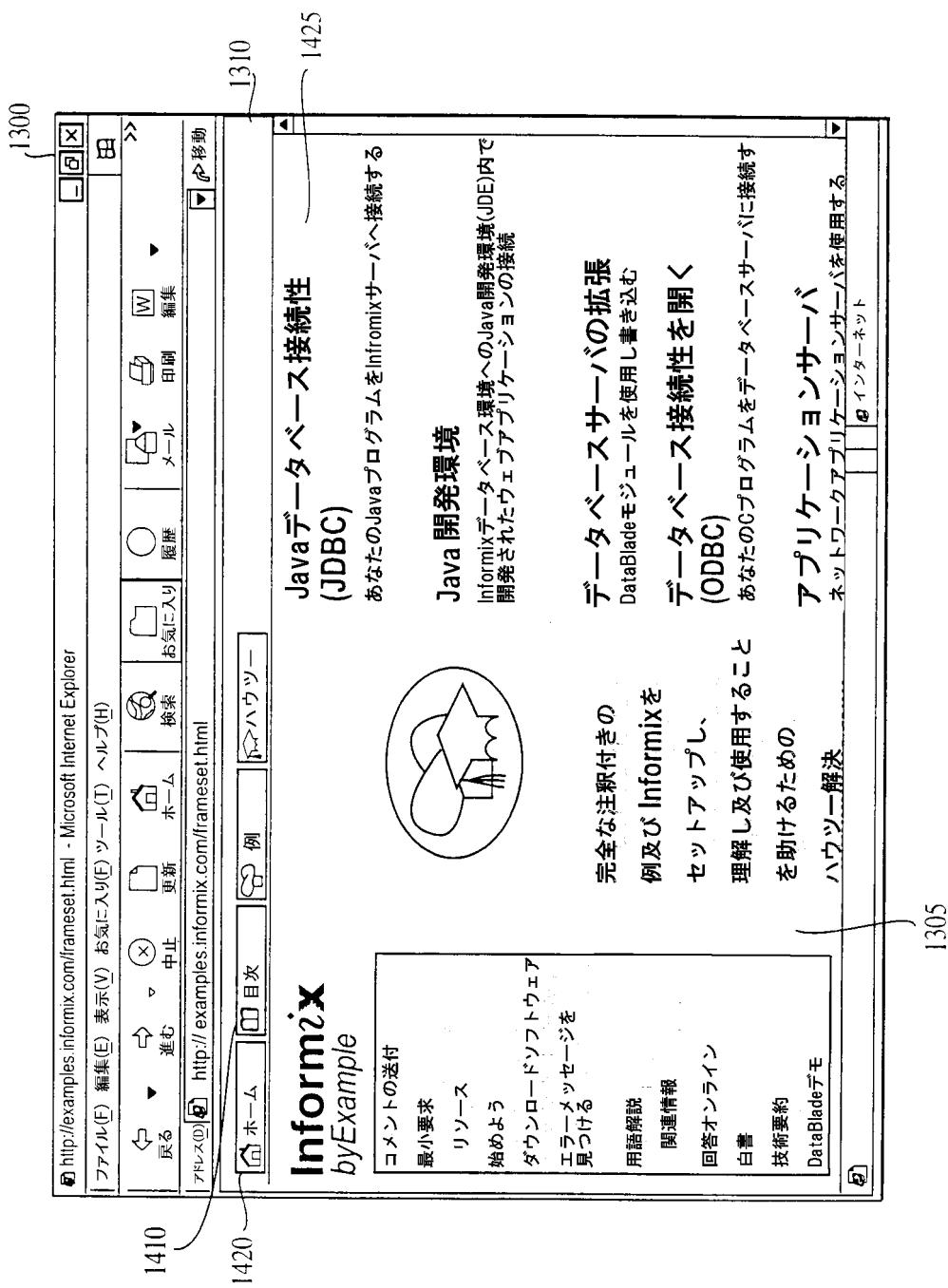
【訂正対象書類名】図面

【訂正対象項目名】図35

【訂正方法】変更

【訂正の内容】

【図35】



【翻訳訂正32】

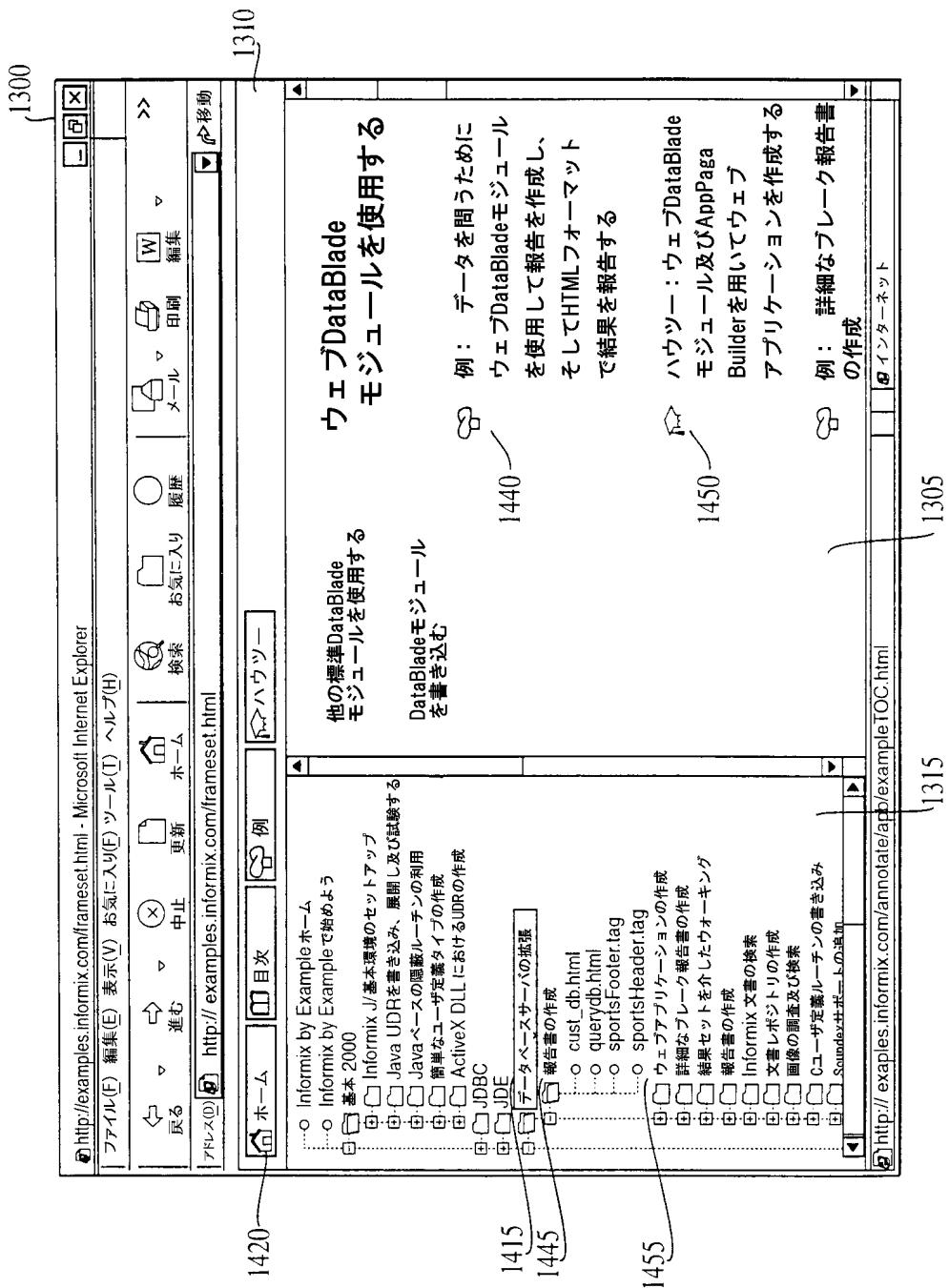
【訂正対象書類名】図面

【訂正対象項目名】図36

【訂正方法】変更

【訂正の内容】

【 図 3 6 】



36

### 【誤訳訂正 3 3】

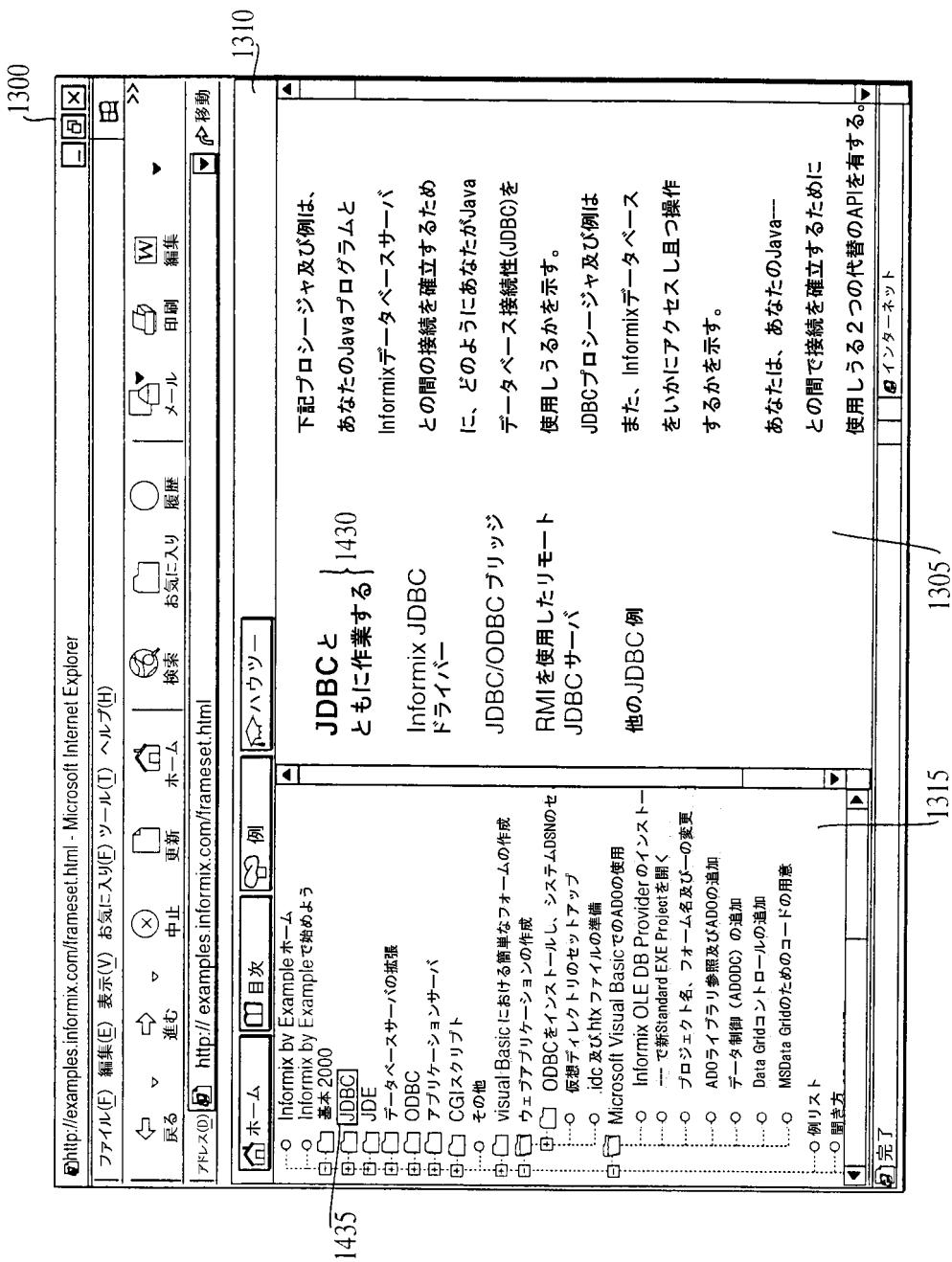
### 【訂正対象書類名】 図面

### 【訂正対象項目名】図 3-7

### 【訂正方法】変更

### 【訂正の内容】

【 図 3 7 】



37

### 【誤訳訂正 3 4】

### 【訂正対象書類名】図面

### 【訂正対象項目名】図 3 8

### 【訂正方法】変更

### 【訂正の内容】

【図38】

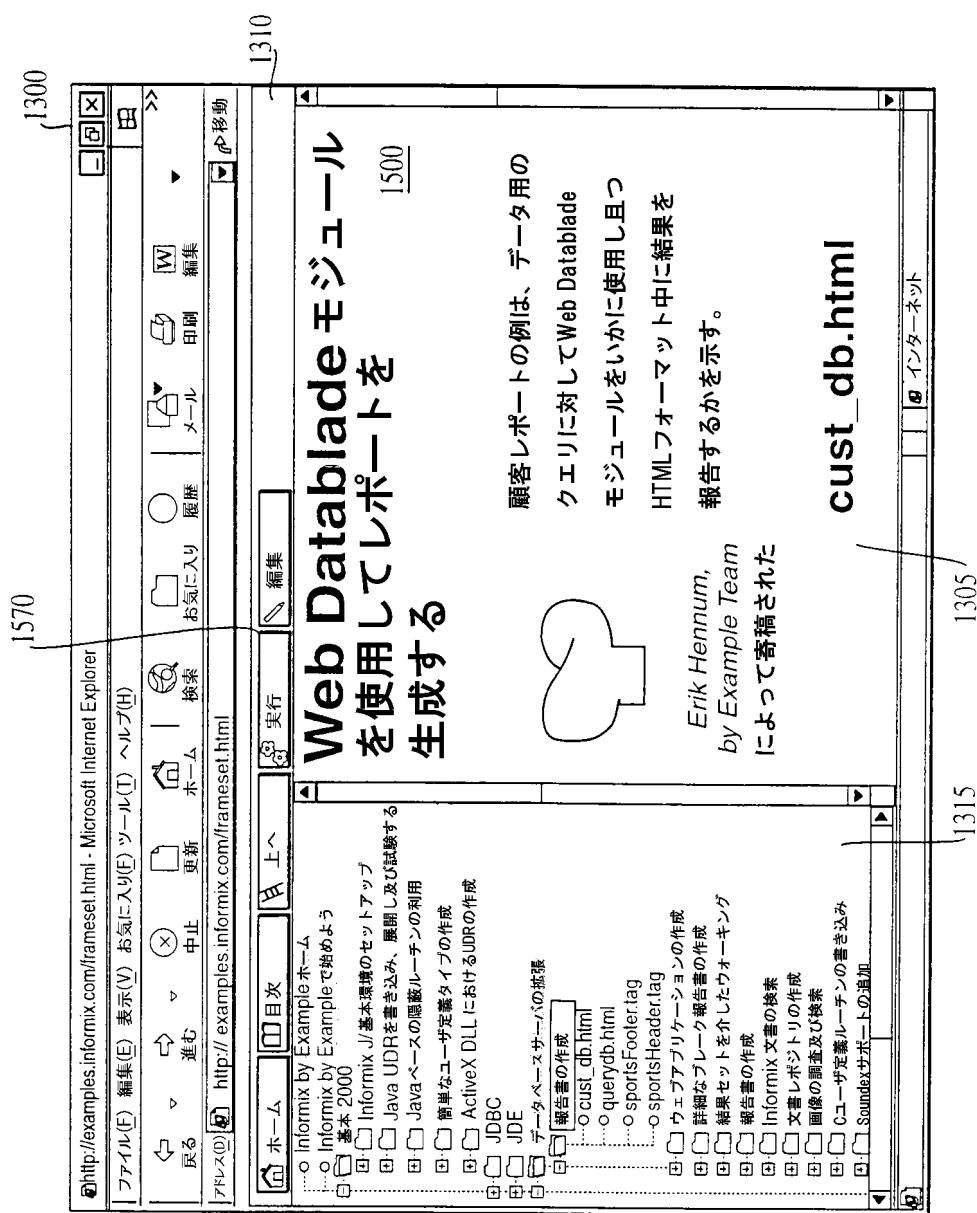


図38

【誤訳訂正35】

【訂正対象書類名】図面

【訂正対象項目名】図39

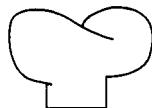
【訂正方法】変更

【訂正の内容】

【図39】

1500

## Web Datablade モジュールを使用して レポートを生成する



顧客レポートの例は、データ用のクエリに対してWeb Datablade モジュールをいかに使用し且つHTMLフォーマット中に結果を報告するかを示す。 — 1505

Erik Hennum,  
by Example Team  
によって寄稿された

`cust_db.html` — 1515

1520

このappページはクエリを受け入れ、そしてHTMLレポートを作成する

1510

`querydb.html` — 1515

1520

このHTMLページはappページを呼び出すフォームを含む

`sportsFooter.tag` — 1515

1520

SportsFooterダイナミックタグは、app page用のフッターを生成する

`sportsHeader.tag` — 1515

1520

SportsHeaderダイナミックタグは、app page用のヘッダーを生成する

☞ この例のソースファイルの全てを見る又は印刷するためには、  
ここをクリックする

図39

【誤訳訂正36】

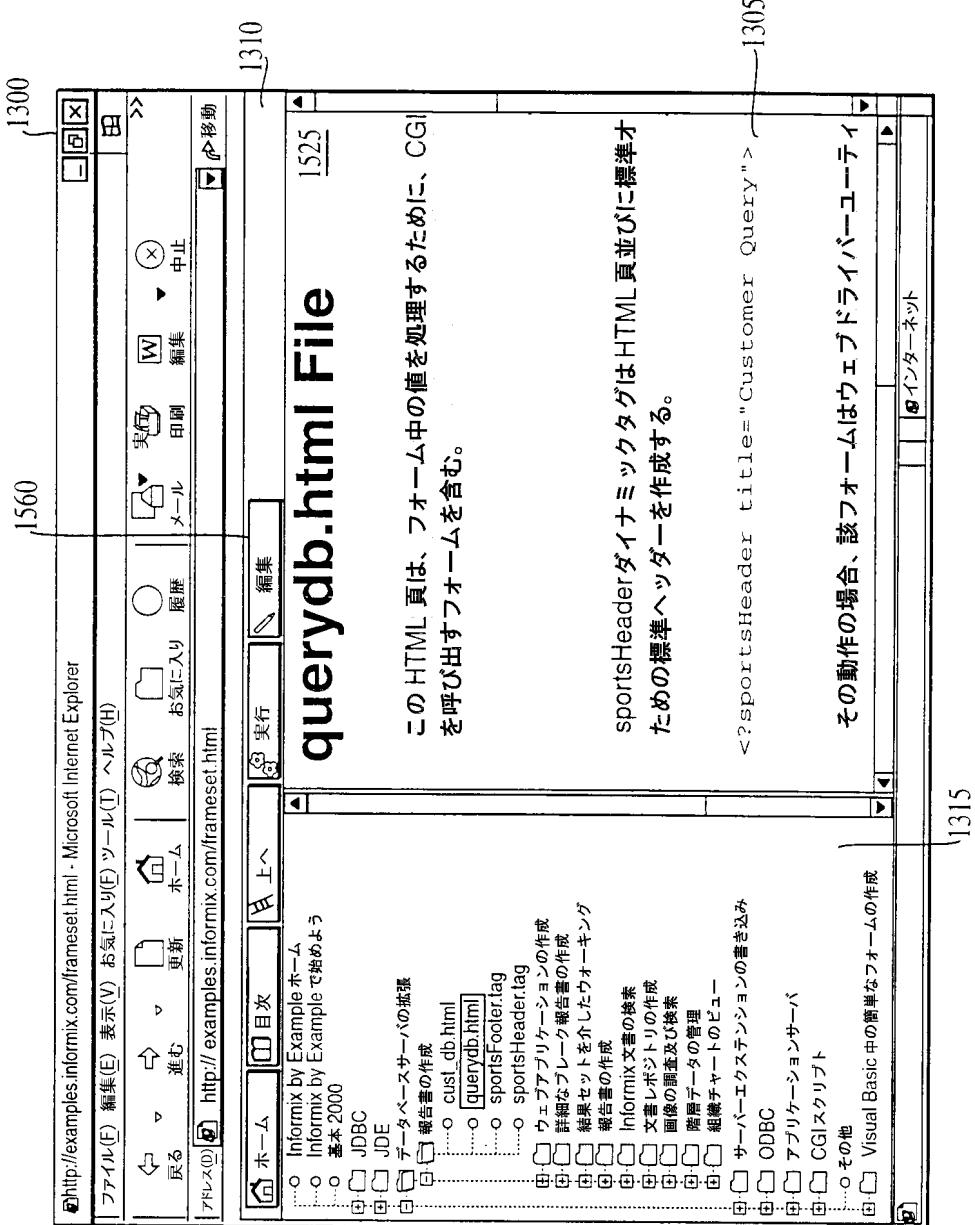
【訂正対象書類名】図面

【訂正対象項目名】図40

【訂正方法】変更

【訂正の内容】

【図40】



【誤訳訂正37】

【訂正対象書類名】図面

【訂正対象項目名】図41

【訂正方法】変更

【訂正の内容】

図40

【図41】

1525**querydb.html File**

このHTML頁は、フォーム中の値を処理するために、CGIの代わりにapp pageを呼び出すフォームを含む。

sportsHeaderダイナミックタグはHTML頁並びに標準オープニングテキストのための標準ヘッダーを作成する。

```
<?sportsHeader title="Customer Query"> } 1530
      1540
```

その動作の場合、該フォームはウェブドライバユーティリティを特定しなければならない。

```
<P>
<FORM ACTION = "<?MIVAR>$WEB_HOME<?/MIVAR" METHOD="GET"> } 1530
      1550
```

app pageを特定するために、該フォームは隠されたコンポーネントを使用しなければならない。該コンポーネントは、MIVARの名前及びapp pageの名前である値を有しなければならない。下記のコンポーネントは、cust\_db.html app pageを特定する。

```
<INPUT TYPE="HIDDEN" NAME="MIVAR" VALUE="/examples/CustRpt/cust_db.html">
Optional state:
<INPUT TYPE="TEXT" NAME="SelectState" SIZE="3" MAXLENGTH="2">
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="SUBMIT">
</FORM>
</P>
```

```
</BODY>
</HTML>
```

1535

1530

図41

【誤訳訂正38】

【訂正対象書類名】図面

【訂正対象項目名】図42

【訂正方法】変更

【訂正の内容】

## 【図42】

```

<!--<ibyx>
<intro>
<p><abstract> この HTML 頁は、app pageを呼び出すフォームを含む。
    </abstract> フォーム中の値を処理するために、CGI の代わりに
    </p>
    </intro>
    </ibyx> -->

<!-- <ibyx>
<p> sportsHeader ダイナミックタグは HTML 頁並びに標準オープニング
テキストのための標準ヘッダーを作成する。
</p>
</ibyx> -->
<?sportsHeader title="Customer Query">

<!-- <ibyx>
<p> その動作の場合、該フォームはウェブドライバーユーティリティを特定しなければならない。
</p>
</ibyx>-->
<p>
<FORM ACTION="<?MIVAR>$WEB_HOME<?/MIVAR>" METHOD="GET">

<!-- <ibyx>
<p> app pageを特定するために、該フォームは隠された入力コンポーネントを使用
しなければならない。該入力コンポーネントは、<strong>MIVar</strong> の
名前及び app page の名前である値を有しなければならない。下記の入力コンポーネント
は、<a href="cust_db.html">cust_db.html</a> app page を特定する。
</p>
</ibyx> -->
<INPUT TYPE="HIDDEN" NAME="MIVar" VALUE="/examples/CustRpt/cust_db.html">

Optional state:
<INPUT TYPE="TEXT" NAME="selectState" SIZE="3" MAXLENGTH="2">
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">

</FORM>
</P>

<?annotate>

</BODY>
</HTML>

```

図42

## 【誤訳訂正39】

【訂正対象書類名】図面

【訂正対象項目名】図46

【訂正方法】変更

【訂正の内容】

【図46】

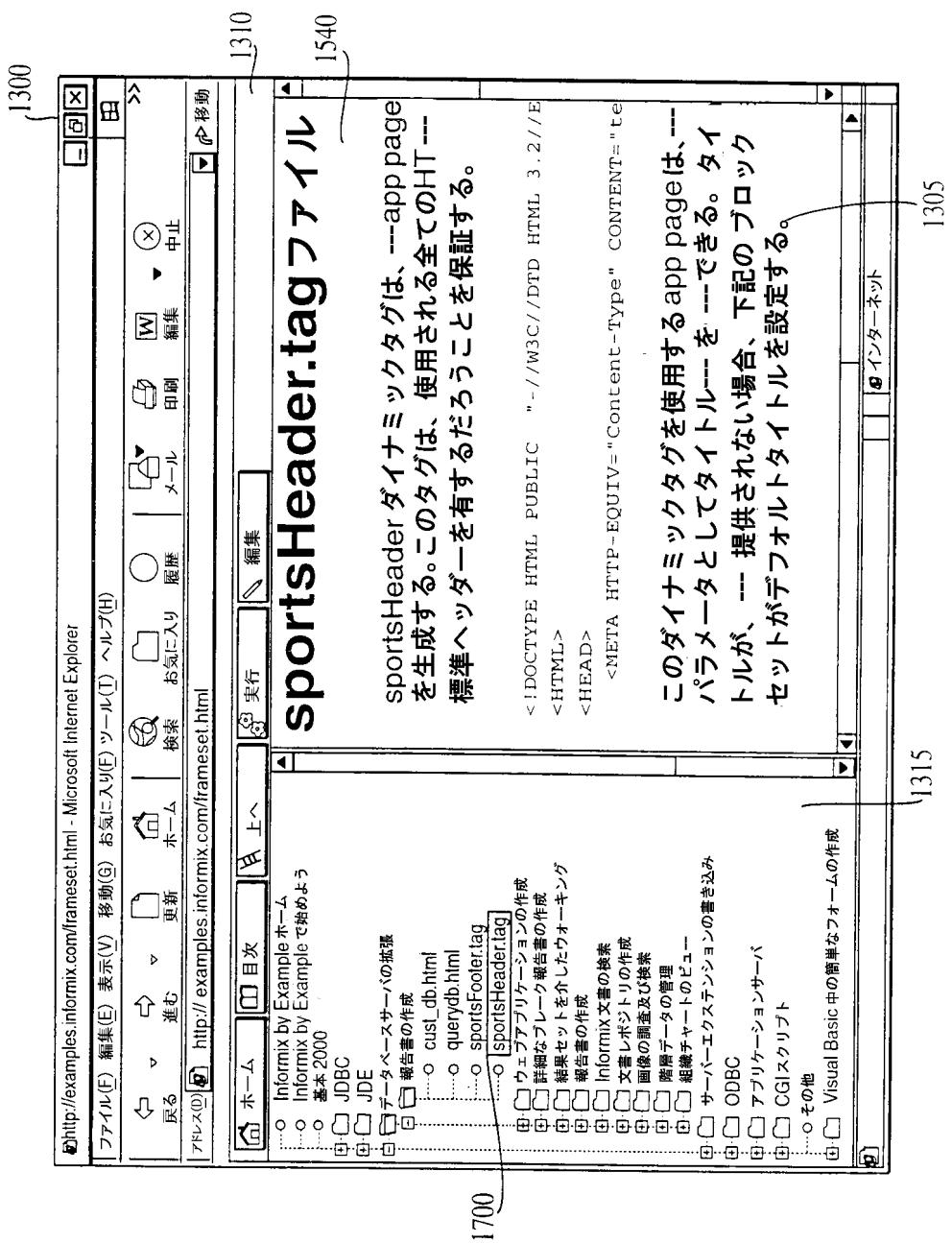


図46

【誤訳訂正40】

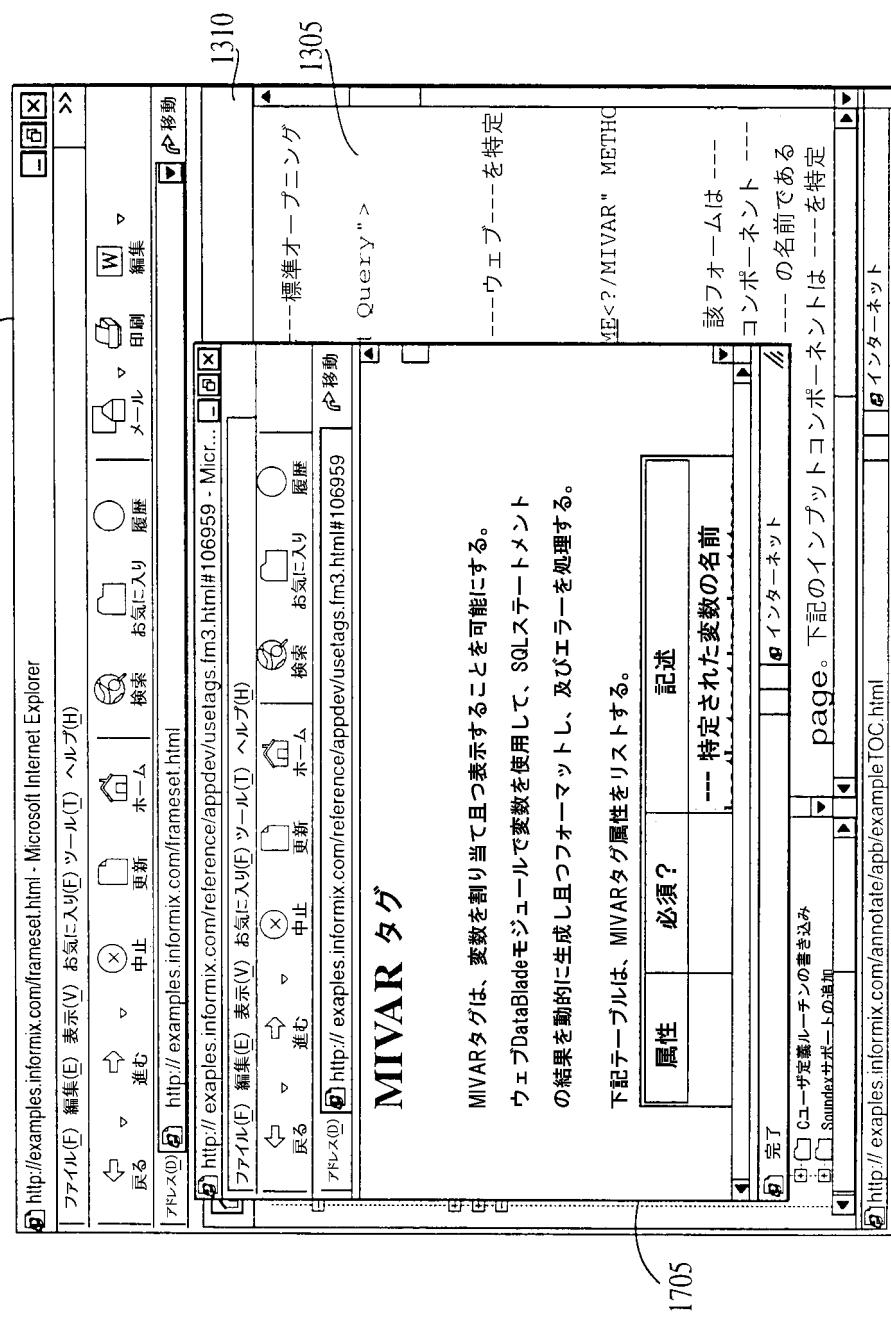
【訂正対象書類名】図面

【訂正対象項目名】図47

【訂正方法】変更

【訂正の内容】

【図47】



47

## 【誤訳訂正 4 1】

### 【訂正対象書類名】図面

### 【訂正対象項目名】図 4 8

### 【訂正方法】変更

## 【訂正の内容】

【図48】

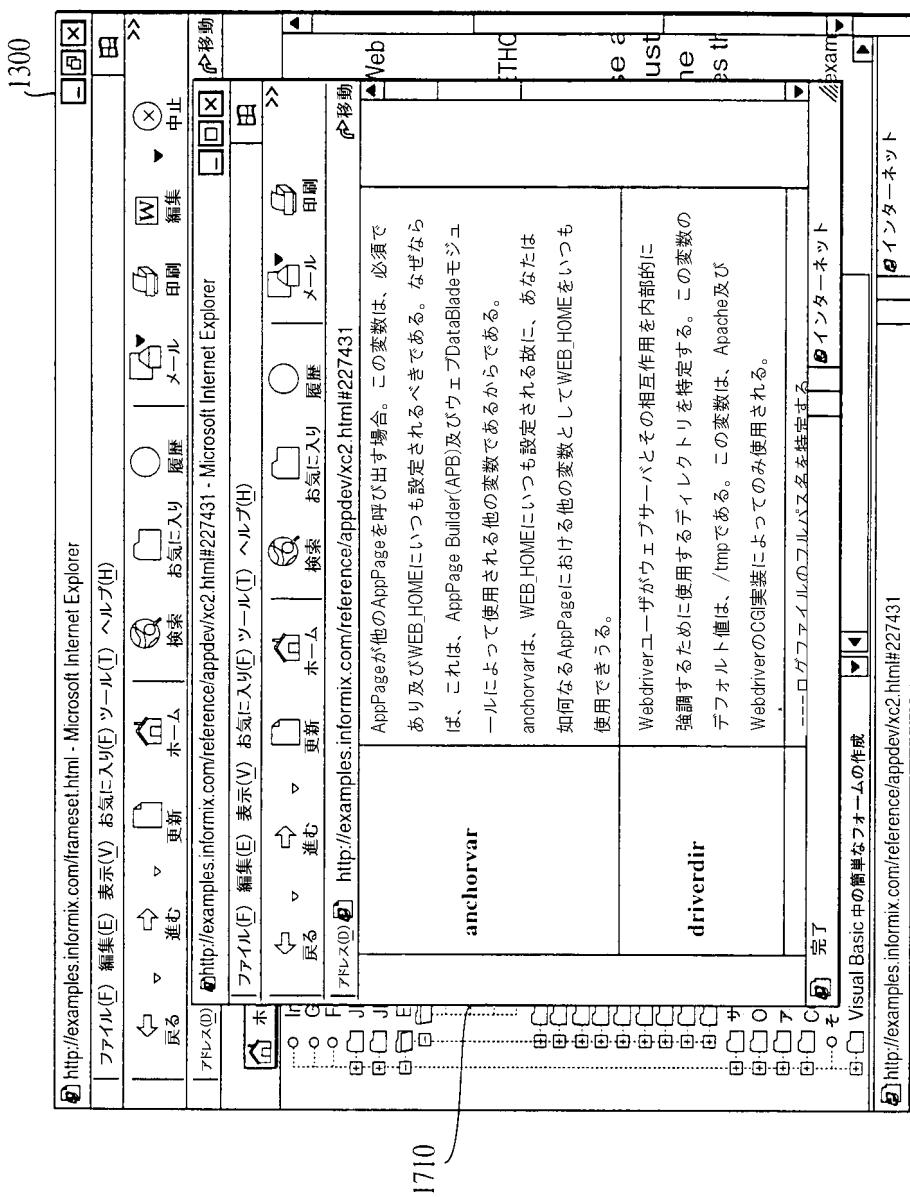


図48

## 【誤訳訂正42】

【訂正対象書類名】図面

【訂正対象項目名】図49

【訂正方法】変更

【訂正の内容】

【図49】

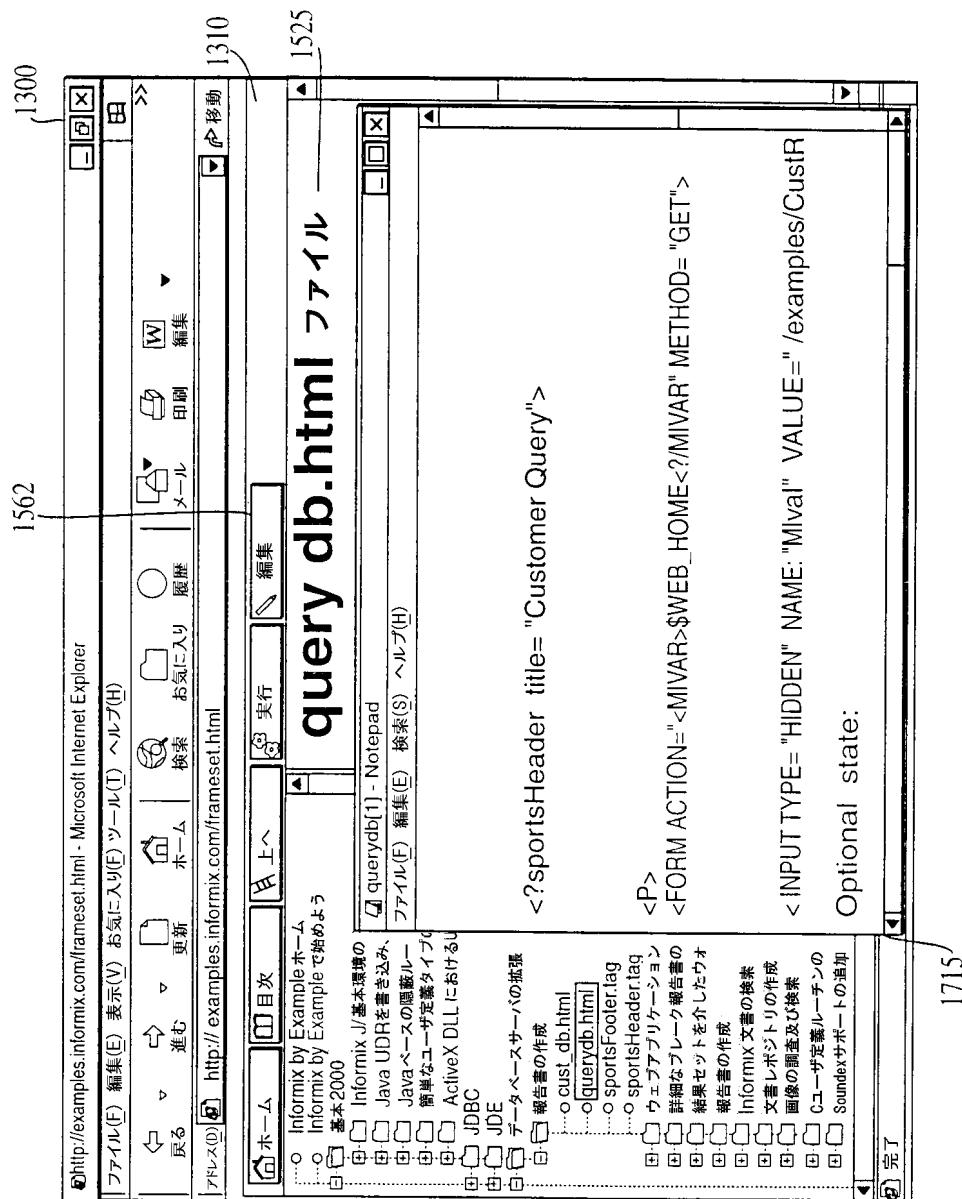


図49

【誤訳訂正43】

【訂正対象書類名】図面

【訂正対象項目名】図50

【訂正方法】変更

【訂正の内容】

【図50】

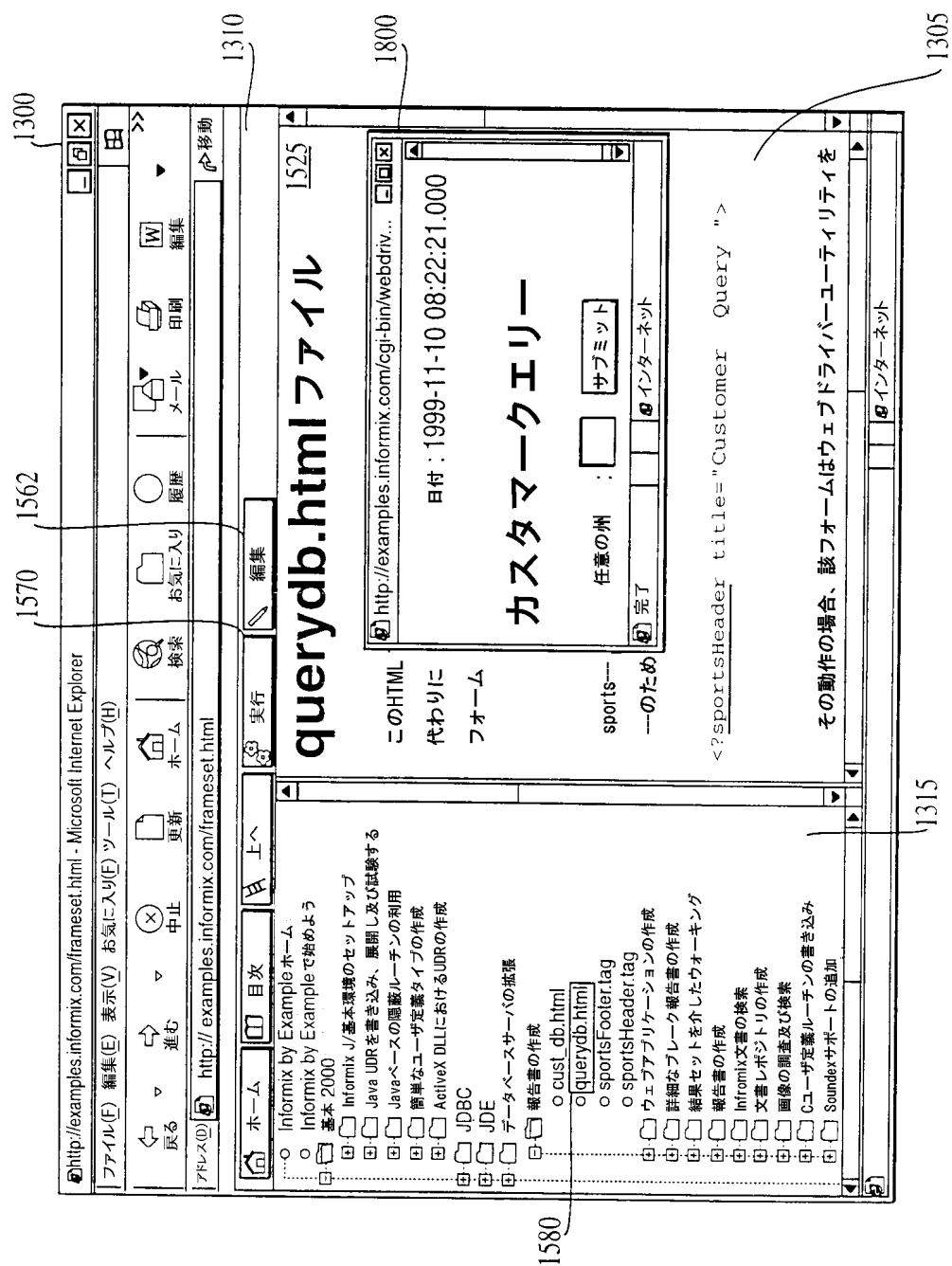


図50

【訳訳訂正44】

【訂正対象書類名】図面

【訂正対象項目名】図51

【訂正方法】変更

【訂正の内容】

【図51】

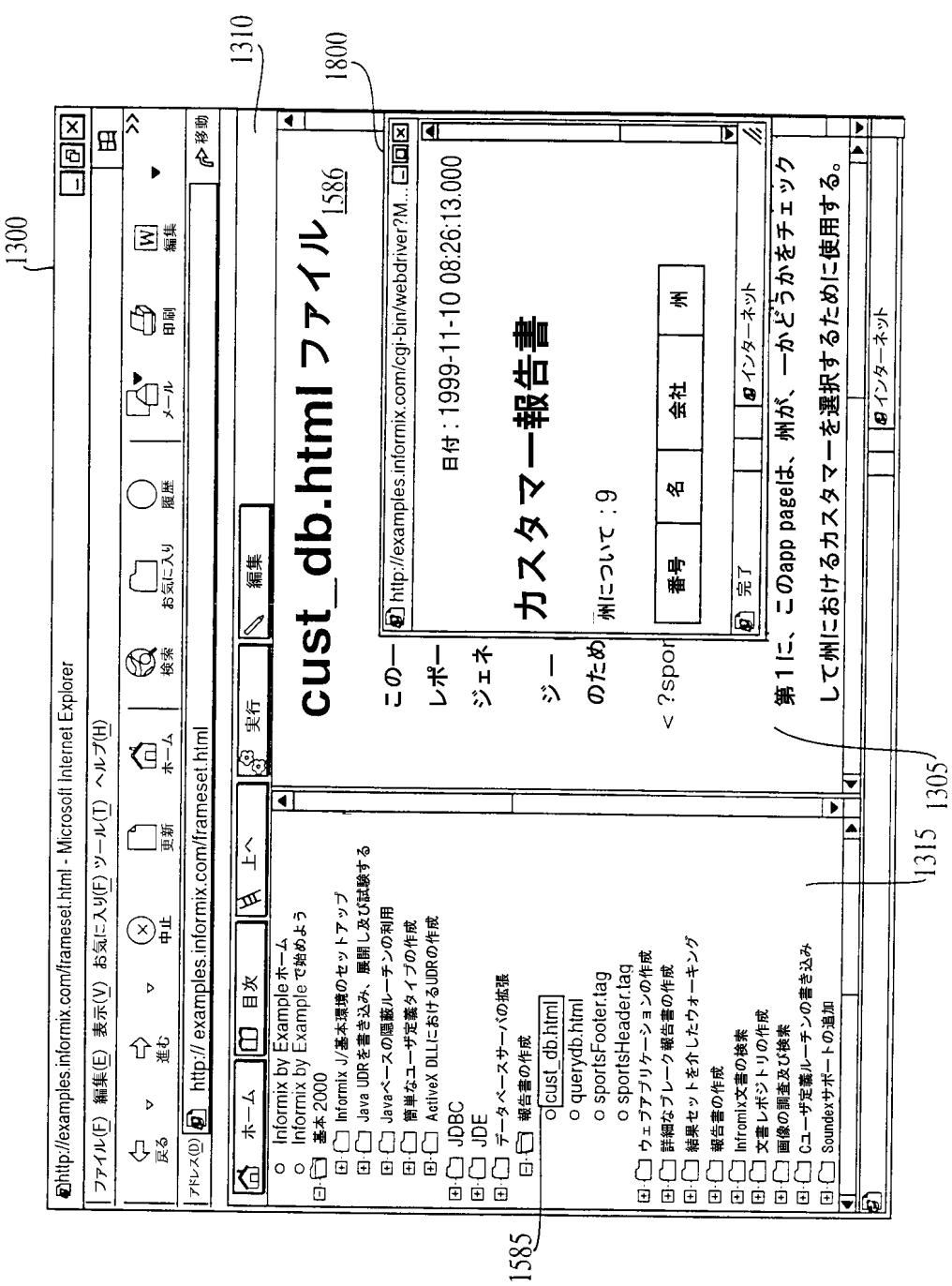


図51

【誤訳訂正45】

【訂正対象書類名】図面

【訂正対象項目名】図52

【訂正方法】変更

【訂正の内容】

## 【図52】

```

1810 {   <!-- <ibyx>
<intro>
<p> <abstract> このapp pageは、クエリを受け入れ、そしてHTML報告書を生成する
      </abstract> 応答において
app pageはダイナミックタグを使用して、HTML報告書についてヘッダー及びフッターを
生成する。
</p>
</intro>
</ibyx> -->
<!-- <ibyx>
<p> sportsHeaderダイナミックタグはHTML真並びに標準オープニングテキストのための
標準ヘッダーを作成する。
</p>
</ibyx> -->
<?sportsHeader title="Customer Report">

<!-- <ibyx>
<p> 第1に、app pageは、州が特定されるかどうかをチェックして、州における
カスタマーを選択するために使用する。
</p>
</ibyx> -->
<?MIVAR NAME=$WHERE_STR><?/MIVAR>
<?MIBLOCK COND="S(AND,S(XST,$selectState) , S (<,0,$ (STRLEN, $selectState)))">
    <?MIVAR NAME= $WHERE_STR>WHERE state= " $selectState"<?/MIVAR>
    <?MIVAR><P>For state: $selectState</P><?/MIVAR>
</?MIBLOCK>

<!-- <ibyx>
<p>次に、app pageは、データを含むだろうテーブルを開始する。
</p>
</ibyx> -->
<P><TABLE BORDER="1">
    <TR>
        <TH> 氏 </TH><TH> 名 </TH><TH> 会社 </TH><TH> 国 </TH>
    </TR>

<!-- <ibyx>
<p> MySQLブロックはカスタマーに質問し、任意に特定された州からのカスタマーのみを
選択する。なぜならば、ブロックのコンテンツはデータの全て列について生成され、
新しいテーブル列は各カスタマーを記述するからである。
&nbsp; HTML エンティティは、改行無しのスペースである。各列における改行無しスペース
を入れることによって、我々はウェブブラウザが値がヌルであるとしてもカラムを表示する
ことを強いる。
</p>
</ibyx> -->
<?MISQL SQL= "SELECT customer_num, fname, lname, company, state FROM customer $WHERE_STR; ">
    <TR>
        <TD> $1&nbsp; </TD><TD>$2&nbsp; $3</TD><TD>$4&nbsp;</TD><TD>$5&nbsp;</TD>
    </TR>
</?MISQL>

</TABLE></P>

<!-- <ibyx>
<p> sportsFooterダイナミックタグは、HTMLページのための標準フッターを生成する。
</p>
</ibyx> -->
<?sportsFooter>

```

図52

## 【誤訳訂正46】

【訂正対象書類名】図面

【訂正対象項目名】図53

【訂正方法】変更

【訂正の内容】

【図53】

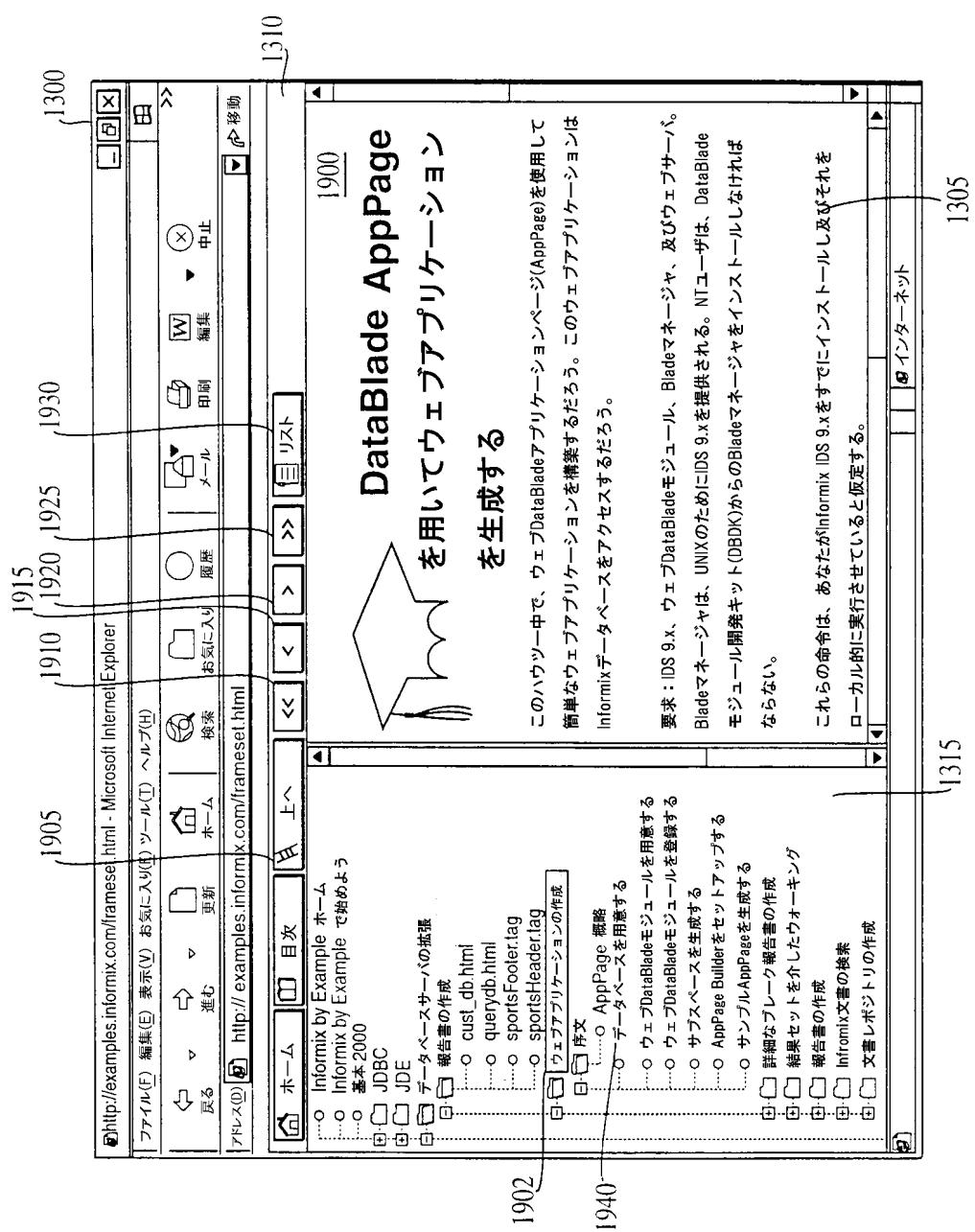


図53

## 【誤訳訂正47】

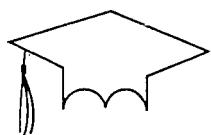
【訂正対象書類名】図面

【訂正対象項目名】図54

【訂正方法】変更

【訂正の内容】

## 【図54】



## ウェブ DataBlade AppPage を用いて ウェブアプリケーションを生成する

1900

1930

このハウツー中で、ウェブDataBladeアプリケーションページ(AppPage)を使用して簡単なウェブアプリケーションを構築するだろう。このウェブアプリケーションはInformixデータベースをアクセスするだろう。

要求：IDS 9.x、ウェブDataBladeモジュール、Bladeマネージャ、及びウェブサーバ。Bladeマネージャは、UNIXのためにIDS 9.xを提供される。NTユーザーは、DataBladeモジュール開発キット(DBDK)からのBladeマネージャをインストールしなければならない。

これらの命令は、あなたがInformix IDS 9.xをすでにインストールし及びそれをローカル的に実行させていると仮定する。

サーバ接続を定義し、かつサンプルデータベースを準備する。

1. サーバ接続を `setnet32 (NT)` で定義する。サンプルデータベースを生成し、又は
  - ▷ データベースを準備する。
 ストア 7 個のデータベースを使用する。

2. ウェブDataBlade 開発環境を準備する。

2. ウェブDataBlade モジュール及び BladeManager をインストールする。
  - ▷ ウェブDataBlade 開発環境を準備する。

3. BladeManager を用いて、デモデータベース内に Web DataBlade モジュールを登録する。

▷ ウェブDataBlade を登録する。

4. gifs のような smart large オブジェクトのための sbspace を生成する。

▷ Smart Blob Space を生成する。

5. あなたのデータベース内に AppPage Builder をインストールする

▷ あなたのデータベース内に AppPage Builder をインストールする

6. あなたのウェブサーバーに AppPage Builder をセットアップする

▷ あなたのウェブサーバーに AppPage Builder をセットアップする

7. サンプル AppPage を生成する

▷ サンプル AppPage を生成する

サンプルアプリケーションを実行する

8. URL を入力 `http:// your_server/scripts/webdriver.exe.`

 このハウツーは、印刷の容易のために2つの別個のファイルにコンパイルされている。基本ファイルはAppPage Builder を用いてウェブアプリケーションを生成するために必要とする全てのステップを含む。第2のファイルはデータベース環境特性を設定し且つ試験するための追加の詳細な命令を含む。

## 図54

## 【誤訳訂正48】

【訂正対象書類名】図面

【訂正対象項目名】図55

【訂正方法】変更

【訂正の内容】

【図 5 5】

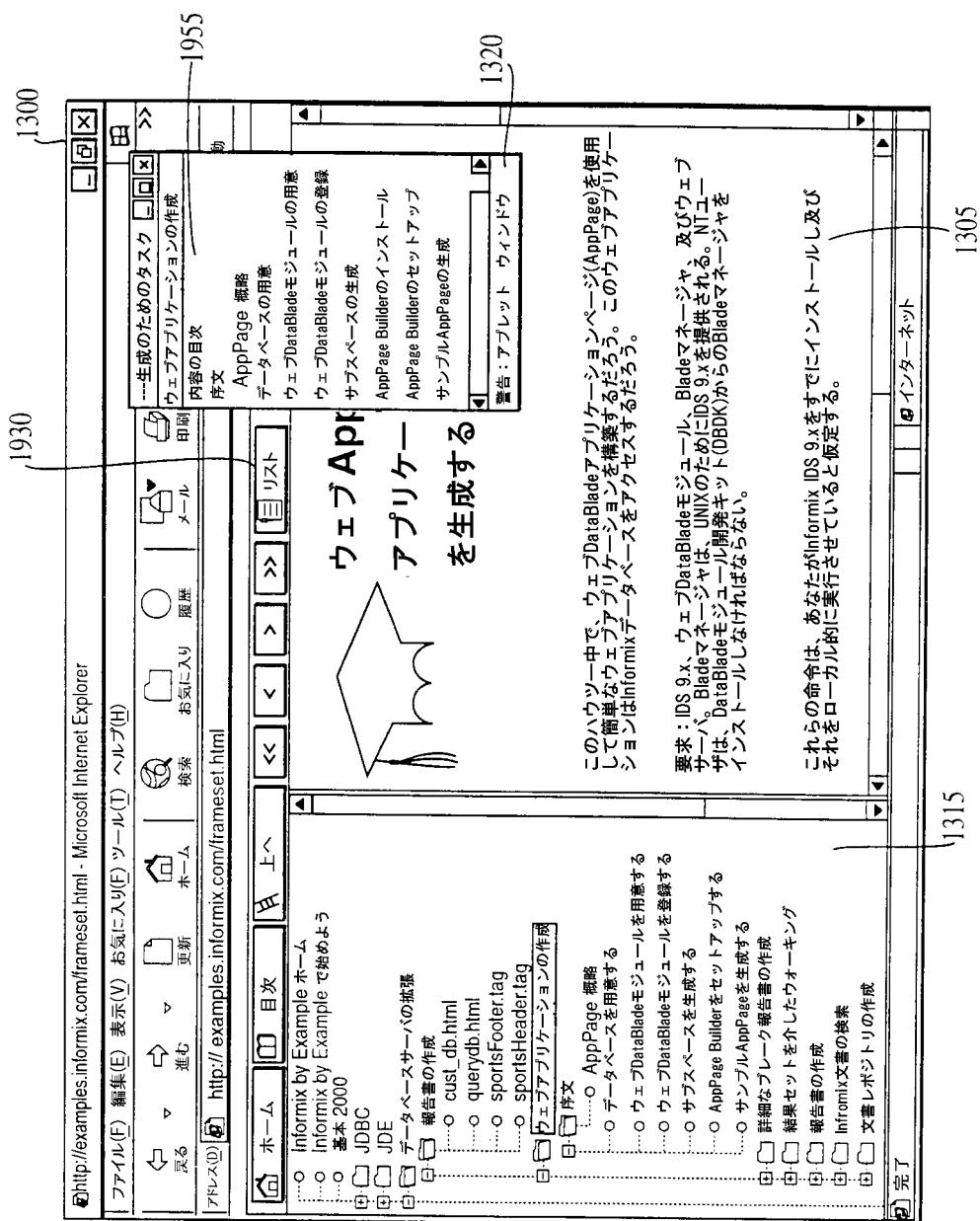


図 5 5

【誤訳訂正 4 9】

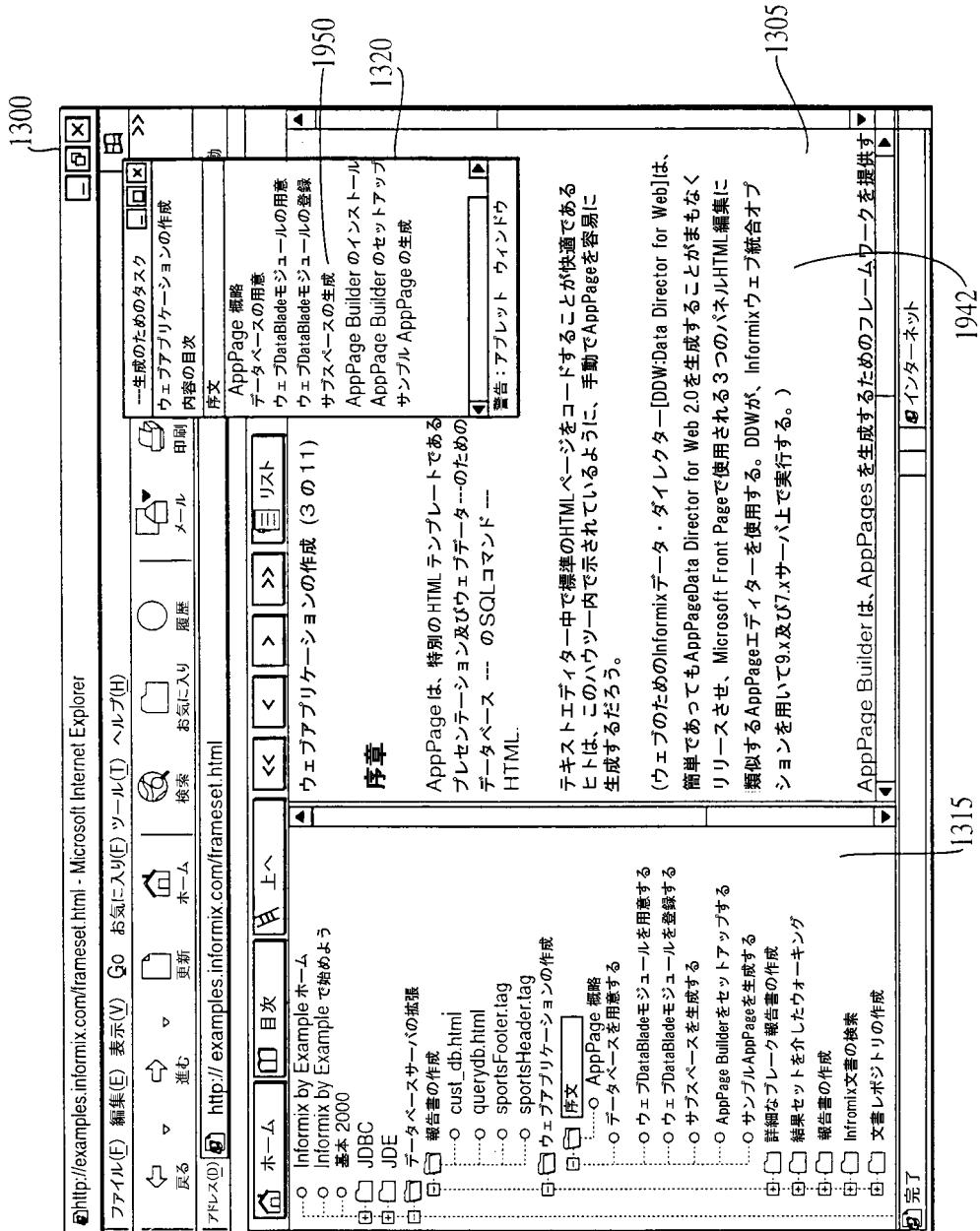
【訂正対象書類名】図面

【訂正対象項目名】図 5 6

【訂正方法】変更

【訂正の内容】

【図56】



## 【誤訳訂正50】

【訂正対象書類名】図面

【訂正対象項目名】図57

【訂正方法】変更

【訂正の内容】

【図57】

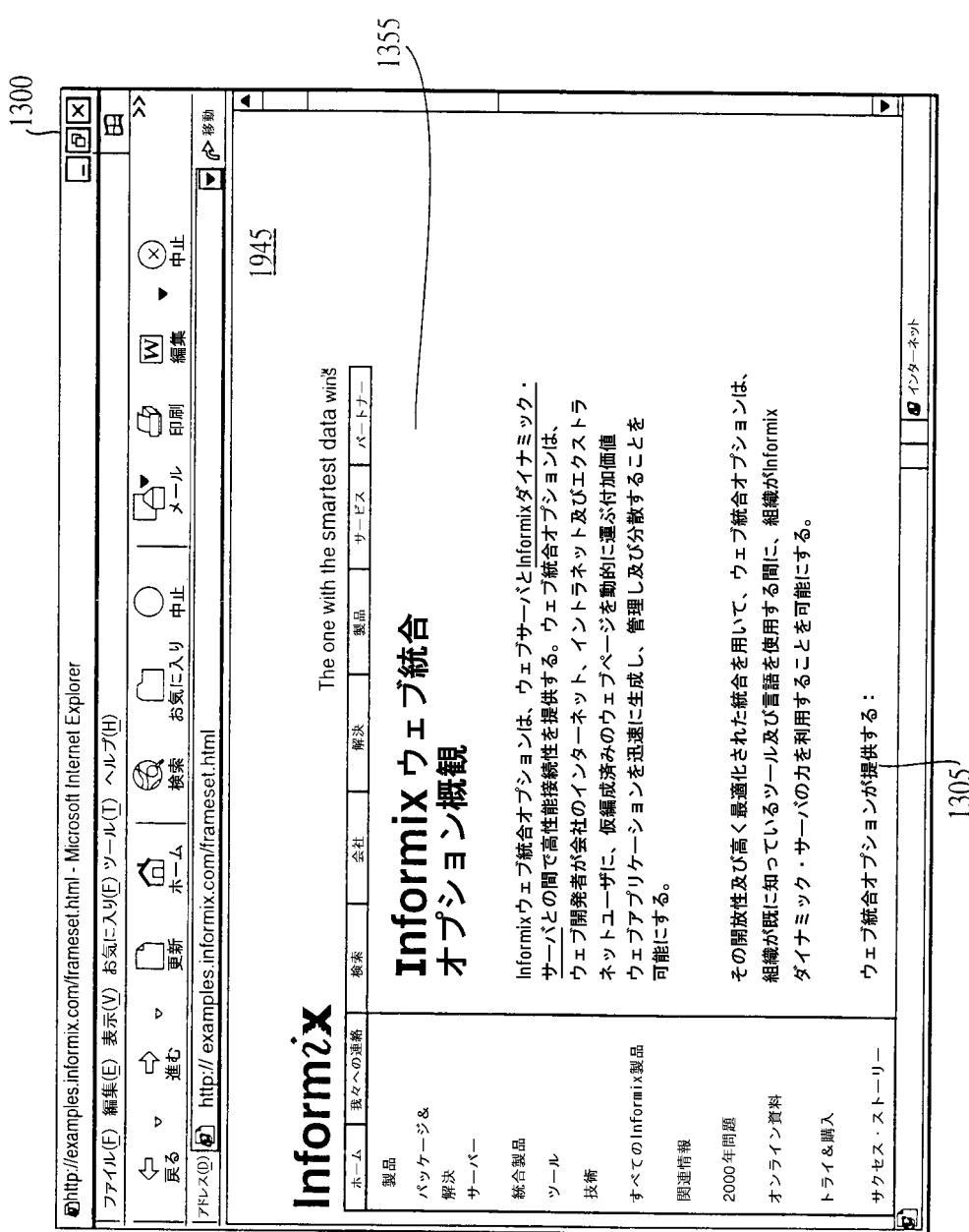


図57

【誤訳訂正51】

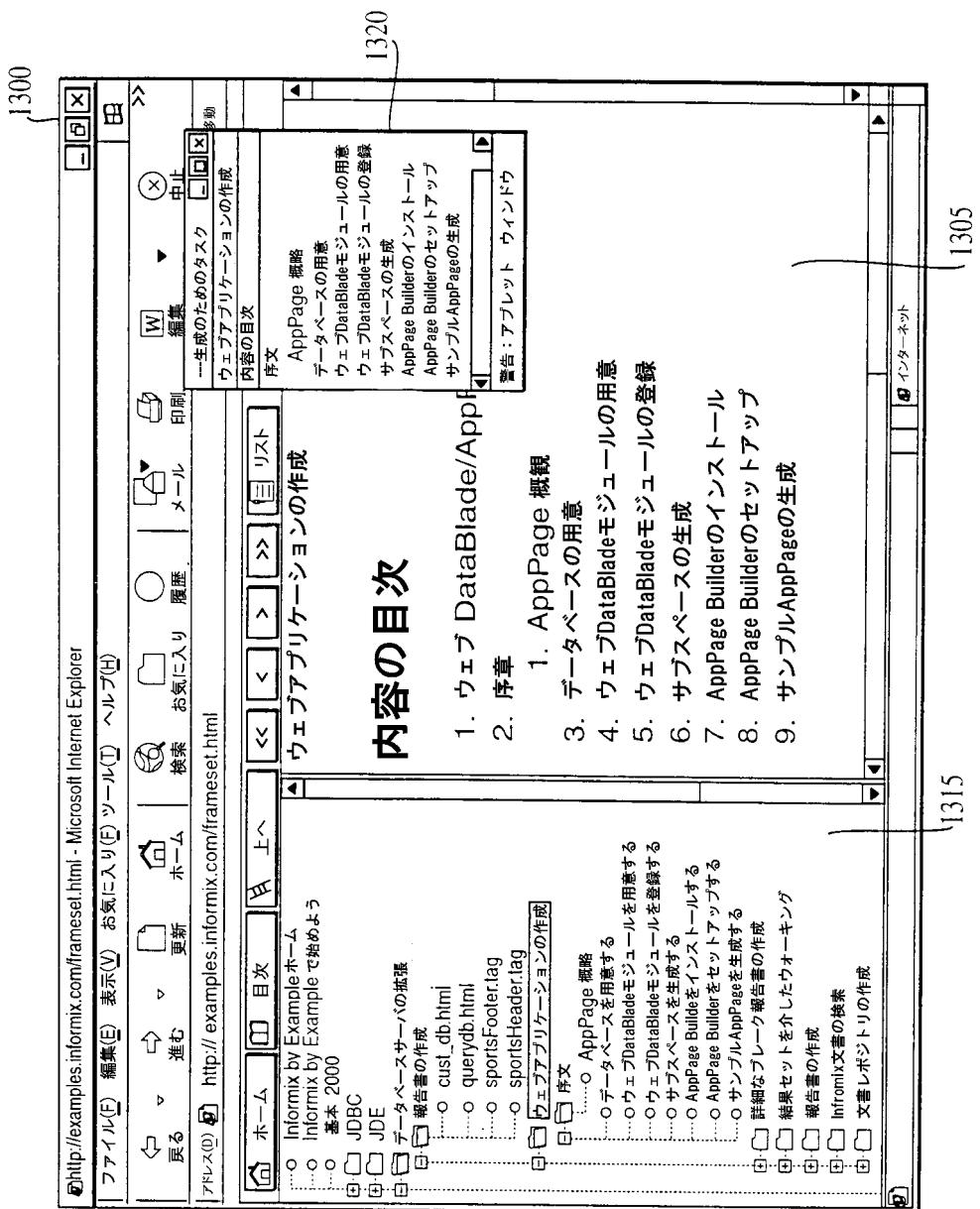
【訂正対象書類名】図面

【訂正対象項目名】図58

【訂正方法】変更

【訂正の内容】

【 図 5 8 】



58