

## ビジネスコンピュータにおける分散GUI化の開発

7M-1

藤田英司、木幡康博、須藤純吾、平井規郎、大江信宏

三菱電機（株） 情報システム研究所

### 1. はじめに

パソコンと当社ビジネスコンピュータをLANで接続し、パソコンのWindows™(\*1)操作の延長でビジネスコンピュータを利用できる操作環境の開発を行った。これまでは文字ベースのダム端末や端末エミュレータから操作を行っていたが、見易く覚えやすい操作画面から、マウス主体の直感的な操作が可能になった。さらに従来のオフコンのホスト端末型処理機能を継承しながら、パソコンをクライアントとするクライアント・サーバ型の処理環境を実現した。本稿ではその特長と実現方式について述べる。

### 2. GUI化の背景

端末装置としてのディスプレイ装置、キーボードは、マンマシンインタフェースを受け持つ装置として親しまれてきた。しかし、これまでは技術的な面から文字ベースのディスプレイ端末が主流であった。パソコンの低価格化、高機能化、Windowsの普及により、GUI (Graphical User Interface) が利用できるようになり、これまでのCUI (Character User Interface) ではできなかった豊かな表現力や多様な入力手段が可能になった。GUIを利用することにより、多様な文字種への対応、イメージの表示、マルチメディアへの対応などにより、より判りやすいマンマシンインタフェースとすることができる。

### 3. 内容・特長

図1に分散GUI化のシステム構成図を示す。

実現した操作環境の特徴を以下に挙げる。

(1) ログイン、プログラム起動、プログラム制御、

Development of Distributed GUI on Business Computers

Eiji Fujita, Yasuhiro Kowata, Jungo Sudo,

Norio Hirai, Nobuhiro Ohe

Mitsubishi Electric Corporation

プリント制御等の一連のOS操作を統合してGUI化した。(2) 既存プログラムメニュー定義を変更することなく、従来のCUIベースメニューをGUI化した。(3) C/S型、ホスト/端末型プログラム及び、バッチ業務などを同一メニューから操作できるようにした。(4) 端末エミュレータ上で動作する従来APPの複数画面同時表示(マルチウィンドウ化)を実現した。(5) Windowsアプリケーションとシームレスな共存を実現した。

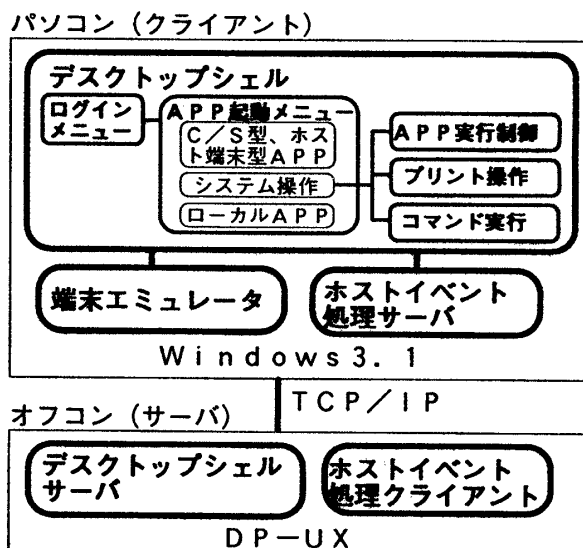


図1 分散GUI化の構成図

### 4. 実現方式

#### 4.1 C/S間の通信にはRPC(\*2)を使用

端末からサーバへの要求はコネクションレスのRPCを使用し、デスクトップシェルサーバがその要求の延長で既存コマンドやライブラリを実行し、その結果を端末へ返す方式をとっている。これによるメリットを以下に挙げる。

(1) サーバ側既存OSのキーボードベースI/F機能に皮をかぶせるだけで実現できる。(2) ユーザからみてサーバの存在を隠蔽できる。(3) コネクションレスなので、要求ごとに接続するオーバ

ヘッドがなくなり、サーバ側の負荷軽減となる。また、端末が遠隔地にある場合など接続時のデータが流れない分ネットワーク料金の軽減にもなる。

(4) サーバOSのオープン化への対応。

#### 4.2 従来APP起動メニューとの互換性への配慮

ユーザの財産である既存アプリケーション起動メニュー（1～99ページにわたる階層化されたメニュー（図2参照））については、そのまま端末側にダウンロードしてデスクトップシェルがウィンドウ化する方式をとった。これにより、以下機能を実現した。

- (1) 既存メニューをまったく変更することなく従来の階層構造を生かしたかたちでGUI化できた。
- (2) アプリケーション属性ごとに異なったアイコンを起動項目に配置し、ビジュアルでわかりやすいメニューに変身させることができた（図3参照）。
- (3) サーバのアプリケーション起動と同期させて端末側のアプリケーションを起動する機能を付加したことにより、従来ではできなかったクライアント・サーバ型のアプリケーション起動も可能となった。
- (4) メニューの更新は、ホスト側のログインユーザごとにあるメニュー定義を既存ユーティリティを用いて変更するだけで、数10～数100台接続されている端末のメニューがログイン時自動的にダウンロードされて更新される。



図2 従来アプリケーション起動メニュー



図3 GUI化されたAPP起動メニュー

#### 4.3 従来業務画面のマルチウィンドウ化

従来端末では画面に1つの業務アプリケーションしか表示できなかったため、複数アプリケーション画面の切替えは、ホスト側でそれぞれの画面情報をディスクにファイルとしてセーブすることによって行っていた。

今回の分散GUI化では、1つの業務アプリケーションごとに別のウィンドウを割付ける方式とした。これにより、画面セーブのオーバーヘッドが削減されサーバ側の負荷軽減及び、画面切替え性能向上を実現しただけでなく、複数業務画面の同時表示及び、それぞれの画面タイトルに業務名が入るなどの画面装飾を可能にした。また、業務画面を表示しながらマルチウィンドウで業務選択ウィンドウから他の業務画面を選択したり、アプリケーション実行制御等が可能となった。

#### 5. あとがき

以上のように、従来ビジネスコンピュータ上に画期的な分散GUIを構築することができた。今後はエンドユーザ対応のカスタマイズ機能強化や次期パソコンへの対応が課題となる。

(\*1) Windows™は、米国Microsoft Corporationの登録商標

(\*2) RPC: Remote Procedure Call