

次世代統合オフィスシステム“アラジンII”

1G-7

オフィス業務の構築環境

若林 宏 杉田 浩二 北野 千春
日本電気(株) 情報処理システム技術本部

1. はじめに

企業の中では、組織運営のために、様々な業務が各オフィスで遂行されている。これらの中には、現在、ホストコンピュータを使ってEDP化した基幹業務システムや、パーソナルコンピュータを利用した個別OAツール(ワードプロセッサ・表計算ソフト等)により、効率化が進められているものがある。

一方、オフィスには、EDP部門のバックログとしてEDP化から取り残されている業務や、EDP化するには対投資効果から難しい業務、汎用的な個別OAツールでは処理しきれない業務が多々あり、効率化の恩恵に預かっているオフィス業務は、まだ、少ないといえる。

このような中で、90年代に入り、企業を取りまく環境はいつそう厳しく、時々刻々と変化する状況の中で、戦略的かつ迅速な組織運営が必要とされ、オフィス業務の電子化による統合化・効率化が急務となってきている。

このため、90年代におけるオフィスオートメーションは様々なオフィスで遂行される業務を、いかに簡単にシステム化していくことができるかが鍵となろう。

本稿では、次世代統合オフィスシステム“アラジンII”の“オフィス業務構築環境”として開発したオフィス伝票システムを通して、特に、一般のオフィスで日常頻繁に行われる事務処理を中心としたオフィス業務の電子化について、その要件、具備すべき機能、実現方式、適用例について述べる。

2. オフィス業務とその電子化

2.1 オフィス業務の特性

一般のオフィスで行われている事務処理を中心としたオフィス業務には、以下の特性がある。

- ① 企業またはオフィスごとに、遂行する業務内容が異なる。
- ② 業務の遂行過程において、人の判断を必要とする場合が多い。
- ③ オフィスを取りまく環境により、業務内容の変化が短期的に発生しやすい。

さらに、オフィス業務を遂行する担当者は、コンピュータに慣れていない場合が多く、また、従来からの業務形態から大幅に変化することを望んでいない。

2.2 電子化するための要件

上述のような特性をもった、業務および業務遂行担当者を対象として、業務の電子化を進めるためには、以下の要件を満たす必要がある。

- ① 企業またはオフィスの業務内容に合ったシステムが、容易に構築できること。
- ② 定型作業はコンピュータに任せ、人間の判断が入るところは、対話的に人間が容易に処理できること。
- ③ 業務内容に変更が入る場合、短時間で容易にシステムの変更ができること。
- ④ 利用者インタフェースを利用者の業務形態に合わせ、きめ細かく容易に構築できること。

3. オフィス伝票システムの機能と特徴

3.1 オフィス伝票システムの機能

オフィス伝票システムは、対話的に帳票のフォーマットを設計する機能と、帳票に記入されたデータの処理手順をWSとホストに分散して記述できる機能を提供することで、利用者インタフェースのやさしい業務システムを容易に構築・変更できる環境を提供する。

このオフィス伝票システムの主な機能について、以下に記述する。

Integrated Office System II
Environment for the Construction
of Office Applications
Hiroshi WAKABAYASHI, Koji SUGITA, Chiharu KITANO
EDP Systems Engineering Division, NEC Corporation

(1) 帳票設計

- ① マウスを利用し、画面を見ながら自由なレイアウトで帳票の設計ができる。
- ② 起票を容易にするために、電子伝票ならではの指定ができる。
 - 帳票ごとのメニューバーの設定
 - 副画面表示、副画面での入力値選択
 - ボタン項目の設定
 - 自動記入、自動演算、入力値の自動判定
 - 入力順序指示、記入禁止項目
 - 文字色、反転
 - 終了時の処理(送付、保存等)
 - アプリケーションとの連携
 - メッセージ表示
 - 音を鳴らす、音楽を鳴らす

③ 帳票を台帳に登録するだけで、帳票の一元管理が図られ、いつでも最新の帳票が利用できる。

(2) 伝票処理記述

- ① 起票時に、きめ細かなサービスができるようにするため、WS・ホストそれぞれに、伝票処理記述言語で処理を定義したスクリプトを作成することができる。このスクリプトは、以下の契機で起動することができる。
 - 項目にデータが入力されたとき
 - 項目がマウスでクリックされたとき
 - ファンクションキーが押されたとき
 - メニューが選ばれたとき
 - 帳票を表示したとき
 - 帳票の表示を終了するとき
 - 常時

② 伝票処理記述には、帳票の取り出し・表示・削除・項目への値の設定等、標準的な伝票処理が記述できる。また、標準機能にない処理は、利用者プログラムを呼び出すことで、実現できる。

表1に、伝票処理記述コマンドの機能概要を示す。

表1 伝票処理記述コマンド一覧

項目	概要
帳票の処理	<ul style="list-style-type: none"> • 未記入帳票の取り出し • 帳票の複写・削除 • 承認 • 帳票の印刷
帳票項目の処理	<ul style="list-style-type: none"> • 項目の転記 • 項目の保護・非保護
台帳の処理	<ul style="list-style-type: none"> • 台帳への記帳 • 台帳記載内容の変更 • 台帳の条件検索
電子メール・ファイル	<ul style="list-style-type: none"> • メール発信・受信 • ファイル保管・取り出し
表示状態の変更	<ul style="list-style-type: none"> • 表示ページの変更 • 表示位置・カーソル位置の変更 • 副ウィンドウ表示
プログラム連携	<ul style="list-style-type: none"> • WSプログラムの呼び出し • ホストプログラムの呼び出し

3.2 オフィス伝票システムの特徴

オフィス伝票システムには、上述した機能を具備することで、以下のような特徴がある。

- (1) きめ細かな利用者インタフェースの構築が容易
従来ホストを中心とした方式では、利用者インタフェースをきめ細かく制御しようとするとトランザクション量が増えレスポンスに影響を与えるといった問題があった。

オフィス伝票システムでは、入力データ・形式情報・伝票処理記述を分離しており、また、伝票処理記述についてはWSとホストに分けて、記述できるようにした。これにより、WS側でより、WS側できめ細かな利用者インタフェースを実現することができる。

- (2) プロトタイプリングが容易
 オフィス業務のシステム化に当たり、早い時期に利用者に操作イメージを提示し評価することは、使いやすいシステムをつくるための大切な要件となる。
 オフィス伝票システムでは、ホスト側の処理と独立に利用者インタフェースの設計・開発ができるので、利用者システムイメージを提示しながら段階的に開発をすることができる。
- (3) 利用者APの構築が容易
 WSとホストそれぞれの機能・資源を十分に生かすには、WS側とホスト側それぞれ合った処理を実行するように、分散化を図ることが必要である。
 オフィス伝票システムでは、ホスト・WSそれぞれに、伝票処理記述言語を使用した利用者APを、容易に組み込むことができる。これにより、処理の最適分散が図れ、ホストの負荷軽減が実現できる。
- (4) 帳票の管理と配布が容易
 WSとホストとの連携処理においては、両方にプログラムを開発する必要がある。これを運用していく場合には、従来、すべてのWSに開発したプログラムを配布したり、WSとホストとの、プログラムバージョンの同期を合わせたりしなければならず、運用が難しくなる。
 オフィス伝票システムでは、WS側で作成した利用者APは、設計した帳票と一緒に、ホストの帳票管理キャビネットに保管し、必要に応じてWSに自動的に配布する。また、版数管理を行っているので、帳票を改版するときにも帳票管理キャビネットの帳票を置換するだけで、それ以降、新しい帳票の利用、および利用者APの実行ができる。

4. オフィス業務の実装方式

オフィス伝票システムでは、オフィス業務を実現するための利用者APをWS側とホスト側にそれぞれ実装することができる。この利用者APは、図1に示すようにWSとホストそれぞれに実装された、伝票処理記述制御部により、実行される。この利用者APの実装方式を、以下に述べる。

- (1) 処理手順の作成
 処理手順は、WS・ホスト共、伝票処理記述言語にてスクリプトを作成する。
 作成したスクリプトは、実行速度を上げるため、パーサにて処理手順オブジェクトに変換し、処理手順オブジェクトファイルに格納する。
- (2) 処理手順オブジェクトと帳票形式の格納場所
 ホスト側で作成した処理手順オブジェクトは、そのまま処理手順オブジェクトファイルに格納する。
 WSの処理手順オブジェクトは、帳票の形式情報と一緒にホストの帳票管理キャビネットに格納する。
 WSにてこれを利用する場合には、ホストの帳票管理キャビネットから、処理対象となる帳票を取り出すことにより、帳票形式はWSの帳票形式ファイルに、処理手順オブジェクトは処理手順オブジェクトファイルに、自動的に取り出される。
 帳票形式については、起票中は、WSとホストの両方に存在し、帳票IDと版数をキーに帳票データの受け渡しを行う。
- (3) WS-ホスト間のデータの受け渡し
 WSからホストに処理を渡すときには、WSで起票した帳票データに対し、以下に示すような情報を付加し、送信する。
 - ホストで実行する処理手順の名前
 - WSスクリプト起動契機
 - WS起動スクリプト名
 - カーソル位置項目
 - 表示ページ
 - 送信ページ数
 - 全ページ数
 - 利用者ID
 ホストでは、WSから渡された帳票データの中の帳票IDをもとに、ホストの帳票形式キャビネットから帳票形式を取り出し、この形式情報とWSから送られてきた帳票データ・付加情報で、処理を行う。
 処理を終えたら、WSに帳票データおよび付加情報を書き換えたものを、返却する。

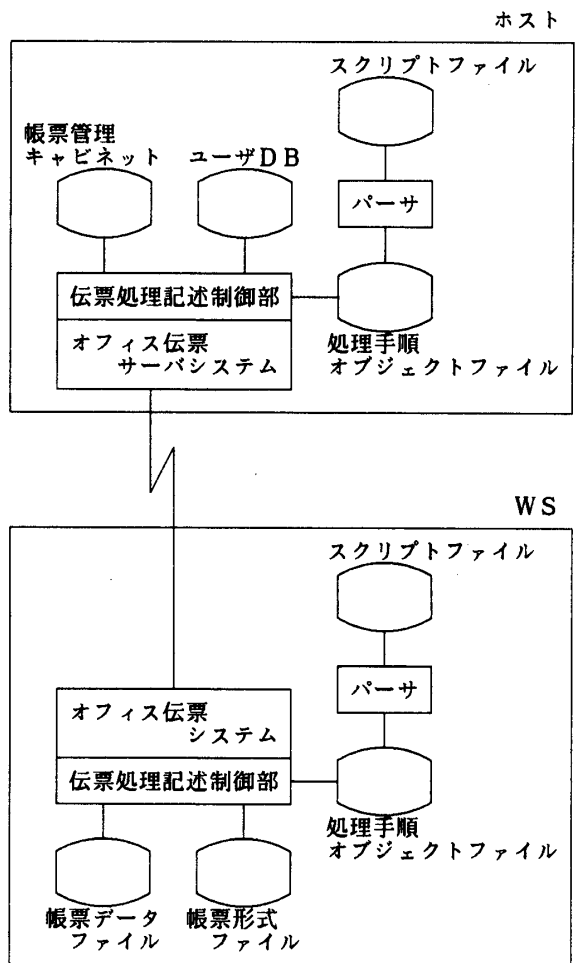


図1 伝票処理記述制御部の実装図

5. 適用例と効果

当社の新本社システムで、オフィス伝票システムを用いて構築した、3つのオフィス業務について、従来のCOBOL言語を用いた見積り規模と、伝票処理記述言語を用いて記述した開発規模の実績の比較を表2に示す。
 WSとホストの伝票処理記述言語の総ステップ数をCOBOL言語の総ステップ数と比較すると、約1/3のステップ数で開発ができる。
 また、COBOL言語に比べ、伝票処理記述言語は、やさしい利用者インタフェースが容易に実現できる。

表2 開発規模の比較 (単位:KLine)

業務名	COBOL	伝票処理記述
電子電話帳システム	52.7	15.0
スケジュール管理システム	22.5	6.4
会議室予約システム	9.0	5.5

注) 見積りは、ファンクションポイント法による

6. おわりに

今後ますます企業が大きくなったり、組織が複雑になるにつれ、オフィスにおける業務も複雑さを増してくるものと思われる。
 このような、複雑なオフィス業務を構築するためには、今後、複数ホストに処理を分散するような業務を構築するための“オフィス業務構築環境”が必要と思われる。