

メニュー操作履歴のマクロ化機能を有する OA用コマンドインタプリタの作成について

高田 久靖 鈴木 達郎 加藤 秀一

NTT 横須賀電気通信研究所

1.はじめに

OAの発展の中で、オフィスユースのマンマシンインタフェース(以下、MMIと略記する)を志向した様々な個人用ワークステーションが開発され、種々のサービス項目が提供されている。これらのサービスは、全ユーザに共通に使用される。しかし、OA環境において、共通のサービス項目だけで全てのユーザを満足させる事は不可能である。エンドユーザが一人一人、別の人間であるように、各ワークステーションは個人に最も適合した環境を提供する必要がある。

本稿では、OA用マルチウィンドウシステムで実現されるべきユーザ個人環境の設定機能(①処理手順省略のためのデフォルトパラメータ設定機能、②コマンド言語プログラミングを用いないマクロ定義機能、③AP間でのパラメータ引き渡し用アイコンを用いたサービスAPの連続的起動機能)の方式検討と実現法について述べる。

2.OA用システムにおける個人環境設定について

OA用システムにおけるMMIに対する考え方として、直接操作性(Direct Manipulation,[1])という概念が提唱されている。この概念では、以下の事が特に重要視される。

- ⑦直観に合致した容易な操作が可能
- ⑧日常の作業環境との類似性が成立すること

ここで、オフィスにおける日常の作業環境とは、狭い意味で、個人の作業の仕方として捕らえる事ができる。

そこで人間のオフィスでの作業の仕方考えた場合、それには以下の4段階が考えられる。

- (i).個別作業を決められた手順で実行する。
- (ii).個別作業を自分なりのやり方/手順で行う。(個別サービスの操作手順の省略/指定)
- (iii).複数の個別作業を組み合わせ、状況に応じた作業を行う。(ユーザ定義サービスの作成)
- (iv).作業間でデータや処理結果を受け渡し、有機的に複数の個別作業を組み合わせ、実行する。(作業の連続性)

ここで、(iv)の状態が一般事務職員の熟練レベル、すなわちオフィスにおける個人の作業の仕方と考えられる。オフィス内作業に対応するOAサービスに対して、以上の状況との類似性が存在すれば、ユーザにとり、そのOAサービスは馴染み易いと考えられる。

特に、(ii),(iii),(iv)が複合された状態において、ユーザの個人環境の設定が最も柔軟かつ強力に行えると考えられる。現在、オフィスに存在するOAサービスは単体サービスが中心であり、その個人環境は(i)の状態にある。

オフィスにおける作業の仕方との類似性から、(ii),(iii),(iv)の状態をユーザに提供するために、OAシステムが持つべき機能は以下の3項目目からなると思われる。

(a).個別サービスの操作手順の省略/指定機能――

サービスAPパラメータのユーザ定義機能。以下、これをデフォルトパラメータ設定機能と呼ぶ。

(b).ユーザ定義サービスの作成機能――

基本サービスAPからマクロコマンド/プロシージャを定義/実行する機能。但し、専門的知識を必要とする従来のコマンド言語によるマクロ定義は行わない。エンドユーザに相応しい簡易なものにする。一般に(a)のデフォルトパラメータ設定機能は、このマクロ定義機能を用いて実現できる。

(c).サービスに連続性を持たせる機能――

直前のサービスAPの処理結果(データ)を、次のサービスAPの入力(データ、パラメータ)として引き渡す事を促進する機能。これはサービスAP間のパラメータ引き渡しであり、ユーザ指定により逐次行われる場合と、自動的に行われる場合がある(前段APの入力及び後段APの出力指定パラメータを一致させるとデータ引き渡しになる)。ユーザ環境設定機能の目標は、自動的なパラメータ引き渡し機能とマクロ機能が結合した状態であると考えられる。

これらの機能は、キャラクタベースのシステムでは既実現されているが、マルチウィンドウ、メニュー、そしてアイコンを備えたシステムでの実現は必ずしも容易ではない。今回の報告では、マクロのマルチウィンドウシステム

上での実現形態((a),(b))及びユーザによるサービスAP間のパラメータ引き渡しを促進させる機能((c))について、述べる。

3.個人環境の設定機能

本章では、ユーザの個人環境設定を行うための機能について検討する。

3.1.実現環境について

ユーザの個人環境設定機能を考察する前に、その実現環境について述べる。(図1参照)

(1).マルチプロセスとマルチウィンドウ([2])

事務作業環境との類似性から、ユーザには複数のサービスAPの同時実行機能(マルチプロセス機能)を提供する。複数のサービスは複数の画面を必要とし、かつ、それらの画面が同時表示される必要がある。そこで単一の高解像度ディスプレイ画面を、サービス毎に複数の画面(ウィンドウ)に分ける。

(2).メニューとアイコン

メニューは複数の選択枝を表示し、ユーザに選択させる機構であり、MMIとして望ましい以下の特徴を有する。

- (i)ユーザは『何をすべきか/何ができるか』を容易に理解できる。
- (ii)すべきでない事は表示されないので、間違いにくい。

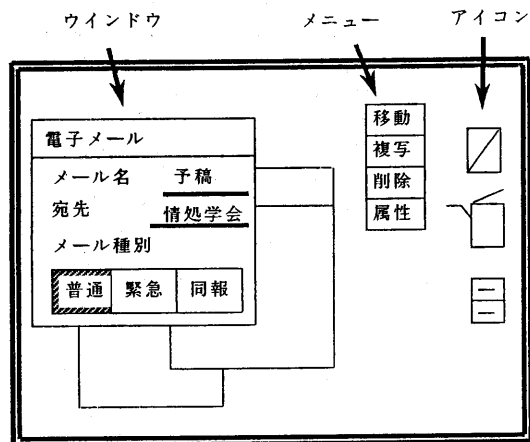


図1 マルチウィンドウ表示画面

アイコンは直観的に資源を表現するために用いられる。アイコンの形状の例としては、OAユースでは以下の物が考えられる。

- (i).印刷(印刷サービス)→電子コピー機
- (ii).メール(電子メールサービス)→メールボックス
- (iii).文書ファイル(文書作成サービス)→文書紙
- (iv).ファイル(電子ファイルサービス)→フォルダ、キャビネット

ここで、(i),(ii)はサービスAPを、(iii),(iv)はデータファイルを表現している。前者をサービスアイコン、後者をデータアイコンと呼ぶ。

サービスアイコンは、なるべく小さな機能単位で作成すると、モードの多階層化に伴う複雑さを減少させる事ができる。

アイコンの存在により、以下の考察ではポインティングデバイスのサポートを前提とする。

(3).フォーム

メニュー、アイコンでは表現が困難な、数値、名前等のデータのキー入力用ウィンドウを用意する。これをフォームウィンドウと呼ぶ。

(4).デスクトップ

各種のアイコンを表示しておくウィンドウを用意する。それは事務所又は事務机を表現するものと考えられる。これがデスクトップであり、ユーザの作業環境でもある。デスクトップは複数個存在し、かつ個人が複数のデスクトップを所

有する事も許す。これは、ユーザの事務机が複数ある事を直観的にユーザに示す。

3.2.OAにおける個人環境の設定機能について

2章で述べた、個人環境の設定機能について、より詳細に述べる。

(1).マクロ定義について

従来、エンドユーザ向け簡易マクロとして良く用いられてきた方式には、大別して以下の2種類がある。

- ①キーストロークのマクロ化
- ②キー入力ストリームのコピー/再入力

①は、単一サービス(e.g.エディタ)内での局所的なマクロ定義機能として、使用されてきた。例えば、ユーザ定義により、幾つかのエディタコマンドを単一のキーに割り当てる機能がそれに当たる[3]。②は、サービス実行時に、あたかもユーザがコマンド投入を行っているかのように、サービスAPに対しエンドユーザ入力をシミュレートする。この方式はサービスAPが特別な機能を持つ必要がないので、インタラクティブなサービスに対して一般的に適用可能である。

これら既存の方式の成立要因は、エンドユーザからの入力がキーストローク入力のみである事、サービスAPが入力対象とする端末は常に物理端末と同一である事、端末入力とファイル入力の切り替えが可能である事である。

しかし、前提とするシステムは、マルチウィンドウ機能を有し、2種類の入力メディア(キー入力とポインティングデバイス入力)をサポートする。このポインティングデバイス入力は基本的に入力端末(ウィンドウ)からの論理画面座標データ入力である。これはマクロ定義/実行、双方の場合に必要な。しかし、サービスAPは複数のウィンドウを使用し、かつ、これらを動的に生成/消滅するため、マクロ定義時には実行時に入力すべきウィンドウを指定できない。また、ポインティングデバイス入力を、ファイル入力によりシミュレートする事は困難である。従って、方式②の実現は、特に困難である。

(2).サービスAP間でのパラメータ引き渡し機能について

既存システムにおける自動的なAP間パラメータ引き渡し機能はキャラクタ列のような単純なデータ構造を前提としていた。しかし、ポインティングデータ等のメディアの異なるデータも対象とする場合、AP間での自動的なパラメータ引き渡し機能の一般的な実現は困難になる。そこで、今回はユーザによる逐次的なパラメータ引き渡しの支援機能のみを考える。この時AP及びユーザ双方にとって、扱い易いファイルを対象にしたパラメータ引き渡しを考える。

そこで、サービスの処理結果として作成されたデータファイル(実体、情報)をデータアイコンとして表現する事により、サービスAP間でのユーザによる逐次的なパラメータ引き渡しを支援する。

特に、サービスの処理結果として表示されるデータファイル情報をアイコン化する事とし、これを以後、項目アイコンと呼ぶ。これにより、例えば、ファイル検索サービスにより表示された項目アイコンを、次の印刷サービスに対する出力ファイル指定に使用できる。この機能もシングルプロセス環境では既に存在するが、マ

ルチプロセス環境での実現は困難である。

3.3.メニュー/アイコンを用いたマクロ定義及びデフォルトパラメータ設定方式の提案

本節では、基本サービスAPからマクロコマンド/プロシージャを定義/実行する機能([4])及び、サービスAPのデフォルトパラメータ設定機能について、OAユーザに相応しい方式を考察する。

(1).マクロ定義

通常、エンドユーザに提供されるサービスは単機能である。しかし、システムに熟練するに従って、エンドユーザは各サービスの複合を望むようになる。例えば、メール用送信文書を作成し、宛て先に送付すると同時に、ハードコピーをとり、フォルダ内に送信文書を保存する等である。この例では、電子メールサービス、文書処理サービス、電子ファイルサービス、印刷サービスが複合されている。このようなサービスの複合は、マクロプロシージャにより行なう。

(a).マクロ定義方式に対する要求条件

マクロコマンドの定義方式には次の要求条件がある。

①エンドユーザが作るので、コマンド言語を使用せずに、メニュー等で(すなわち、3.1で述べた実現環境で)定義させたい。

②マクロのサポートにより、サービスAP等のプログラム構造を複雑にする事は避ける。

要求条件①により、メニュー等でマクロコマンド定義を行うとすると、マクロ定義プログラムを作る必要がある。しかし、パラメータは各サービスAP内で集められるため、マクロ定義プログラムはサービス個別になり、プログラム作成が困難である。従って、マクロコマンドの定義は各サービスAPの機能を使用するが、共通的に外部で行うべきである。この時、要求条件②を考慮する必要がある。

(b).マクロ定義モードの導入

上で述べた要求条件を満足するために、以下の方式を提案する。

サービス実行以前にパラメータの設定のみを行うモードをサービスAPに与える。これを以後、マクロ定義モードと呼ぶ。また、サービスを実際に実行するモードを実行モードと呼ぶ。

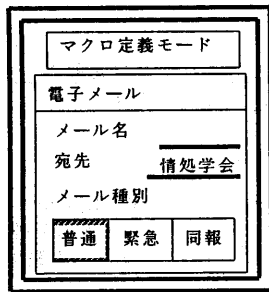


図2 マクロコマンドの定義

マクロ定義モードで起動されたサービスAPは、サービス実行時と同じ画面を表示し、ユーザにパラメータ設定を要求する。この際、ユーザは予め指定しておきたいパラメータを指定する。この時、全てのパラメータを指定する必要はない。指定しなかったパラメータは、そのサービスの実行時に(すなわち、実行モードでのサービスAPの起動時に)指定する(図2参照)。この2つのモードを導入し、サービス実行時とマクロ定義時で画面を共通化する事により、サービスAPに付加的にマクロ定義機能を組み込み、プログラム構造が単純化できる。

マクロ定義時に指定されたパラメータは、そのサービスAP名と共に、サービスAPによりファイルに出力される。このように出力された情報は、コマンドインタプリタに対して与えることにより、マクロコマンドになる。かつ、複数のマクロコマンド情報を同一ファイルに連続的に出力することにより、複数サービスを連続実行するマクロプロシージャが定義可能になる。

このように定義されたマクロコマンド/プロシージャも、アイコンとして表現する。これにより、マクロコマンド/プロシージャをサービスアイコンと同様に扱い、かつ階層的なマクロコマンド/プロシージャの定義を可能にする。

(c).提案方式について

今回、キーストロークのマクロ化方式の発展形として、マウスボタンクリックの解釈情報を蓄積し、マクロコマンドを定義する方式を提案した。キーストロークのマクロ化方式を採用したOA向けシステムは既に存在するが([5],[6])、ポインティングデバイス入力可能なマルチウィンドウシステムで採用したものはあまり知られていない。

(2).デフォルトパラメータ設定

2(b)で述べたようにデフォルトパラメータ設定機能はマクロ定義機能の一部と考えられる。従って、デフォルトパラメータ設定操作はマクロ定義操作と同様にすることが望ましい。

4.実現法

本章では、前章までに述べた機能の内、マクロ定義/実行機能,サービスのデフォルトパラメータ設定機能と項目アイコンの実現法について述べる。

4.1.マクロ定義の実現法

3.3(1)で述べた方式を実現するため、サービスAPの構成法について検討する。

サービスAPは、構成上、2つの部分に分かれる。すなわち、メニュー等を介してユーザとインタラクションを行い、パラメータを取り込む部分(以後、ユーザインタフェース部と呼ぶ)とそれをもとにして、実際のサービスを実行する部分(以後、実行部と呼ぶ)である。

実行モードでは、サービスAPはユーザインタフェース部、実行部の順で走行する(図3参照)。

それに対し、マクロ定義モードではユーザインタフェース部のみ走行し、取り込まれたパラメータを適当な形式に変換して、出力する(図4参照)。このマクロ定義モードのみで必要な機能はユーザインタフェース部に持たせるのが自然である。マクロコマンドの実行形態を図5に示す。

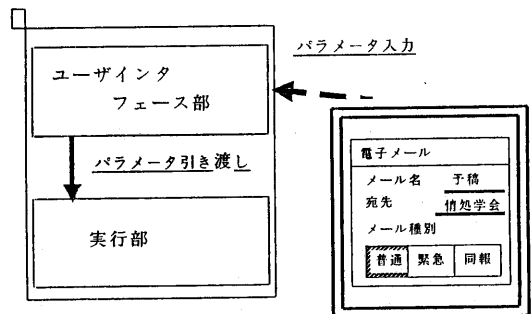


図3 サービスAPの構造

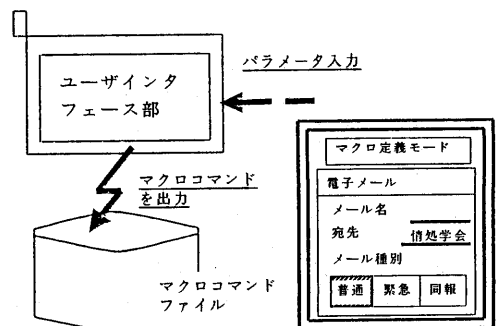


図4 マクロ定義モードの動作

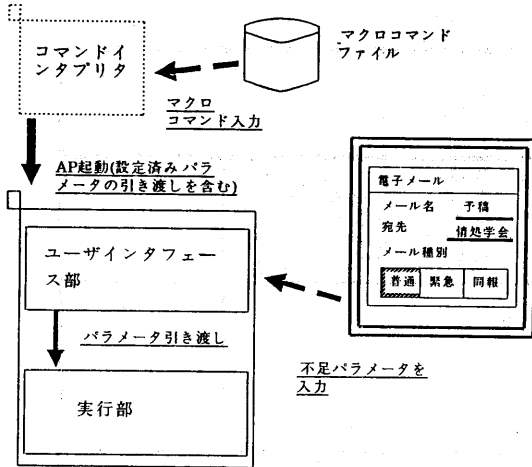


図5 マクロコマンドの実行

上記の様に考えると、ユーザインタフェース部と実行部は、相互に独立したモジュールとして構成される。そこで、以下の3つの利点も考慮して、ユーザインタフェース部と実行部を別のソース/実行プログラムとして構成する。

- ①ユーザインタフェース部と実行部が独立して走行するため、サービス走行のために必要なメモリ空間が少なくすむ。
- ②メニュー画面等を変更する際にユーザインタフェース部を改造するだけで良い。
- ③実行部はコマンドとして単独で走行可能。

4.2. サービスAPに対するデフォルトパラメータ設定

サービスアイコンのデフォルトパラメータ設定は、個別のアイコン単位で行われる。そこで、各アイコンのパラメータ情報を、各サービスアイコン単位にファイルを作成し、それに格納する。このファイルはコマンドファイルになる。コマンドファイルの移動/複写により、任意のキャビネット、フォルダ、デスクトップ内のにまたがるサービスアイコンの移動/複写が自然に実現できる。

各サービスアイコンが表現するサービスAPのデフォルトパラメータ設定は、サービスの実行時と同じユーザ操作により行わせる。このため、マクロ定義の場合と同様に、

- ①サービスAPをマクロ定義モードで起動する
- ②ユーザ操作によるパラメータ設定
- ③サービスAPはパラメータをファイルに出力する

④③で出力された情報を、コマンドインタプリタがコマンドファイルに登録する。

4.3. 項目アイコンの実現法

項目アイコンは、デスクトップ以外のウィンドウに表示されるアイコンである。しかし、これもデスクトップ上のアイコンと全く同様に、ユーザに扱われる。従って、デスクトップを管理するコマンドインタプリタは項目アイコンに対するユーザ操作情報を、デスクトップ上のアイコンに対するそれと同様に得る必要がある。

項目アイコンの実現法には以下の3案が考えられる。(図6参照)

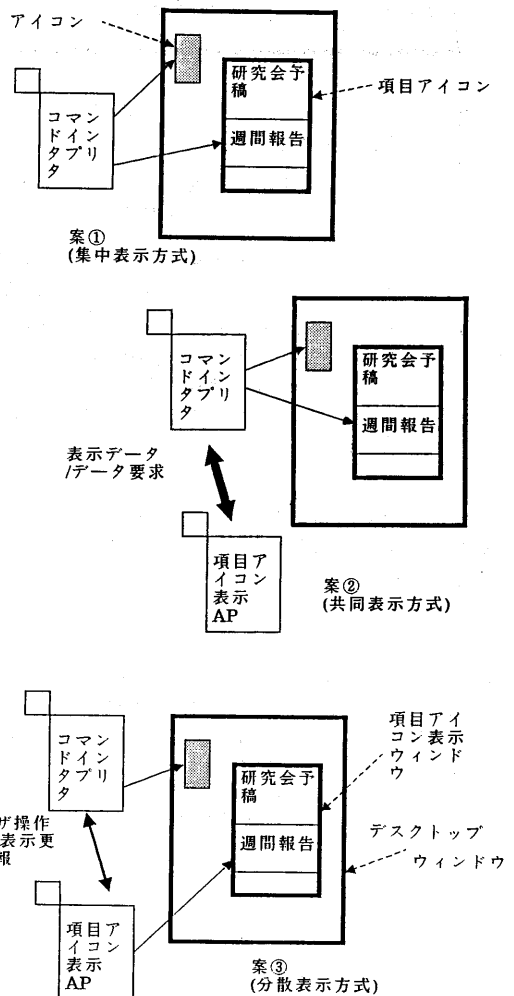


図6 項目アイコン実現方式案概要

表1 項目アイコン実現方式案一覧

評価項目 方式案	サービスAPとコマンドインタプリタ間のインタフェース	コマンドインタプリタの構造	評価
案① (集中表示方式)	最も疎になる。	最も複雑になる(各表示サービスによって異なる項目アイコンの表示情報を作成するため)。また、項目アイコン表示機能の追加/変更が困難。	△
案② (共同表示方式)	サービスAPとコマンドインタプリタ間のデータ授受が最も多い(初期表示及び表示更新の際、サービスAPが作成した項目アイコンの表示データをコマンドインタプリタに送る必要があるため)。	簡単。	△
案③ (分散表示方式)	両者間で、ユーザ操作情報と表示更新指示だけが、授受される。	最も簡単。但し、各APが項目アイコン表示機能を持つ必要がある。	○

案① コマンドインタプリタが項目アイコン表示サービスを行う。(集中表示方式)

案② サービスAPからの指示をうけ、コマンドインタプリタが表示制御を行う。但し、項目アイコン表示に必要なデータはサービスAPが作成する。(共同表示方式)

案③ サービスAPが項目アイコン表示制御を行う。但し、サービスAPは項目アイコンに対するユーザ操作情報を、コマンドインタプリタに通知する。(分散表示方式)

これら3案の検討結果を表1に示す。ここでの評価ポイントは項目アイコンの表示更新機能の実現性である。

表1に示すように、コマンドインタプリタの構造及び、コマンドインタプリタとサービスAP間のインタフェースの双方を簡易化できる案③を採用する。また、案③の短所については、項目アイコン表示機能をモジュール化し、各APにサブルーチンとして共用させる事により、ソフトウェアの構造を単純化する。

5.おわりに

今回は、高性能ワークステーション上のOA用システムにおいて、ユーザ固有の作業環境を提供するコマンドインタプリタの機能設計と実現法の検討について報告した。

今後は、(i)マクロ変数の導入による、マクロ機能とサービスの連続性の結合方式、(ii)マクロプロシージャへの制御構造の導入とその簡易なユーザ定義法について、更に検討を進める。

<謝辞>

日頃御指導いただく分散処理装置研究室川田室長、本検討にあたり有益な御助言をいただいた総合実験網研究室岡田室長、分散処理装置研究室赤木調査員、岸本調査員に深謝致します。

<参考文献>

- [1].Shneiderman,B. : "Direct manipulation : A step beyond programming language" : IEEE Computer, No.8, 57-69, 1983, 16
- [2].菅原他 : マルチウインドウ表示を考慮した画面制御方式の検討 : 画像工学研究会、昭和59年11月
- [3].Good,M. : "Etude and the Folklore of User Interface Design" : Proceedings of the ACM SIGPLAN/SIGOA Symposium on Text Manipulation, 1981
- [4].高田他 : メニュー操作履歴のマクロ化に関する一検討 : 情報処理学会全国大会(60前)、7F-5
- [5].中村他 : OA用簡易言語OFIS/POLのコマンド方式 : 情報処理学会全国大会(59前)、6C-2
- [6].小柳他 : 帳票ベースにおける手順操作について : 情報処理学会全国大会(59前)、6C-6