ゴールド賞

Windows Like 5250への道のり
―さまざまな場面で使えるDelphiおよびDelphi/400

小山 祐二 様
澁谷工業株式会社
経営情報システム部
課長代理

澁谷工業株式会社
http://www.shibuya.co.jp/

パッケージプラントを主力製品とする東証・名証１部上場の機械メーカー。とくに国内外の大手飲料メーカーに採用されているボトリングシステム製造では、世界トップの地位を確立している。近年では無菌化などの技術力を活かし、再生医療事業も積極的に展開している。

1.はじめに

当社は 1931 年創立、1949 年設立の会社である。今日まで多くのお客様に支えられ、2016 年に創立 85 周年を迎えた。創立以来、カスタマーファースト（お客様第一主義）を貫き、お客様のニーズに合わせたパッケージングプラントを、ターンキー（すぐに稼働できる状態）で提供するビジネスを主体としている。また最近では、再生医療分野にも進出している。

当社のホスト・コンピュータの変遷は、1972 年に S/32 を導入したことから始まる。その後、各種モデルを経て、現在のPureFlex System導入に至る。【図1】

その間、多くの基幹システムをキーボード操作入力（以下、CUI）主体の 5250 画面（以下、5250）で自社開発してきた。

近年における当社の基幹システムは、主に Delphi および Delphi/400 で構築しているが、膨大な旧資産の関係上、現在も多くの基幹システムが 5250 で稼働している。それに加え、今後も運用・開発面で 5250 を利用し続けることになる。【図 2】

2.5250の「操作性」評価

現在のインターフェースは、マウス操作入力（以下、GUI）とタッチ操作入力（以下、NUI）が主流である。

そこで、当社のエンドユーザー部門およびシステム部門のメンバー（以下、5250 利用者）の協力のもと、5250 の「操作性」に関してアンケートを実施した。

IBM i は、一般にユーザー評価が非常に高い（『日経コンピューター』 顧客満足度調査　ミッドレンジサーバー部門 18 年連続 1 位）。しかし、当社の 5250 利用者における「操作性」評価の結果とは反比例することがわかった。【図 3】一般ユーザーにおける 5250 の「操作性」評価も、同様だと推測する。

3.5250の「操作性」に対する要望

5250 利用者に、5250 の「操作性」に関する要望をアンケートし、以下にまとめた。【図 4】

(A) マウスホイールによる画面スクロール
(B) 右クリックによるコピー & 貼り付け　等
(C) スクロールバーによる画面スクロール
(D) ショートカットによるコピー & 貼り付け　等
(E) チェックボックスによる項目選択
(F) ダブルクリックによる実行キー打鍵
(G) ラジオボタンによる項目選択
(H) メニューバーによるプログラム（以下、PGM）実行
(I) ダブルクリックによるメニュー PGM 実行
(J) ダブルクリックによる機能キー打鍵

アンケート実施時点では、具体的な対

図1　ホスト・コンピュータの変遷

当社ホストコンピューターの変遷

1972年	…	1979年	…	1984年	…	1989年	…	1991年	…
S/32	…	S/34	…	S/38	…	AS400 (B50)	…	AS400 (D70)	…

1993年	…	1998年	…	2003年	…	2008年	…	2013年	…
AS400 (F70)	…	AS400 (S30)	…	iSeries (i825)	…	Power Systems (9409-M50)	…	Pure Flex System	…

図2　基幹システム構築状況

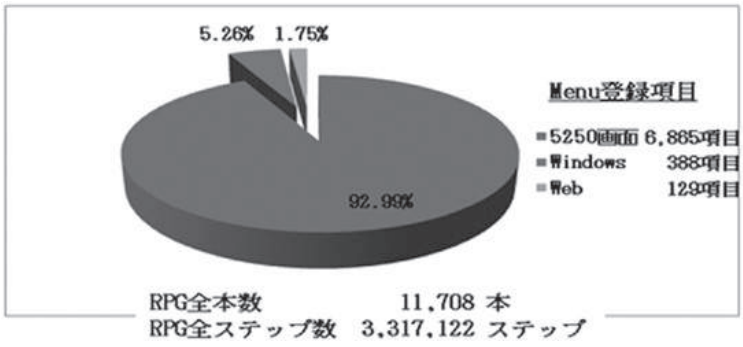


図3　5250「操作性」評価

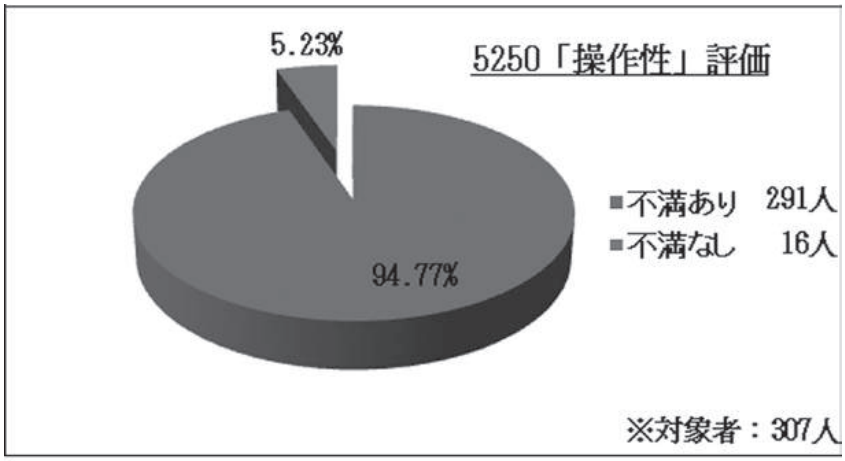
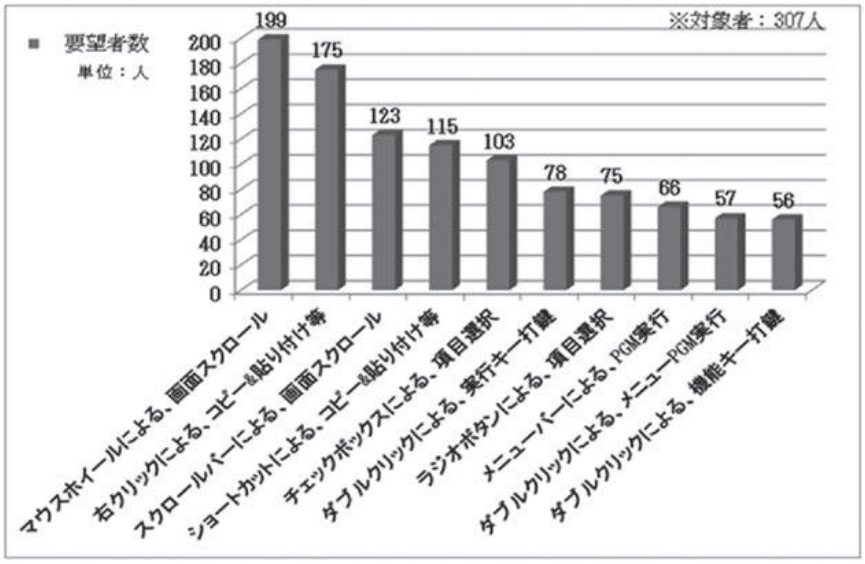


図4　5250「操作性」に対する要望1



応策はなかった。しかしゼロベース思考で、「5250 は CUI」という既成概念を捨て、「5250 でも GUI」との仮説思考をもつ。そこからポジティブ思考で、「Windows Like 5250」を模索することになった。

4.Windows Like 5250への道

最初に、各種既存機能（5250、RPG、画面ファイル等)を調査した。その結果、(C)（E)（F)（G)（H)（I)（J）は実現可能であるが、これらの説明は割愛する(必要であれば別途、問い合わせていただきたい)。本稿では、(B)（D)（A）の詳細を述べる。

4.1 (B) 右クリックによるコピー & 貼り付けと (D) ショートカットによるコピー & 貼り付けの設定方法

5250 既存機能では「ポップアップ・キーパッドの設定」【図 5】で、5250 右クリック時のメニュー設定が可能である。しかし初期値では、(B) は使えない。

調査の結果、「ユーザー定義」に以下の設定変更 ／追加で、(B) を実現した。

設定変更

① NumberOfPads = 3（新しいパッド 3 作成：2 → 3 へ設定変更）

設定追加

② NumberOfRowsPad_3 = 4
（パッド 3：行指定）
③ NumberOfColsPad_3 = 1
（パッド 3：列指定）
④ POP3-1-1 =［edit-copy］
（パッド 3：コピー機能割り当て）
⑤ POP3-1-2 =［edit-cut］
（パッド 3：切り取り機能割り当て）
⑥ POP3-1-3 =［edit-paste］
（パッド 3：貼り付け機能割り当て）
⑦ POP3-1-4 =［edit-clear］
（パッド 3：クリア機能割り当て）

また 5250 既存機能「キーボードの設定」【図 6】の「ユーザー定義」で、キーごとに各種機能の割り振りが可能である。調査の結果、「ユーザー定義」に以下の設定追加で、(D) を実現した。

設定追加

① C-KEY47 =［edit-cut］
② C-KEY48 =［edit-copy］
③ C-KEY49 =［edit-paste］

4.2 (B) 右クリックによるコピー & 貼り付けと (D) ショートカットによるコピー & 貼り付けの PC への展開方法

当社は、全国約 2000 台の PC で 5250 を利用している。そのため、これらの機能をどのように導入するかを検討した。

まず、他企業に問い合わせてみた。その結果、(B) の認識は低いが、(D) は高かった。そこでそれらの導入方法を問い合わせてみたが、有益な情報は得られなかった。

思考錯誤の末､IBM i の IFS(Integrated File System：IBM i の UNIX 互換ファイルシステム）を利用した。そこに設定変更用の Delphi/400 PGM を配置し、5250 利用者が 5250 メニューからその PGM を実行することで、該当 PC の設定変更を実現した。【図 7】【図 8】【図 9】【図 10】【図 11】(*1)

また設定変更時、各種情報を取得した。これは Delphi/400 を利用すれば、まったく問題なく取得可能である。そしてテーブルにトリガー設定を組み込み、エラー時には即座に電子メール配信する仕組みを構築した。【図 12】【図 13】【図 14】

(*1)　5250 画面から Delphi/400 プログラムを実行する方法は「Delphi/400 および Delphi/400 を利用したオンライン個人メニューの構築」（ミガロ .テクニカルレポート　No.7）を参照

4.3 (A) マウスホイールによる画面スクロール

5250 をスクロールする場合、Page Up/Page Down キーの打鍵が必要である。つまり、(A) を実現するには、何らかの方法で 5250 に Page Up ／ Page Down キー打鍵の代替が必要である。

Windows には、常駐 PGM がある。これを利用してマウスホイール操作を監視すれば、実現可能と考えた。後は、どのようにマウスホイール操作を監視するかである。

その後、Windows のメッセージ機能を知ることになった。それは、キーボードやマウスなどの操作情報を OS と PGM 間で受け渡す機能である。

調査後さらに、このメッセージを監視できるフック機能 (*2) について知った。その機能の実装は DLL 化する必要があるが、幸いなことに Delphi や Delphi/400 の開発環境でも、DLL 作成機能が備わっている。

そこで、以下のプロセスを実現する常駐 PGM を作成することで、(A) を実現した。【図 15】【図 16】【図 17】

① 「WH_GETMESSAGE」で、マウスホイール操作情報取得（DLL）
② 「GetforegroundWindow」で、最前面 Window 情報を取得
③　② が 5250 時、Page Up ／ Page Down キー打鍵機能送信（要 IME 機能考慮）
④　上記をスタートアップ登録

(補足)

マウスホイール操作では、「WH_MOUSEWHEEL」がある。これでマウスホイール操作情報は収集可能である。しかし筆者の知る限り、ScrollUp ／ Down が判断できない。多くのユーザーはここで挫折していると推測する。

(*2) 参考文献：Delphi Library［Mr. XRAY］ http://mrxray.on.coocan.jp/index.htm

5.取り組み実施後

取り組み実施後、先の要望に関する利用状況を確認した。【図 18】のとおり、かなり成果があったと考えている。

6.おわりに

IBM i は、安全性・堅牢性などで非常に評価の高いサーバーであるのは、多くのユーザーが認識している。しかしその評価に反比例し、現在の IBM i はさまざまな理由で属人化が進んでいる。また 5250 の「操作性」から、IBM i は古いマシンだと勘違いするユーザーも少なく

図5　ポップアップ・キーパッドの設定

図6　キーボードの設定

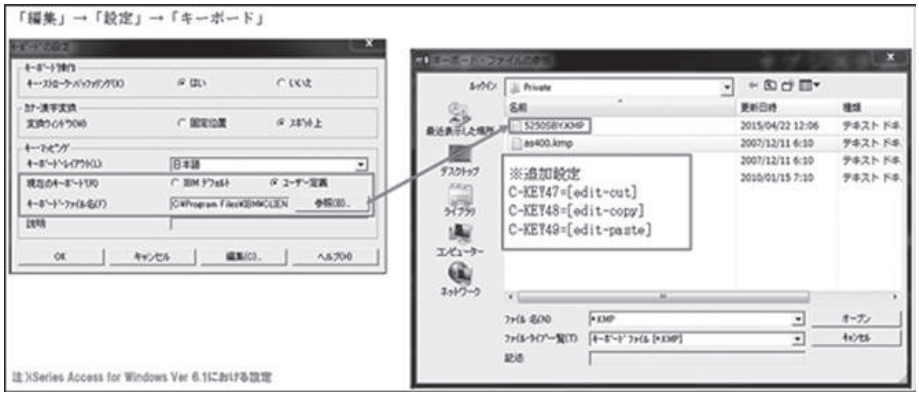
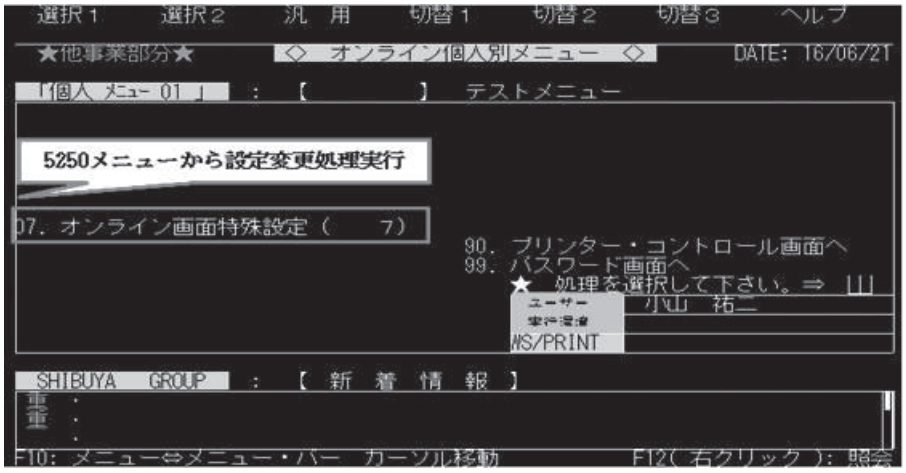


図7　5250メニュー例



ない。

確かにIBM i情報の少なさは、ユーザー共通の悩みだと思う。そのためユーザーは、IBM i関連で実現したいさまざまなプロセスを断念していると推測する。まさに、「IBM iが宝の持ち腐れ」になっていると感じている。非常にもったいない話である。

しかし、Delphi および Delphi/400は情報も多く、IBM i連携も非常に簡単である。また今回紹介した例や各種IBM i運用など、さまざまなプロセスで利用可能である。

今後も、各種プロセスを実現するツールとして、Delphi および Delphi/400を利用していきたい。

最後に、IBM i Access Client Solutionsについて。Windows 7の延長サポートは、2020年1月14日に終了する。しかし既存の5250は、Windows 10に未対応となり、IBM i Access Client Solutionsを利用することになる【図19】。これは既存の5250とほぼ同機能だが、今回紹介した(A)(B)(D)機能は標準で備わっている。

今回の機能を導入したいが、Windowsには詳しくない方は、早急にIBM i Access Client Solutionsの検証（データ転送、ODBC等）を始めるよう推奨する。

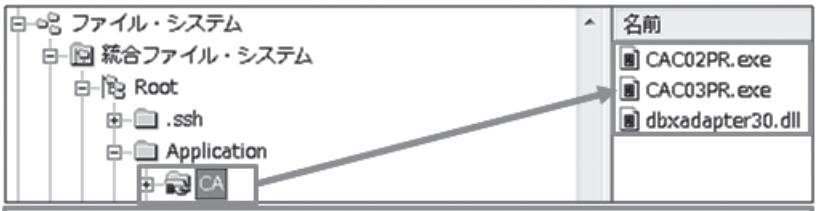
M

図8　CL PGM例

```
/*****************/
/*  IFS の接続 */
/*****************/

         CHGVAR     &W_CMD1 ('NET USE ¥¥' || &W_SYS |< '¥CA /USER:' || +
                      'A¥' ||&W_USER |<  '  || &W_PASS)
         STRPCCMD   PCCMD(&W_CMD1) PAUSE(*NO)

/************************/
/* 設定変更ＰＧＭ実行 */
/************************/

         CHGVAR     &W_CMD2 ('¥¥' || &W_SYS |< '¥CA¥CAC02PR.EXE ' || +
                    &W_JOB || ' 1   || &W_USER |< ' '  || &W_PASS  |< +
                      || &W_SYS |< ' '  || &W_USER2)
         STRPCCMD   PCCMD(&W_CMD2) PAUSE(*NO)
```

任意の文字が必要

図9　Delphi IFS配置例およびPGM実行制約条件



【ユーザーアカウント制御による、PGM実行制約条件】

① 設定変更PGM実行は、管理者権限が必要
② ユーザーアカウント制御画面が表示され、PC管理者権限パスワードが必要
③ Windowsセキュリティー画面が表示され、5250サインオンパスワードが必要

図10　DelphiおよびDelphi/400　コーディング例1

```
begin
  //1. パラメータ取得
  F_Session  := CAC03BFOM.Get_Parameter1() ;
  F_Runkubun := CAC03BFOM.Get_Parameter2() ;
  F_Host     := CAC03BFOM.Get_Parameter3() ;
  F_Directory:= CAC03BFOM.Get_Parameter4() ;
  F_User     := CAC03BFOM.Get_Parameter5() ;
  F_Password := CAC03BFOM.Get_Parameter6() ;
  F_User2    := CAC03BFOM.Get_Parameter7() ;

  //2. 実効PCのWindowsバージョン取得
  F_Winversion := CAC03BFOM.Get_Winversion() ;

  if (F_Winversion <> '8.1')  then
  begin

    MessageDlg('このPCのWindowsバージョンには対応していません。', mtError, [mbYes], 0);
    MessageDlg('「オンライン画面特殊設定」処理をキャンセルしました！', mtInformation, [mbYes], 0);
    exit;

  end;

  //3. コンピュータ名取得
  F_ComputerName := CAC03BFOM.Get_ComputerName(
  //4. IBM i接続
  CAC03BFOM.Str_Connection(F_User,F_Password);

  BringWindowtotop(F_Handle);

  //5. 処理済みチェック
  if CAC03BFOM.Chk_processed(F_ComputerName ,F_Session) then
  begin

    MessageDlg(                                                    rning, [mbYes], 0)
    MessageDlg(
    //IBM i切
    CAC03BFOM.
    exit ;

  end;

  //6. 管理者として実行
  CAC03BFOM.Run_administrator(F_Session,F_Runkubun,F_Directory,F_User,F_Password,F_Host,F_User2);

  //7. IBM i切断
  CAC03BFOM.End_Connection();

end.
```

※Delphi400にて接続

A54001

【設定ファイル変更プロシージャー】
※5250設定値を変更するため、必ず当運用決定者の責任の上、実行する事
（当社は、この処理中で、バックアップを作成している）

図11　DelphiおよびDelphi/400　コーディング例2

```
procedure TCAC03BFUM.Run_administrator(var F_Session  ,
                                       F_Runkubun ,
                                       F_Directory,
                                       F_User     ,
                                       F_Password ,
                                       F_Host     ,
                                       F_User2    :String) ;
var
  sei      : TShellExecuteInfoA ;
begin
 ZeroMemory(@sei, SizeOf(sei));
 sei.cbSize := SizeOf(sei);
 sei.fMask := SEE_MASK_FLAG_DDEWAIT or SEE_MASK_FLAG_NO_UI;
 sei.lpVerb := 'runas';
 sei.lpFile := PAnsiChar('CAC02PR.exe');
 sei.lpParameters:=PAnsiChar(F_Session  +
                             ' '        +
                             F_Runkubun +
                             ' '        +
                             F_User     +
                             ' '        +
                             F_Password +
                             ' '        +
                             F_Host     +
                             ' '        +
                             F_User2    );
 sei.lpDirectory:=PAnsiChar(F_Directory);
 sei.nShow := SW_SHOW ;
 try

   ShellExecuteEx(@sei) ;

 finally

 end;
end;
```

※管理者としてPGM実行

※Delphi400にてRPG実行

Call4001

図12　テーブル設定変更情報

COMPUTER	SESSION	C.A VER	WIN VER	区画名	.PMP ERR	.KMP ERR	実行者
COM01	SES01	5			0	0	
COM02	SES02	5			0		
COM03	SES03	5			0		
COM04	SES04	5	5.1	IBMi01	0	0	社員04
COM05	SES05	5	5.1	IBMi01	0	0	社員05
COM06	SES06	5	5.1	IBMi01	0		社員06
COM07	SES07	5	5.1	IBMi01	0		社員07
COM08	SES08	5	5.1	IBMi01	0		社員08
COM09	SES09	5	5.1	IBMi01	0	0	社員09
COM10	SES10	5	5.1	IBMi01	1	0	社員10

ポップアップキーバットの設定
エラーフラグ

キーボードの設定
エラーフラグ

エラー時
メールにて通知

図13　CL PGMからJava(電子メール配信)実行例

```
 桁 . . . . . :   1  71          走査検索
SEU==>                                                          SMP77AP
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
0026.00 /*******************/
0027.00 /* CCSID ⇒ 5035 */
0028.00 /*******************/
0029.00
0030.00           CHGJOB     CCSID(5035)
0031.00
0032.00 /***************/
0033.00 /* MAIL 送信 */
0034.00 /***************/
0035.00
0036.00           RUNJVA     CLASS(jp.co.shibuya.isd.smp77a) +
0037.00                      PARM(&SMTP &MAIL1 &MAIL2 &NAME &TITLE &TEXT +
0038.00                           &FOLDER &FLNM &DLT) +
0039.00           CLASSPATH('.:+
0040.00                     /jp/co/shibuya/isd/lib/mail.jar:+
0041.00                     /jp/co/shibuya/isd/lib/activation.jar')  +
0042.00           OUTPUT(*NONE)

F3= 終了  F5= 最新表示  F9=ｺﾏﾝﾄﾞの複写  F10=ｶｰｿﾙ移動  F11= 切り替え  F12= 取り消し
F16= 検索の反復    F24= キーの続き
```

図14　Java IFS配置例および配信電子メールイメージ

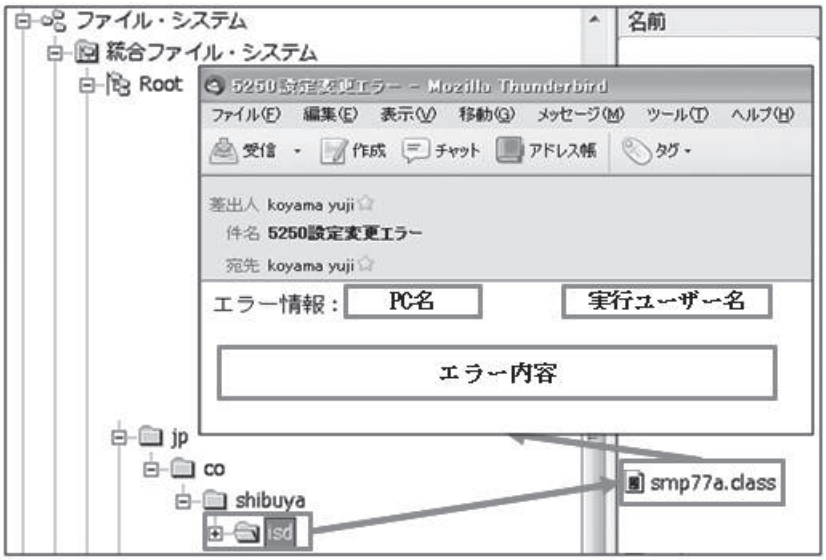


図15　(A)の操作イメージ

図16　フック関数「WH_GETMESSAGE」コーディング例（DLL側）

```
function StartGetMessageHook(HostWnd: HWND; AMsg: Integer): Boolean; stdcall;
var
  F_LMapWnd : THandle;
  F_LpMap   : Pointer;
begin

  Result := False;

  //メモリマップドファイル使用準備
  MapFileMemory(F_LMapWnd, F_LpMap);

  if F_LpMap = nil then
  begin
    F_LMapWnd := 0;
  end
  else
  begin

    //フック情報構造体初期化とフック関数の登録
    pHookInfo(F_LpMap)^.HostWnd    := HostWnd;
    pHookInfo(F_LpMap)^.RevMessage := AMsg;

    //フックをインストール
    //ローカルフックの場合は3番目の引数は0(null)でOK
    //グローバルの場合はDLLのインスタンス
    //4番目はフック対象のスレッドID(システム全体の時は0)
    pHookInfo(F_LpMap)^.HookHandle := SetWindowsHookEx(WH_GETMESSAGE,
                                                       @MyHookProc,
                                                       hInstance,
                                                       0);

    //フック成功
    if (pHookInfo(F_LpMap)^.HookHandle > 0) then
    begin

      Result := True;

    end;

    //メモリマップドファイル使用終了処理
    UnMapFileMemory(F_LMapWnd, F_LpMap);

  end;

  end;
```

図17　5250時、Page Up／Page Downキー打鍵 コーディング例（EXE側）

```
function StartGetMessageHook(HostWnd: HWND; AMsg: Integer): Boolean; stdcall;
var
  F_LMapWnd : THandle;
  F_LpMap   : Pointer;
begin

  Result := False;

  //メモリマップドファイル使用準備
  MapFileMemory(F_LMapWnd, F_LpMap);

  if F_LpMap = nil then
  begin
    F_LMapWnd := 0;
  end
  else
  begin

    //フック情報構造体初期化とフック関数の登録
    pHookInfo(F_LpMap)^.HostWnd    := HostWnd;
    pHookInfo(F_LpMap)^.RevMessage := AMsg;

    //フックをインストール
    //ローカルフックの場合は3番目の引数は0(null)でOK
    //グローバルの場合はDLLのインスタンス
    //4番目はフック対象のスレッドID(システム全体の時は0)
    pHookInfo(F_LpMap)^.HookHandle := SetWindowsHookEx(WH_GETMESSAGE,
                                                       @MyHookProc,
                                                       hInstance,
                                                       0);

    //フック成功
    if (pHookInfo(F_LpMap)^.HookHandle > 0) then
    begin

      Result := True;

    end;

    //メモリマップドファイル使用終了処理
    UnMapFileMemory(F_LMapWnd, F_LpMap);

  end;

  end;
```

図18　各種取り組み実施後の要望利用状況

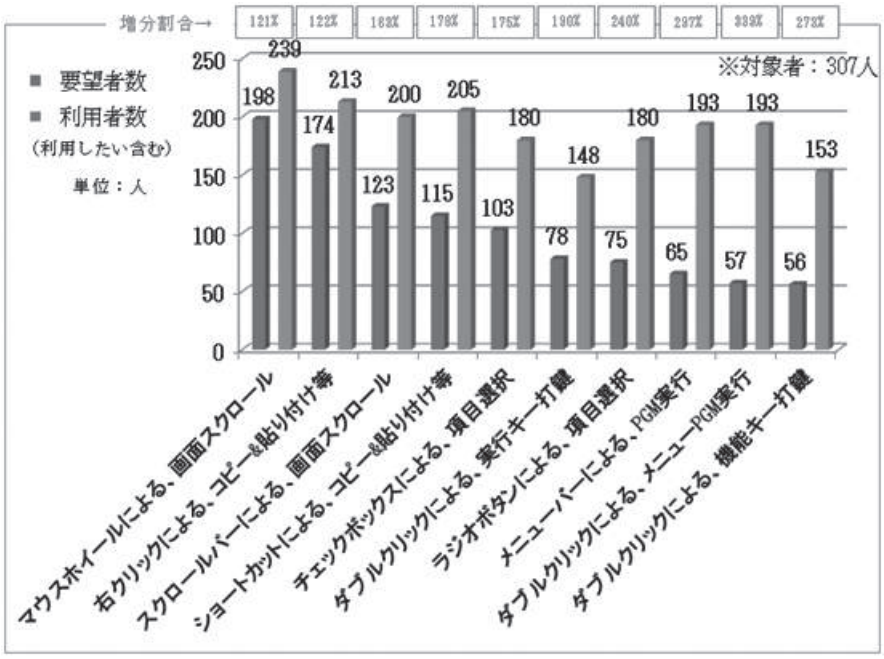


図19　IBM i Access Client Solutions