ショートノート

日本語プログラミング用エディタ評価のための識別子ベンチマークデータ

尾関 哲[†] 佐藤邦弘[††]
太田健一[††] 宮脇冨士夫[††]

我々は，プログラムの生産性を向上させる手段として日本語プログラミングに着目し，開発環境を構築した．本環境のエディタは，日本語識別子の簡略入力方式に特徴がある．その評価を行うためには日本語識別子のテストデータが必要となる．しかしながら，現状では実用的な日本語プログラムの入手が困難である．したがって，ベンチマークテストプログラムのような標準的日本語識別子のデータも作成されていない．そこで，我々が本環境で開発した日本語プログラムの中から識別子を抽出し，テスト用データを作成した．このデータは，識別子の文字種，文字列長さの分布をもとに作成したものである．これを日本語プログラムにおける識別子のベンチマークデータとしたい．

Benchmark Data of Identifier for Evaluation of Editors in Japanese-based Programming

SATOSHI OSEKI,[†] KUNIHIRO SATO,[††] KENICHI OHTA[††] and FUJIO MIYAWAKI[††]

We have developed an environment for Japanese-based programming to promote software productivity. The editor in this environment has a feature in the abbreviated input method. Its evaluation requires test data of identifiers in Japanese-based programming. Since Japanese-based programming has not been popular yet, it is very difficult to get any test data like benchmark test programs. We picked up identifiers from Japanese-based programs developed in this environment and made a set of data. This data was made under the distribution of the kind of characters and the length of identifiers. We regard this data as benchmark data of identifiers in Japanese-based programs.

1. はじめに

我々は，プログラムの生産性を向上させる手段として日本語プログラミングに着目し，**図1**に示す開発環境を構築した．本環境の構成は，日本語プログラムの専用エディタ[1]，日本語 C++ から C++ へのトランスレータおよび既存の OS 上のコンパイラ，リンカからなる．本環境のエディタは，コマンドベースのユーザインタフェースをもつエディタで，日本語識別子の簡略入力方式に特徴がある．この簡略入力方式による日本語識別子入力の効率を評価するためには日本語プログラミングにおける識別子のテストデータが必要となる．しかしながら，現状では実用的な日本語プログラムが公開されていないので，ベンチマークテストプログラムのようなテストデータの入手が困難である．そこで，我々が本環境の中で開発した日本語プログラムの中から識別子を抽出し，識別子のベンチマークデータを作成した．このデータは，他の日本語プログラム開発環境の評価にも用いることができると考える．

† 神戸市立工業高等専門学校電気工学科
Department of Electrical Engineering, Kobe City College of Technology
†† 姫路工業大学工学部情報工学科
Department of Computer Engineering, Faculty of Engineering, Himeji Institute of Technology

2. 識別子の抽出

識別子は，本環境の日本語エディタとトランスレータのソースプログラムから抽出した．両プログラムとも日本語 C++ で書かれている．大きさは，行数にして合計約 7200 行である．日本語 C++ とは，言語 C++ の予約語を日本語化し，識別子に日本語が使え

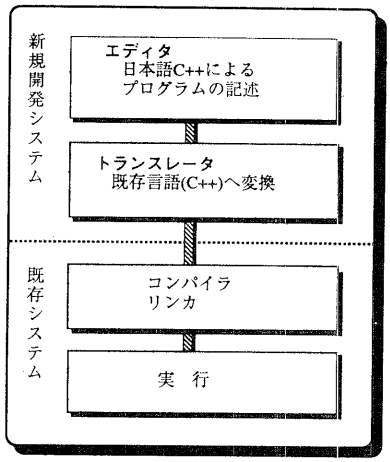


図 1 日本語プログラミング環境
Fig. 1 Japanese-based programming environment.

るようにしたものである．ここで識別子とは，変数名，関数名，定数名，マクロ名などプログラマがつける名前を指す．日本語 C++ では識別子に使える文字の種類および綴りには制限を設けていない．**図 2** に日本語エディタのソースプログラムから日本語C++プログラムの一部を示す．

次に，抽出した識別子の統計処理について述べる．我々は，作成したベンチマークデータを日本語エディタの識別子入力効率の評価に使用することが目的であるので，識別子の入力時の手間を考慮した統計処理を行った．日本語プログラムにおける識別子には各種の文字が使われる．現在，日本語をキーボード入力するには仮名漢字変換を経るのが一般的である．したがって，識別子を構成する文字種により入力の手間は異なる．そこで，識別子を入力の過程に着目し，つぎの4つのグループに分けた（表2参照）．

1) 漢字および平仮名からなる識別子
仮名漢字変換操作を必要とする．

2) 平仮名・片仮名のみの識別子
仮名漢字変換操作は伴わないが，仮名文字は仮名入力かローマ字入力かにより入力の手間が異なる．

3) 英数字記号からなる識別子
1キー入力1で文字が入力できる．

4) 各文字種の混合した識別子
仮名漢字変換操作および入力モードの変換操作が必要である．

識別子を構成する文字種による分類結果を**表 1** に示す．

また，入力の手間には識別子の長さが影響する．前述のとおり日本語識別子では単純に識別子の長さを入力の手間に換算できないが，前述の同一グループ内であれば入力の手間は，識別子文字列の長さ（バイト数）にほぼ比例すると考えて，各グループについてその度数分布を取った(**図 3～6**)．平仮名・片仮名のみの識別子グループにおいて片仮名は，1バイト・コードとしてカウントした．

```
空値　名前ﾃｰﾌﾞﾙ::初期状態設定()
{
    終わりﾏｰｸ=500。
    ｷｬﾗｸﾀﾌﾟｰﾙ容量=5000。
    分類順先頭=終わりﾏｰｸ。
    入力順先頭=終わりﾏｰｸ。
    ｷｬﾗｸﾀﾌﾟｰﾙ空地先頭=0。
    繰返条件(整数　ｶｳﾝﾀ=0。ｶｳﾝﾀ!=終わりﾏｰｸ。ｶｳﾝﾀ+=1)で次の文を繰り返す。
        名前情報[ｶｳﾝﾀ].分類順= -1。
}
```

図 2 日本語 C++ のプログラム例
Fig. 2 An example of Japanese C++ program.

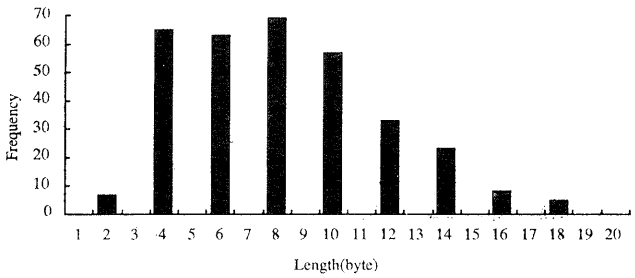


図 3 漢字および平仮名グループの文字列長さ別頻度
Fig. 3 Length of strings with Kanji and Hiragana characters.

表 1 識別子のグループ
Table 1 Groups of identifier.

グループ	種 類	割合(%)
漢字および平仮名	330	47.2
平仮名・片仮名のみ	23	3.3
英数字記号	82	11.7
各文字種の混合	264	37.8
合 計	699	100

3. 識別子ベンチマークデータ

以下にベンチワークデータの作成手順を述べる．この作成手順は，3章で述べたように識別子を構成する文字の種類により分

表 2 ベンチマーク識別子データ
Table 2 Benchmark identifiers to evaluate editors in Japanese-based programming.

	A	B	C	D		A	B	C	D		A	B	C	D		A	B	C	D
項		○		○	着目行		○	○	○	画面退避	○	○	○	○	関数定義変換				○
式	○	○	○	○	入力順	○			○	名前情報			○	○	関数頭部情報		○		○
行			○	○	短整数		○		○	初期設定		○	○	○	日本語文字列		○	○	○
陽			○	○	宣言名			○	○	終了時刻			○	○	関数末尾処理			○	○
多義			○	○	要素長		○	○	○	要素挿入	○		○	○	文字定数検査	○		○	○
印刷		○		○	先頭頁		○	○	○	関数検索		○	○	○	先頭画面表示			○	○
黄色		○		○	終端行				○	文字位置			○	○	使用関数確認	○	○		○
限度	○	○	○	○	行まで		○	○	○	退避領域			○	○	識別名集読込			○	○
行順			○	○	先頭行			○	○	演算子表			○	○	初期状態設定	○	○	○	○
新規		○	○	○	行情報		○	○	○	画面再現	○	○	○	○	関数末尾情報		○	○	○
桁数			○	○	表示色			○	○	名前表示		○		○	関数頭部処理		○	○	○
部分			○	○	型指定	○	○	○	○	表示位置	○			○	関数階層印刷		○		○
場合	○	○	○	○	列挙子			○	○	関数終り			○	○	名前以外印刷	○	○	○	○
定数			○	○	倍精度		○		○	関数の数		○		○	列挙子指定子		○	○	○
終了	○	○		○	桁情報	○		○	○	名前入力		○		○	全定数文字数				○
行数	○	○	○	○	平均値	○		○	○	位置設定		○	○	○	全名前文字数			○	○
頭部			○	○	演算子	○			○	識別名番号		○	○	○	名前添字印刷			○	○
複写		○	○	○	文変換		○	○	○	入力順検査				○	末尾画面表示	○	○	○	○
同類				○	式並び	○		○	○	最終入力行		○		○	関数階層順先頭	○	○	○	○
自動		○		○	項変換			○	○	被置換名長			○	○	行末ならば改行			○	○
外部			○	○	見出し				○	画面再表示			○	○	宣言指定子変換	○		○	○
改行				○	次要素		○	○	○	入力最終行	○	○	○	○	関数分類順先頭			○	○
名前	○			○	識別名		○	○	○	直前比較名			○	○	指定行から表示		○	○	○
継続		○		○	行分割		○	○	○	抽象宣言子				○	初期設定子変換				○
番号			○	○	次の頁			○	○	入力文終了			○	○	大引数宣言並び			○	○
関数		○	○	○	置換名		○	○	○	名前開始行	○		○	○	初期設定子並び		○	○	○
複文	○		○	○	上移動			○	○	演算子出力	○	○	○	○	識別名辞書更新	○	○	○	○
末尾			○	○	入力位置			○	○	演算子印刷	○	○	○	○	画面表示頁番号		○	○	○
挿入		○		○	画面消去		○	○	○	文字列終端		○	○	○	現在時刻文字列			○	○
固定			○	○	行番号へ	○	○	○	○	時刻文字列		○	○	○	関数宣言子変換				○
無用	○		○	○	全文字数			○	○	識別名登録		○	○	○	空白付文字読込		○		○
公開			○	○	繰返条件		○	○	○	行番号から		○		○	指定文字まで変換		○	○	○
移動		○	○	○	全要素数	○			○	入力順先頭		○	○	○	次の文を繰り返す			○	○
状態		○	○	○	空地整理			○	○	分類順先頭			○	○	繰返開始点に戻る		○		○
尾部		○	○	○	終了信号	○	○	○	○	名前要素数	○	○	○	○	複合型指定子変換	○		○	○
自体		○	○	○	定数検査		○	○	○	色画面表示			○	○	前の文を繰り返す			○	○
引数			○	○	繰返終了			○	○	式並び変換		○	○	○	中かっこ組合せ検査	○		○	○
定義	○		○	○	単引用符			○	○	次画面表示		○	○	○	大引数宣言並び変換			○	○
倍長		○	○	○	文書作成	○	○		○	宣言指定子		○		○	初期設定子並び変換		○		○
設定	○	○	○	○	終わり行		○		○	行番号読取		○	○	○	メンバリスト			○	○
文字		○	○	○	被置換名			○	○	関数階層順			○	○	そうでなければ			○	○
表示順			○	○	正常表示		○		○	識別名出力		○		○	ｶｳﾝﾀ		○	○	○
行接続		○	○	○	引数宣言		○	○	○	文字列連結			○	○	ｲﾝﾗｲﾝ		○	○	○
無符号	○	○	○	○	選択番号			○	○	更新初期化		○		○	クラス		○	○	○
多義化		○	○	○	使用目印		○	○	○	分類開始行			○	○	ならば			○	○
行の数	○			○	要素削除			○	○	表示開始行	○		○	○	ｵﾍﾟﾚｰﾀ		○	○	○
桁位置				○	属性領域		○		○	簡略名処理	○		○	○	ｱｾﾝﾌﾞﾙ	○			○
定数数		○		○	終わり桁	○		○	○	関数宣言子			○	○	ﾌｧｲﾙｸﾛｰｽﾞ		○	○	○
行から			○	○	定義登録	○	○		○	置換名要素	○			○	ﾌｧｲﾙﾊﾝﾄﾞﾗ	○			○
終文字	○	○	○	○	文字属性			○	○	複合型指定子			○	○	ﾌｧｲﾙ･ｵｰﾌﾟﾝ		○	○	○

A---100個の場合 B---200個の場合 C---300個の場合 C---400個の場合

1. 漢字および平仮名グループ 項～初期設定子並び変換
2. 平仮名・片仮名のみグループ メンバリスト～ｽｲｯﾁ
3. 英数字記号グループ ｣～J2CPLS01
4. 各文字種の混合グループ 式。～日本語C++をC++へ変換

表 2 (続き)
Table 2 (continued.)

	A	B	C	D		A	B	C	D		A	B	C	D		A	B	C	D
が		○	○	○	行_5			○	○	行番号_2	○	○	○	○	continue文	○	○	○	○
ｽｲｯﾁ	○			○	行_3			○	○	ﾌｧｲﾙ不明			○	○	クラス頭部	○	○	○	○
]		○	○	○	桁_2	○	○	○	○	case文_1		○		○	ﾌｧｲﾙ指示子			○	○
)			○	○	do文		○	○	○	else文_1			○	○	ｶｰｿﾙ位置_Y			○	○
+		○	○	○	桁_3				○	switch文	○	○		○	退避領域_1	○		○	○
¦	○			○	桁_1	○	○	○	○	ｺﾒﾝﾄ検査	○	○	○	○	終わり行ﾏｰｸ		○		○
(		○	○	○	桁_X				○	elseif文		○		○	include文_1	○		○	○
}			○	○	.記号			○	○	return文				○	名前ﾃｰﾌﾞﾙ_1		○	○	○
.		○	○	○	ﾘｽﾄ済	○	○		○	16進記号		○	○	○	演算子ﾃｰﾌﾞﾙ			○	○
。			○	○	1次式			○	○	ﾌｧｲﾙ名他		○	○	○	関数名ﾘｽﾄ_1				○
&		○	○	○	0記号		○	○	○	OSに戻る		○	○	○	C++文法ﾁｪｯｸ			○	○
,			○	○	文字_3		○		○	行情報_1			○	○	ﾌﾟﾛｸﾞﾗﾑ終了		○	○	○
H	○		○	○	定数_1			○	○	要素長_1			○	○	default文_1	○	○	○	○
=			○	○	ｶｳﾝﾀ_2		○	○	○	ｷｰ入力_1		○	○	○	処理ﾒﾆｭｰ表示		○		○
{		○	○	○	ｶｰｿﾙ_2			○	○	演算子_X	○		○	○	辞書ﾌｧｲﾙ読込	○	○	○	○
@		○		○	左上_Y			○	○	印刷ﾒﾆｭｰ			○	○	ｼｽﾃﾑ領域表示			○	○
x	○			○	ﾊﾟｰｻ_4		○	○	○	ﾌｧｲﾙ入力		○	○	○	文字列指示_1		○	○	○
-		○	○	○	ｶｳﾝﾀ_4	○	○	○	○	ｺﾒﾝﾄ検索		○		○	全ｺﾒﾝﾄ文字数				○
!	○			○	ﾊﾟｰｻ_1			○	○	関数名_2			○	○	switch文変換			○	○
!=		○	○	○	要素_3		○	○	○	関数名_3		○		○	次非ｺﾒﾝﾄ要素	○		○	○
<<		○	○	○	目印_2		○	○	○	ﾌｧｲﾙ追加			○	○	現在時刻(秒)		○	○	○
++			○	○	「」の数				○	if文変換	○		○	○	define文変換				○
*=			○	○	ｶｳﾝﾀ_5		○	○	○	識別名_1			○	○	elseif文変換			○	○
ah		○	○	○	添字_2			○	○	define文	○	○	○	○	クラス指定子		○	○	○
^=	○	○		○	ｷｰ入力				○	ﾌｧｲﾙ読込		○		○	演算子番号_2			○	○
ds		○	○	○	要素_O´		○		○	break文_1			○	○	関数宣言子_1	○			○
::			○	○	右下_X	○		○	○	ｺﾝｿｰﾙ入力			○	○	演算子番号_3	○	○	○	○
al		○		○	要素_2	○	○	○	○	行ﾃｰﾌﾞﾙ_1	○	○	○	○	ｷｬﾗｸﾀﾌﾟｰﾙ容量	○		○	○
tm			○	○	文字_2	○	○	○	○	関数ﾘｽﾄ順		○		○	1文字読み込み				○
!=				○	目印_1				○	include文				○	ﾌﾟﾘﾌﾟﾛｾｯｻ処理		○	○	○
di	○	○	○	○	ｶｰｿﾙ行	○	○	○	○	終わりﾏｰｸ		○	○	○	名前ﾃｰﾌﾞﾙ項数		○		○
¦¦	○		○	○	case文	○		○	○	関数名ﾘｽﾄ	○	○	○	○	演算子ﾃｰﾌﾞﾙ_1	○		○	○
==	○	○	○	○	ｶｳﾝﾀ_1				○	1文字表示			○	○	ｷｬﾗｸﾀﾌﾟｰﾙ終端		○	○	○
&&			○	○	要素_N			○	○	default文	○	○	○	○	ﾌｧｲﾙ追加後処理		○	○	○
>=				○	無用_2		○	○	○	ﾃｰﾌﾞﾙ整理		○	○	○	日本語ﾌｧｲﾙ読込	○		○	○
>>=			○	○	文字_4			○	○	1次式変換				○	識別名対応ｺﾒﾝﾄ		○	○	○
far	○	○		○	ﾘｽﾄ目印				○	終り桁ﾏｰｸ			○	○	戻値無return文			○	○
dos		○	○	○	ﾘｽﾄの数			○	○	case文変換		○	○	○	Cの識別名入力		○		○
time	○	○		○	break文	○	○	○	○	86割り込み			○	○	演算子ﾌｧｲﾙ出力	○		○	○
FILE			○	○	ﾘｽﾄ番号			○	○	ｺﾒﾝﾄ要素数				○	戻値付return文		○		○
math			○	○	1行表示		○	○	○	要素挿入_1	○	○	○	○	ｼｽﾃﾑ領域初期化			○	○
SREGS	○	○	○	○	C++ﾌｧｲﾙ	○			○	elseif文_1				○	continue文変換			○	○
conio			○	○	画面ｸﾘｱ			○	○	ｶｰｿﾙ位置_X				○	名前ﾃｰﾌﾞﾙ項数_1		○	○	○
string			○	○	1行空白		○	○	○	文字位置_2			○	○	桁ﾃｰﾌﾞﾙ空地先頭			○	○
_Cdecl	○	○	○	○	行ﾃｰﾌﾞﾙ		○	○	○	ｺﾒﾝﾄ文字数		○		○	移動複写ﾒﾆｭｰ表示	○		○	○
time_t				○	文字ﾀｲﾌﾟ				○	表示文字_1			○	○	宣言指定子変換_1			○	○
J2CPLUS		○		○	関数名_1	○	○	○	○	文字ﾀｲﾌﾟ_1		○	○	○	クラス指定子変換		○		○
delline			○	○	ｱｾﾝﾌﾞﾙ文			○	○	C++へ変換		○	○	○	ｷｬﾗｸﾀﾌﾟｰﾙ空地先頭		○	○	○
J2CPLS01		○	○	○	辞書ﾌｧｲﾙ			○	○	ﾌｧｲﾙ読込_1	○	○	○	○	戻値付return文変換		○	○	○
式。		○	○	○	桁情報_1	○		○	○	ｶｰｿﾙ位置_x		○		○	日本語C++をC++へ変換		○	○	○

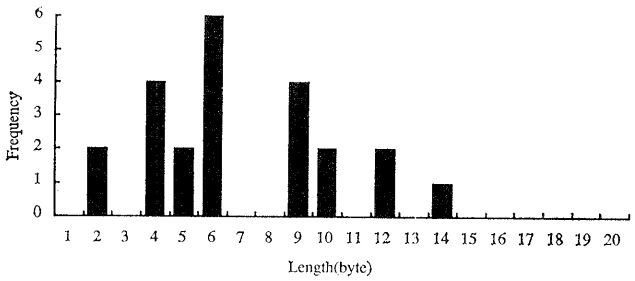


図 4 平仮名・片仮名のみグループの文字列長さ別頻度
Fig. 4 Length of strings with Kana characters.

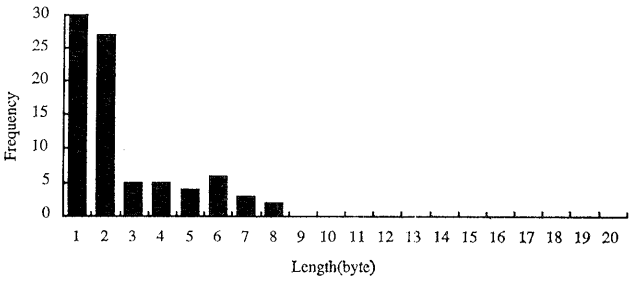


図 5 英数字記号グループの文字列長さ別頻度
Fig. 5 Length of strings with alphanumeric characters and symbols.

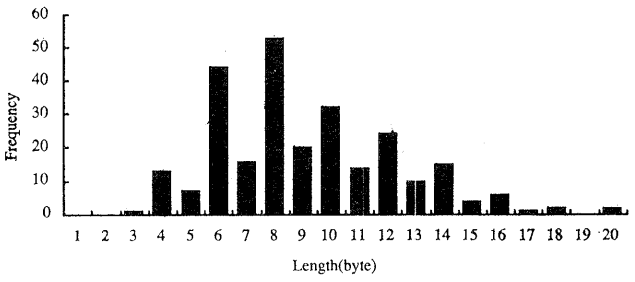


図 6 各文字種の混合グループの文字列長さ別頻度
Fig. 6 Length of strings with mixed characters.

類したグループ内では，入力時間がその文字列長さに比例するという考えに基づく，そこで，まず取り出す個数を表1のグループの割合で分割する．作成するベンチマークデータのデータ（識別子）数を N とすると，表1より，漢字および平仮名グループから $N\times$ 47.2/100，平仮名・片仮名のみグループから $N\times$3.3/100，英数字記号グループから $N\times$11.7/100，各文字種の混合グループから $N\times$37.8/100 のように分割する．つぎに各グループごとに文字列の長さ分布（図3〜6）に応じて取り出す個数を決定する．こうすることによって，元の識別子集合のグループ割合，識別子長さの分布を満たす集合が得られる．今回，この手順で100個，200個，300個，400個の識別子ベンチマークデータを作成した．**表2**に作成したベンチマークデータを示す．一般的なエディタと日本語入力機能に工夫のなされたエディタで，この日本語識別子データの入力に要する時間を計測，比較すれば，日本語プログラミングにおけるエディタのひとつの評価になると考える．簡易な評価は個数の少ないもので，入念な評価は個数の多いもので行うことができる．

4. おわりに

日本語プログラミング言語の普及のためには，言語自体の特質，開発環境に対する考察が重要である．今回は，開発環境の一部であるエディタの評価の基準としての識別子のベンチマークデータを作成することができた．今後は，このベンチマークデータを使って日本語プログラムエディタの評価を行い，我々の提案する簡略入力方式の有効性を検証する予定である．

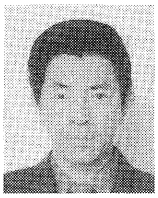
参考文献

1) 尾関 哲，佐藤邦弘，宮脇冨士夫：日本語識別名入力機能を強化した日本語プログラミング用エディタの開発，第8回ソフトウェアコンファレンス・プロシーディングス，pp. 181-184 (1992).

(平成5年11月29日受付)
(平成6年 6 月20日採録)

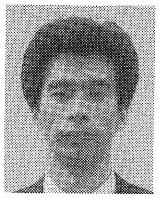
尾関　哲（正会員）

昭和 41 年生．平成元年姫路工業大学電気工学科卒業．平成 3 年同大学院修士課程（電気電子工学専攻）修了．同年神戸市立工業高等専門学校電気工学科助手．現在同高専講師．日本語プログラムおよびプログラミング環境に関する研究に従事．日本ソフトウェア科学会会員．



佐藤　邦弘

昭和 27 年生．昭和 51 年熊本大学工学部電気工学科卒業．昭和 56 年京都大学大学院博士課程単位取得退学．昭和 58 年 10 月姫路工業大学電気工学科助手．平成 5 年 4 月情報工学科助手，現在に至る．京都大学工学博士．制御熱核融合炉開発に関する理論解析および計算機シミュレーション研究に従事．日本物理学会，プラズマ・核融合学会，米国物理学会各会員．



太田　健一（正会員）

昭和 29 年生．昭和 51 年信州大学卒業．昭和 53 年同大大学院修士課程修了．同年～平成 4 年兵庫県立工業技術センター主任研究員．平成 4 年神戸山手女子短期大学助教授．平成 5 年姫路工業大学工学部助教授，現在に至る．工学博士．色彩画像処理，テキスタイル CAD・CAM 開発などの研究に従事．平成 2 年度繊維科学振興会表彰受賞．平成 3 年度日本繊維機械学会学術奨励賞受賞．日本繊維機械学会，電子情報通信学会各会員．

宮脇冨士夫（正会員）

昭和 10 年生．昭和 37 年姫路工業大学電気工学科卒業．同研究生を経て，昭和 41 年姫路工業大学電気工学教室助手，昭和 53 年同講師，昭和 56 年同助教授，平成 4 年同教授，平成 5 年同情報工学教室教授，現在に至る．計算機関係の研究に従事．京都大学工学博士．電子情報通信学会，電気学会，日本シミュレーション学会，日本ソフトウェア科学会，日本科学史学会各会員．