

東海地域能楽番組データベースの使用プログラム (I)

——能楽番組データ検査プログラム“CHECK1”を中心に——

飯塚恵理人・三木邦弘・深谷 哲

1. はじめに

平成9年4月に名古屋能楽堂が開館した。この能楽堂の一つの特徴として、能楽に関する展示室が設けられ、市民に対して能楽に関する啓蒙活動が意図されている点がある。市民に能楽に関する啓蒙活動を行うためには、まず江戸時代より現代に至る名古屋の能楽の特徴を明らかにする必要がある。この展示室の構想が具体化し、展示物を選ぶなどの要求から、資料の収集と整理を迅速に進める必要が生じた。また、この展示室の計画と別に、これらの資料の収集・整理から名古屋の近代の能楽を支えた人を記録する本を作ろうという計画もあった。能楽協会名古屋支部が企画した『近代名古屋の能楽を支えた人々』編纂計画がそれで、この本は番組の整理とその索引による「資料編」と、特に能楽界に功績のあった人を中心に記述する「人物編」とに分けて発行する予定であった。その後、事業主体は能楽協会名古屋支部より東海能楽研究会に移管された。諸般の事情で『近代名古屋の能楽を支えた人々』の発行には到っていないが、その準備段階の本として『東海地域能楽番組一覧（明治元年～昭和二六年）』¹⁾を発行することが出来た。

本事業は、当初能楽協会名古屋支部の委託を受ける形で始めたため、番組などの資料は東海地域に限定されている。また、資料を整理するためのデータベースプログラムも能楽番組向け専用で作られた。しかしながら、情報機器の発達とともに国文学・文化史レベルでのデータベースの構築・運用が盛んになるにつれて、能楽以外でもデータベースの構築を行いたいという人が増えてきた。ところが、飯塚宛に質問してこられた方の多くは国文学・文化史専攻の方で、お互いにデータベースソフト作成についての知識が乏しく、ソフトそのものを提供して欲しいと言う依頼が多かったので、今回より使用しているソフトのうち汎用性の高い物を選び、この論集において公開することとした。能楽番組の収集は現在も継続して行っており、ソフトも改良を続けている。その経過も必要に応じて報告する予定である。

本稿においては、使用するソフトの開発を三木が、データベースの構成及び運用の方針を深谷が、能楽に関する知識を飯塚が担当した。本稿の責任は三人が負うものとする。

2. 東海地域能楽番組で作成した資料とプログラム

現在、能楽番組データベースには、以下のような形式で番組が入力されている。

会名, 奉献能組 1 日目	形態, 能
日時, 明治10年10月18日	演目, 邯鄲
場所, 不明	シテ, 寺田左門治
	ワキ, 不明
形態, 翁	ワキツレ, 不明
演目, 翁	笛, 吉田万兵衛
翁, 寺田左門治	小鼓, 高田栄久
千歳, 筒井秀之助	大鼓, 石井孫三
三番叟, 山脇録弥	太鼓, 鬼頭為太郎
注記, 橋掛ノ舞	
笛, 吉田万兵衛	(中略)
小鼓, 高田栄久	形態, 狂言
小鼓, 森勘介	演目, 末廣
小鼓, 中島豊吉	シテ, 山脇伊津美
大鼓, 石井孫三	

能の番組に「会名」「日時」「形態」「演目」などのタグをつけて分類している。そしてこのタグは、その時代に行われた能の演奏の形式と番組の記述方法によって、何回かの改訂を加えた。このことに関しては次節以降に述べる。

3. 検査プログラム“CHECK1”とその問題点

能楽番組データベースはタグごとに分類され、その各タグの中で、表示する順序を決めている。例えば「演目」「シテ」のタグはそのタグのついている項目の漢字に読みをつけ、五十音順で配列している。また「形態」は能楽として重く扱われるもの、重要なものから順序を付け、その順番に表示されるようにしている。ただし、形態については後述のように、同じ形態に属するものでも複数の呼び方がある場合があり、それを統一する必要がある。そのためにタグを検索・表示するプログラムが必要となった。また、入力時の誤りによって、タグを付け忘れたり、二重に付けたりと言う不正なデータも存在する。このような不正データをチェックし、元データを訂正する必要もある。このようなデータの検査用プログラムが“CHECK1”である。以下に“CHECK1”のプログラムを挙げる。

3.1 check1.c

```

/*                                     */
/*  能楽データ検査プログラム         */
/*      by K.Miki                     */
/*  94/05/05 ver. 1 from wi.c (ver.1) */
/*  94/05/12 use init.c               */
/*  94/06/05 put ref. pos.            */
/*  95/04/28 to UNIX                  */

```

```

/* 98/01/26 multi file version */
/* */

#include <stdio.h>
#include <string.h>

#include "init.h"

#define LINE_LENGTH 1000
#define REF_MAX 5

FILE *fpi,*fpo;

struct reftyp {
    int count,p[REF_MAX],fn[REF_MAX];
    struct reftyp far *right, far *left;
    char far *val;
};

struct wordtyp{
    char far *word;
    struct reftyp far *top;
    struct wordtyp far *next;
    int pos;
};

struct wordtyp far *root=NULL;

void set_data(char *r, char *m, int n, int f) {
    struct wordtyp far *p, far *pp;
    struct reftyp far *q, far *qq;
    int i;

    for (p=root,pp=NULL;p!=NULL;pp=p,p=p->next)
        if (far_strcmp(p->word,r)==0) {
            for (q=p->top,qq=NULL;q!=NULL;)
                if ((i=far_strcmp(q->val,m))==0) {
                    if (q->count<REF_MAX) {
                        q->p[q->count]=n;

```

```

        q->fn[q->count] = f;
    }

    q->count++;
    return;
} else if (i>0) {
    qq=q;
    q=q->left;
} else {
    qq=q;
    q=q->right;
}

q=getm(sizeof(struct reftyp));
q->p[0] = n;
q->fn[0] = f;
q->count=1;
q->left=q->right=NULL;
q->val=getm(strlen(m)+1);
far_strcpy(q->val, m);
if (qq==NULL)
    p->top=q;
else if (i>0)
    qq->left=q;
else
    qq->right=q;
return;
}

p=getm(sizeof(struct wordtyp));
p->next=NULL;
p->word=getm(strlen(r)+1);
p->pos=n;
far_strcpy(p->word, r);
if (pp==NULL) root=p;
else pp->next=p;
q=getm(sizeof(struct reftyp));
q->p[0] = n;
q->fn[0] = f;
q->count=1;
q->right=q->left=NULL;
q->val=getm(strlen(m)+1);
far_strcpy(q->val, m);

```

```

    p->top=q;
}

void read_data(int f) {
    char line [LINE_LENGTH], *p;
    int  n;

    if (f==0) printf("Read Start ");
    else      printf("Read Start [%d] %n", f);
    n=0;
    while (fgets(line, LINE_LENGTH, fpi) !=NULL) {
        if ((++n % 100) ==0)
            if ((n % 6500) ==0) puts(".");
            else                putchar('.');
        if (line[0] == '\n') continue;
        if ((p=strstr(line, ",")) !=NULL) {
            *p = '\0';
            p=p+2;
            set_data(line, p, n, f);
        } else {
            printf(" %n *** No 、 at %d *** %n %s", n, line);
            fprintf(fpo, " %n *** No 、 at %d *** %n %s", n, line);
        }
    }
    fclose(fpi);
    if (f==0) printf(" Read end. %n Total %d lines. %n", n);
    else      printf(" Read end [%d]. %n", f);
}

char tango [LINE_LENGTH]; /* for Stack Over Flow */

void write_ref(struct reftyp far *q) {
    int  i, j;

    if (q->left !=NULL) write_ref(q->left);
    far_strcpy(tango, q->val);
    tango[strlen(tango)-1] = '\0';
    fprintf(fpo, "%6d : %s =", q->count, tango);
    if (q->count > REF_MAX) j=REF_MAX-1;
    else                    j=q->count-1;
}

```

```

    for (i=0;i<j;i++)
        if (q->fn[i]==0) fprintf(fpo," %d,", q->p[i]);
        else
            fprintf(fpo," %d-%d,", q->fn[i], q->p[i]);
    if (i+1==q->count)
        if (q->fn[i]==0) fprintf(fpo," %d. %n", q->p[i]);
        else
            fprintf(fpo," %d-%d. %n", q->fn[i], q->p[i]);
    else
        if (q->fn[i]==0) fprintf(fpo," %d... %n", q->p[i]);
        else
            fprintf(fpo," %d-%d... %n", q->fn[i], q->p[i]);
    if (q->right!=NULL) write_ref(q->right);
}

void write_data() {
    struct wordtyp far *p;
    int n = 0;

    printf("Write start ");
    for (p=root;p!=NULL;p=p->next) {
        if ((++n % 5)==0) puts("");
        printf("%s, ", far_strcpy(tango, p->word));
        fprintf(fpo,"%s = ", tango);
        fprintf(fpo,"%d. %n", p->pos);
        write_ref(p->top);
    }
    fclose(fpo);
    printf(" Write end. %n");
}

main(int argc, char *argv[]) {
    int i;

    if (argc<=3) {
        if (!init(argc, argv)) exit(1);
        read_data(0);
    } else {
        for (i=1;i<argc-1;i++) {
            if (!ropen(argv[i])) exit(1);
            read_data(i);
        }
    }
#ifdef LSI_C
    printf("Memory Left = %ld byte. %n", farcoreleft());
#endif
}

```

```
#endif
    }
    if (!wopen(argv[argc-1])) exit(1);
}
write_data();
exit(0);
}
```

3.2 init.h

```
/* Header file for init.c */
```

```
#ifdef LSI_C
#include <dos.h>
#include <process.h>
#include <farstr.h>
#else
#define far
#define far_strcmp(x,y)    strcmp(x,y)
#define far_strcpy(x,y)    strcpy(x,y)
#define farmalloc(x)      malloc(x)
#define far_strncpy(x,y,z) strncpy(x,y,z)
#define far_strncmp(x,y,z) strncmp(x,y,z)
#define far_strstr(x,y)    strstr(x,y)
#endif
```

```
int init(int argc, char *argv[]);
```

```
void far *getm(unsigned long size);
```

```
void freem();
```

```
int ropen(char *fni);
```

```
int wopen(char *fno);
```

3.3 init.c

```
/* */
/* ファイルオープン & メモリアロケーション */
/* by K.Miki */
/* 94/05/12 ver. 1 */
```

```
/* 94/05/13 + freem() */
/* 95/04/28 to UNIX */
/* 99/08/06 wopen, ropen from checkl */
/* */

#include <stdio.h>
#include <string.h>

#ifdef LSI_C
#define BUFF_SIZE 50000
#else
#define BUFF_SIZE 500000
#endif

#include <stdlib.h>
#include "init.h"

extern FILE *fpi, *fpo;

int init(int argc, char *argv[]) {
    char  fni [60], fno [60];

    if (argc>=2)
        strcpy(fni, argv[1]);
    else {
        printf("Input File Name = ");gets(fni);
    }
    if ((fpi=fopen(fni,"r"))==NULL) {
        printf("There is no %s.¥n", fni);
        return (0);
    }
    if (argc>=3)
        strcpy(fno, argv[2]);
    else {
        printf("Output File Name = ");gets(fno);
    }
    if (strcmp(fni, fno)==0) {
        printf("You can't use same file.¥n");
        fclose(fpi);
        return (0);
    }
}
```



```

    }
    if ((fpo=fopen(fno,"w"))==NULL) {
        printf("Can not open %s. \n", fno);
        fclose(fpi);
        return (0);
    }
    return (1);
}

char far *getm_p, far *getm_start;
unsigned long getm_rest = 0;

void far *getm(unsigned long size) {
    char far *q;

#ifdef LSI_C
    size=((size+7)/8)*8;
#endif
    if (size>BUFF_SIZE) {
        printf("Too Big Request %d byte. \n", size);
        exit(1);
    }
    if (size>getm_rest) {
        getm_p=getm_start=farmalloc(BUFF_SIZE);
        if (getm_p==NULL) {
            printf("Sorry, No more memory. \n");
            exit(1);
        }
        getm_rest=BUFF_SIZE;
    }
    q=getm_p;
    getm_p+=size;
    getm_rest-=size;
    return (q);
}

void freem() {
    getm_p=getm_start;
    getm_rest=BUFF_SIZE;
}

```

```

int ropen(char *fni) {
    if ((fpi=fopen(fni,"r"))==NULL) {
        printf("There is no %s. %n",fni);
        return (0);
    }
    return (1);
}

int wopen(char *fno) {
    if ((fpo=fopen(fno,"r"))!=NULL) {
        printf("Output file %s exist. %n",fno);
        fclose(fpo);
        return (0);
    }
    fclose(fpo);
    if ((fpo=fopen(fno,"w"))==NULL) {
        printf("Can not open %s. %n",fno);
        return (0);
    }
    return (1);
}

```

このプログラムを用い、参考例として、現在校正を進めている昭和20年から昭和63年までのデータにおける形態の部分の頻度を“CHECK1”で求めた。

頻度は「上演数：番組名＝データ名（最初の5つを表示。データがフロッピー8枚に分割して納めてあるので、例えば下記の引用の第1行目の傍線部1-1810は「翁」という形態の能が1枚目のフロッピーの1810行にあることを示す。）」で示した。

```

74：翁＝1-1810, 1-5603, 1-22460, 1-32979, 1-41888……
5316：能＝1-10, 1-58, 1-166, 1-176, 1-197……
7：袴能＝3-26506, 3-26541, 3-37444, 3-45102, 6-66715……
11330：素謡＝1-77, 1-96, 1-115, 1-139, 1-144……
24828：囃子＝1-7, 1-121, 1-129, 1-187, 1-403……
28757：仕舞＝1-19, 1-23, 1-27, 1-36, 1-40……
4：脇仕舞＝6-83082, 6-83092, 6-83239, 6-83249
3：脇謡＝5-80070, 6-83102, 8-3974
3210：独吟＝1-111, 1-411, 1-511, 1-639, 1-706……
3495：連吟＝1-1073, 1-1078, 1-2069, 1-2783, 1-2789……
938：一調＝1-415, 1-722, 1-845, 1-1237, 1-1242……
0：独調・独鼓

```

575：連調＝1-14860, 2-693, 2-5033, 2-5243, 2-7864……
19：一調一管＝2-15597, 3-43432, 3-43463, 3-43503, 3-45096……
11：一調一声＝1-19556, 1-32646, 2-26758, 2-50277, 3-6042……
52：一管＝1-19547, 2-71371, 2-78292, 2-86180, 3-39764……
74：連管＝1-19551, 2-71375, 2-71446, 2-71561, 3-39799……
3637：狂言＝1-31, 1-52, 1-170, 1-191, 1-1468……
0：袴狂言
20：狂言語＝2-67623, 3-41303, 4-44519, 4-46182, 5-60388……
37：小謡＝1-361, 1-676, 1-680, 1-1065, 1-1069……
520：小舞＝1-70090, 1-70094, 1-70098, 1-75221, 1-83463……

以上は形態として認めたものであるが、このうち、「独調・独鼓」はタグとして廃し、独調とした。また形態と認めていないものも以下のように検出され、これらについてもタグとして認めるかを考える必要が生じた。この一部分を引用すると、

10：*仕舞＝4-76826, 4-76834, 4-76842, 4-76850, 4-76858……
5：*囃子＝8-42444, 8-49846, 8-53326, 8-53650, 8-53662

となる。このようにして発見したものの内、「一管独吟」「連調連管」を新しいタグとして認め、他は従来のタグに分類しなおすこととしてデータの方のタグの訂正を行った。また「半能」「能（半能）」は、形態としては能に含めたが、備考というタグを作り、そこに「半能」と入力することとした。“CHECK1”はこのようなタグの誤りを見つけるのに非常に便利なソフトで、能楽番組データベース以外にも汎用性がある。

4. 能番組をデータ入力する際の問題点

能楽番組のデータは大量であり、これを一人で入力することは到底不可能である。このため業者に委託したのだが、この過程でも様々な問題が生じた。最も大きな問題は、タグの統一性を維持する方法である。データベースを作成し始めた当初は、タグの名称について十分な配慮をしていなかった。入力が続けるうちに、例えば「上演演目」というタグは「演目」と短くした方が入力費が安く合理的であることや、「独調」「独鼓」のように同じ形態であるのに、番組上で名称が異なる場合があることなどが判明してきた。これらについては、そのつど文書で業者に連絡したが、業者内での連絡が徹底せず、最終的には“CHECK1”でその不統一を見つけ、個々に訂正する方法を取らざるを得なかった。実際に入力する人数が多くなる仕事ほど、タグの統一性は保ちにくく、この点は今後も課題として残った。

もう一つ問題となるのは、上演形態が年々変化しており、入力する番組によってはそれに対応するタグを次々と考えて行く必要があるということである。例えば、小鼓・大鼓・太鼓などが複数で、小数の地謡に合わせて演奏する「連調」という形態がある。この形態は戦前の番組には見られないが、戦後の番組において初めて現れたものである。

また、形態が番組に記されていない場合、どのような形態と認定して入力するかも問題

となる。例えば、小鼓・大鼓・太鼓のいずれかのアシライで一人の謡に合わせる「一調」という形態がある。これは囃子方一人、謡一人で、どちらも完全に詞章・節を覚えておかななくてはならないため、囃子方・謡のともに相当の技量が要求される。一調は能一番に相当するほど重いものとして、大事に扱われており、玄人でさえもこれが出来る人は相当の技量があると判断できる。ところが、これと同じ形態であっても、「独調」「独鼓」と呼ばれる時は素人が行う場合もある。つまり、同じ形態で三種の呼び名があり、番組面は同じだが、そのどちらと認定するかでその演奏をした人の技量が全く異なるものと判断される場合がある。このような場合は、行っている人が玄人であるか素人であるか、またその人の技量がどの程度であったかを判断して入力する必要がある。また、独調・連調の場合は、その演目が小鼓によるのか、大鼓によるのか、もしくは太鼓によるのかを判断して入力する必要がある。これなども、小鼓の先生の社中の素人会であれば、この連調は小鼓であろうなどと、催しの性格や、その演目を担当する人が何を誰に習っていたかなどを判断して入力しなければならない。つまり、入力する人に、能楽に関する知識と、その人物に関する知識の両方が要求されるのである。これらについては、業者には全く要求できないので、自分で他の番組・文献等を参考に調べるか、能楽師の方などに尋ねる以外に方法がない。校正の際も同様である。その意味で能楽と、その時代の能楽界に詳しい人の協力が不可欠である。本データベースの校正は大倉流大鼓方の寛鉾一師が担当して下さった。寛師の協力がなければ、このデータベースの精度は到底実用に堪えなかったであろう。

また、新所作・新作狂言など、能楽の様式を用いて新たに新しい形式を生み出す場合がある。例えばそのような形式に「能舞」がある。能と同じ面・装束を用いて舞うので、袴装束で行う「囃子」ではない。しかしながら、ワキ方が登場しないから「能」とは言えない。このようなことで「能舞」と呼ばれるのだが、この形態も近年名古屋で上演されるようになった。また、新所作《空海》では、能の囃子である笛・小鼓・大鼓・太鼓以外に、僧侶による「声明」が加わった。このような場合、「声明」というタグを作成し、担当者を記す必要があるだろう。このような能の形式を用いながらも、新しい効果や他の芸能を取り入れた芸能を扱う際には「どこまでを能・狂言と認めるか」という難しい問題がある。明治維新直後に三味線を用いた「吾妻能」「照葉能」「照葉狂言」と言った芸能が流行したが、これらは「能」としては扱われていない。新作能の際に他の芸能を取り入れると言う試みは今後も続くと思われる。データベースの拡充を考える上で能楽の範囲をどのように定めるかも大きな問題となるだろう。

5. 今後の拡充目標とデータベースの利用目的

平成6年に能楽協会名古屋支部より関係者に配布された資料²⁾によれば、この番組の整理のみならず、以下のようなデータを収集したいと考えていた。

一、各人について以下の情報を出来る限り収集したいと思います。

- ①本名、芸名、幼名等、②生没年（没年令）、③職業（役職）、商売、屋号等、④所属役名、及び流儀、⑤師弟・親子・血縁関係、⑥主宰および所属団体、⑦舞台活動、舞台歴（初舞台・披露等）、⑧著作及び参考資料があれば資料名、⑨特記事項（表彰、公職、そ

の他), ⑩現住所, 或いは縁者(子孫等)の連絡先, その他本人(故人)にまつわるエピソードの類

「近代名古屋の能楽を支えた人」についての記録ということであれば, 能番組の記録とその索引のみでは不十分であり, これらの項目についても調査をすすめる必要がある。またこれらも何らかの形でデータとして入力し, 番組データベースとリンクさせる必要がある。またこのデータベースをどのように活用して欲しいかと言うことも, 使用者にきちんと説明すべきであろう。前掲の能楽協会名古屋支部からの配布資料には

一. この基礎資料をデータベースとしてすべて入力することにより,

- ①どのような人々が名古屋の能楽界を支えて来たか。
 - ②名古屋ではどんな曲の上演が多いか, またどんな曲が上演されていないか。
 - ③どの時期にどんな曲の上演が増えたか, その要因は何か。
 - ④どの人とどの人との共演が多いか。
 - ⑤どの楽師の来演が多いか, またどんな関係で来演するようになったのか。
- などの事象を調べる事が可能となります。

一. 中京・名古屋能楽界の実態を正確に把握する資料として, これをもとに,

- ①名古屋であまり上演されていない曲を選びたい。
- ②名古屋来演の少ない人に演じてもらいたい。
- ③あまり共演の見られない顔合わせで企画したい。

など, 今後の能楽運営に際しても有益な検討材料となります。

とある。東海地域能楽史の資料というのみならず, 能楽を支えた人の顕彰や, 能楽催しの企画にも役に立つことを意図している。データベースはその原則としては, 一般に公開すべきものであろう。しかしながら, この能楽番組データベースは個人のプライバシーにかなり関係する情報となるので, 公開の範囲・方法には相当の配慮が必要となる。これらについても, 今後検討を続けて行きたい。

注

- 1) 『東海地域能楽番組一覧(明治元年～昭和二六年)』深谷哲・三木邦弘・寛鈺一他編, 東海能楽研究会, 平成9年7月発行
- 2) 同注1, 巻末収載

参考文献

- 「能楽研究と情報——東海地域能楽データベースを中心に——」『生活の科学』20号, 17-28頁, 相山女学園大学生生活科学部, 平成10年5月発行
- 「冊子作成システムについて」三木邦弘・飯塚恵理人, 『近代東海地域能楽史の研究——能楽資料の調査・収集とデータベース化』平成7・8年度科学研究費補助金〔基盤研究(C)〕研究成果報

飯塚恵理人・三木邦弘・深谷 哲

告書（課題番号 07610438），研究代表者 飯塚恵理人
『平成 5 年版 東海能楽年鑑』東海能楽研究会編，能楽の友社出版部，平成 6 年 10 月発行
『近世能楽史の研究——東海地域を中心に——』飯塚恵理人，椋山女学園大学研究叢書 1，雄山
閣，平成 11 年 2 月発行

付記 本研究は平成 11 年度椋山女学園大学学園研究費助成(A)の成果の一部となります。