資　料

バスケットボール用作戦支援システムの開発（1）

—スコア・データベース参照プログラム—

児玉善廣[*1]・鈴木敏明[*2]・吉田祐子[*3]

1．はじめに

バスケットボール競技においては，技術や戦術の高度化にともなって，事前に対戦相手チームのフォーメーションや攻守パターンを計数的に分析し，それに対応する自チームのゲームプランを数値目標の形で立てたうえで試合に臨む必要性がますます高まってきている。そのような傾向は，競技力水準の高くなるほど顕著である。

先に筆者らは国際レベルの大会のスコア記録の構造分析を行い，チームの戦力を構成する主要な3次元を抽出した(児玉・鈴木, 1986; 1990; 鈴木・児玉, 1988; 鈴木, 1994)。しかし，それらは静的な分析に基づいてチームの攻守パターンを要約的に描き出すことには成功しているものの，既に終了したゲームの構造的特徴を事後的に記述するにとどまるものであった。

現在筆者らは，上述の成果を踏まえつつ，ゲームの進行にともなって時系列的に変化する自チームと相手チームの攻守パターンの解析結果を即時にデータベースに反映させ，提示できるリアルタイム型の作戦支援システムの開発を進めている。その成果は今後順次公表することとしているが，本稿では，スコア・データベースを参照するためのモジュールの概要について報告する。

2．処理システムの構成

作戦支援システムは，ベンチサイドに持ち込んで使用することになるので，可搬性，処理速度，ファイル容量，操作性等の要素について総合的に検討した結果，Apple社製のラップトップ型パソコンである Macintosh PowerBook Duo 270c を採用した。この機種は CPUがMC68030(33Mhz)でFPUを搭載しており，ベンチサイドでの使用にストレスを感じさせない十分な処理速度を実現できる。OS は漢字 Talk 7 で，操作方法はGUI環境下で統一されており，DOS ベースのマシンに比べて格段に操作感が良い。表示装置はカラーTFT液晶を使用しているので，視認性が極めて良い。メモリ容量は標準で12MBあるので，複数のアプリケーションやプログラミング言語を同時に開くことができる。ファイル容量も240MBで，複数の大会の数年分のスコア記録を余裕を持ってデータベース化できる大きさを備えている。また，標準で Express モデムを内蔵しているので，本体に格

＊1　仙台大学体育学部　コーチング・コース
＊2　仙台大学体育学部　スポーツ心理・行動学系
＊3　東北大学大学院医学系研究科　障害科学専攻

納しきれないデータを参照したり，多変量解析などで高速の処理を必要とする場合には，遠隔地からでも通常の電話回線を介して本学のワークステーション群と連携をとることが可能である。さらに重量は2.2kg と極めて軽量であり，バッテリー駆動時間も2～4時間と可搬性・移動性に優れている。本体の拡張性については，MiniDock というアダプターを介することで確保してある。

筆者らが開発中の作戦支援システムは，プログラミング言語とアプリケーションを組み合わせて構成することになるので，現時点では，言語としては Pascal，C++，アプリケーションとしては Excel(表計算)，Jterm(通信)，ResEdit(リソース・エディタ)，Edit7(テキスト・エディタ)，Mac-WORD(JWP)，corkboard (アイデア・プロセッサ)，Atok8 (FEP) 等をインストールしてある。

3．スコア・データベースのレコード形式

スコア記録は大会ごとに一括され，1ファイルにテキスト・データとして格納される(図1)。

```
11          ←―― 大会参加チーム数
CHN         ←―― チーム名
KOR
JPN
TPE
HKG
KAZ
KGZ
THA
MAS
PHI
INA
1 4         ←―― チーム・コードと選手の登録番号
Zhang_Wei_Juan      ←―― 選手名
1 3 3 8 5 6 5 3 7 1 3 2 0 37  ←―― スコアデータ (14項目)
-99         ←―― セパレータ (整合性確認用)
1 5
Li_Xin
1 1 15 29 4 5 8 3 3 4 17 5 0 99
-99

    (途中省略)

11 13
Merciana_Anggani
0 0 18 38 8 14 12 10 7 4 8 5 1 155
-99
11 14
Tuti_Gunawan
2 15 5 10 1 1 5 3 10 3 5 2 1 74
-99
11 15
Yenny_Siswanto
0 2 8 13 10 14 12 4 6 6 2 1 0 64
-99
-99 -99     ←―― 終了判定用データ
```

図1　スコア・データベースのレコード形式

ファイルの第１行目には大会参加のチーム数が入っている。２行目以降にチーム名がチームコード順に配列される。それに続いて各選手のデータが４行にわたって順次配列される。選手データの第１行目はチームコードと選手の登録番号である。第２行目は選手名，第３行目は14 項目のスコアデータからなる。それらは①３点シュート投射本数，②同成功本数，③２点シュート投射本数，④同成功本数，⑤フリースロー投射本数，⑥同成功本数，⑦パーソナル・ファール/チーム・ファール本数，⑧オフェンス・リバウンド獲得数，⑨ディフェンス・リバウンド獲得数，⑩ターン・オーバー数，⑪アシスト数，⑫スティール数，⑬ブロック・ショット数，⑭出場時間数，である。4行目はレコードのセパレータ(-99)である。ファイル終端にはEOFのマークとしてチームコード -99，選手コード-99 がセットされている。

このような形式に従って，大会ごとに作成されたデータ・ベースをプログラムで選択して抽出するのである。図１で例示したのは，1994年４月から５月にかけて仙台市で開催された第15回女子アジアバスケットボール選手権大会の全スコア記録を収録したデータベースである。

４．データベース参照プログラムの概要

参照プログラムは SYMANTEC社製のTHINK Pascal によって記述されている。THINK Pascal はANSI Pascal 規格を包含しており，Macintosh ToolboxおよびOSルーチンを拡張サポートしている。プロジェクト・マネージャが，関係するすべてのファイルを制御し，専用エディタ，コンパイラ，リンカ，ソースレベル・デバッガなどのプログラミングに必要なすべてのツールを統合化された環境下で使用することができる

１）プログラムの構造

図２－１～８はデータベース参照プログラムのソースコードである。全体の処理の流れは，指定されたデータファイルを入力した後，所定の集計処理を行い，問い合わせに対して応答する形になっている。以下ではソースコードに沿って説明を加える。

２）宣言部

図２－１はラベルと定数の宣言部である。999は処理終了の際の飛び先である(図２－８の最後部)。

nameOfPlayer と nameOfTeam は，それぞれ選手名とチーム名の最大文字数，firstNumberOfPlayer，lastNumberOfPlayer，lastNumberOfPlayerT は，登録選手の開始番号(4)と最終番号(15)および最終番号＋１(16)が設定されている。最終番号＋１は，スコアを格納する配列の中で，チーム合計を入れる位置を示すために使用する。

numberOfTeams，numberOfItems には最大チーム数(20)とスコア項目数(16)が設定されている。最大チーム数はデータベースとして登録する大会の規模に応じて変更してよい。スコア項目については，システム全体が先に示した14項目と合計得点，合計リバウンド獲得数を加えた16項目に固定して設計してあるので，変更してはならない。

scale1，scale2，scale3 は画面表示の際の項目用見出しとセパレータである。

teamTotal，noName は合計欄のタイトルと非登録選手名用のデフォルトのタイトルである。

図２－２は型宣言と変数宣言部である。

nameVector1 はチーム名格納用の２次元配列 namelistOfTeams のサイズを定義している。nameVector2 は選手名格納用の３次元配列 namelistOfPlayers のサイズを定義している。

```
program scoreReference (input, output, f);

 label
  999;

 const
  nameOfPlayer = 30;
  nameOfTeam = 5;

  firstNumberOfPlayer = 4;
  lastNumberOfPlayer = 15;
  lastNumberOfPlayerT = 16;

  numberOfTeams = 20;
  numberOfItems = 16;

  scale1 = 'id name                    pts 3pm 3pa 2pm 2pa ftm fta  fl rbo
            rbd  reb tur ast stl blk  min';
  scale2 = '-------------------------------------------------------------
            ----------------------------';
  scale3 = '=============================================================
            ============================';
  teamTotal = 'Team total                      ';
  noName = '                                  ';
```

図２－１　データベース参照プログラム（１）

```
 type
  nameVector1 = packed array[1..numberOfTeams, 1..nameOfTeam] of char;
  nameVector2 = packed array[1..numberOfTeams, firstNumberOfPlayer..
                lastNumberOfPlayerT, 1..nameOfPlayer] of char;
  fileName = packed array[1..8] of char;
  scoreForm = array[1..numberOfTeams, firstNumberOfPlayer..
              lastNumberOfPlayerT, 1..numberOfItems] of integer;

 var
  f: text;
  af: fileName;

  namelistOfTeams: nameVector1;
  namelistOfPlayers: nameVector2;

  scoreCube: scoreForm;

  teams: integer;
  tpg_m, tpg_a: integer;
  fg_m, fg_a: integer;
  ft_m, ft_a: integer;
  pts: integer;
  ptfl: integer;
  reb, reb_o, reb_d: integer;
  tur, ast, stl, blk, min: integer;

  i, j, k, l, m, n: integer;
  c: char;
  teamCode, playerCode: integer;
  teamId: integer;
  playerId: integer;
  separator: -100..-98;
```

図２－２　データベース参照プログラム（２）

fileName は複数のスコアデータ・ファイルを扱う場合にファイル名 af を定義するために使用する。ただし現在はファイルが１つだけ（第15回女子アジアバスケットボール選手権大会のデータ）なので未使用である。

scoreForm はスコアデータ格納用の３次元配列 scoreCube のサイズを定義している。

f はスコアデータのファイルである。プログラム中では read，readln 用の入力チャンネルとして用いる。今回の例では，abc94 というファイル名がプログラム中でアサインされている。

teams はデータベース・ファイルの先頭にセットされている実チーム数を保持する。tpg_m から min まではスコア項目の変数であり，デバッグを容易にするため，いきなり配列を使用するのではなく，それぞれ対応する名称がつけてある。teamCode，playerCode はデータベースからチームコードと選手コードを取り込む変数である。teamId, playerId は参照すべきチームおよび選手を指定する際に使用する。separator は，ファイルの整合性チェック用の分離コードを読み込む変数であり，常に -99 が正常値である。

３）チーム名一覧表示用ルーチンの宣言

図２－３は，指定すべきチーム名の一覧表を表示するための手続宣言部である。このルーチンは参照手順の最初で必ず引用される。これによって表示されるチーム名は，データベース・ファイルの先頭部分に書き込まれているものがそのまま用いられる。今回のプログラムでは，ほとんどの手続がメインプログラム中に展開されているが，今後，パラメータと方法が確定した部分の手続については，メインプログラム中に直接コードを記述するのではなく，このようにモジュール化していくことになる。

４）初期値設定部

図２－４は，入力ファイルのオープン，選手名格納用配列（namelistOfPlayers）の初期設定，チーム名をチーム名格納用配列（namelistOfTeams）へ読み込むための手順を記述したものである。

５）スコアデータの読み込み部

図２－５は，チームコード，選手コードを読み込み，それに従って，後続の選手名とスコア項目をそれぞれの格納用配列のセルにセットする手続を示している。チームコードがマイナスとなった場合がファイルの終端であり，入力手続を終了して次のステップに進む。

６）集計処理部

図２－６は各スコア項目についてチームごとの合計値を計算する手順を示している。合計値は scoreCube の下端部（lastNumberOfPlayerT）に集積される。現在は単純集計のみを行っているが，シュート成功率の計算やスコアの標準得点化などの機能を追加する予定である。

```
procedure teamMenu (var namelistOfTeamsD: nameVector1; nameOfTeamD:
                    integer; var teamsD: integer);
 var
  i, j: integer;
begin
 writeln;
 writeln(scale3);
 for j := 1 to teamsD do
  begin
   write(j : 2, ':');
   writeln(namelistOfTeamsD[j]);
  end;
 writeln
end;
```

図２－３　データベース参照プログラム（３）

```
    begin

     open(f, 'abc94');
     readln(f, teams);

     for i := 1 to numberOfTeams do
      begin
       for j := firstNumberOfPlayer to lastNumberOfPlayer do
        begin
         namelistOfPlayers[i, j] := noName
        end;
       namelistOfPlayers[i, lastNumberOfPlayerT] := teamTotal;
      end;

     for j := 1 to teams do
      begin
       i := 1;
       c := '*';
       while c <> ' ' do
        begin
         read(f, c);
         namelistOfTeams[j, i] := c;
         i := i + 1;
        end;
       while i <= nameOfTeam do
        begin
         namelistOfTeams[j, i] := ' ';
         i := i + 1;
        end;
       readln(f);
      end;
```

図２－４　データベース参照プログラム（４）

```
readln(f, teamCode, playerCode);
while teamCode > 0 do
 begin
  i := 1;
  c := '*';
  while c <> ' ' do
   begin
    read(f, c);
    namelistOfPlayers[teamCode, playerCode, i] := c;
    i := i + 1;
   end;
  while i <= nameOfPlayer do
   begin
    namelistOfPlayers[teamCode, playerCode, i] := ' ';
    i := i + 1;
   end;
  readln(f);

  readln(f, tpg_m, tpg_a, fg_m, fg_a, ft_m, ft_a, ptfl, reb_o,
         reb_d, tur, ast, stl, blk, min);
  pts := tpg_m * 3 + fg_m * 2 + ft_m;
  reb := reb_o + reb_d;

  scoreCube[teamCode, playerCode, 1]  := pts;
  scoreCube[teamCode, playerCode, 2]  := tpg_m;
  scoreCube[teamCode, playerCode, 3]  := tpg_a;
  scoreCube[teamCode, playerCode, 4]  := fg_m;
  scoreCube[teamCode, playerCode, 5]  := fg_a;
  scoreCube[teamCode, playerCode, 6]  := ft_m;
  scoreCube[teamCode, playerCode, 7]  := ft_a;
  scoreCube[teamCode, playerCode, 8]  := ptfl;
  scoreCube[teamCode, playerCode, 9]  := reb_o;
  scoreCube[teamCode, playerCode, 10] := reb_d;
  scoreCube[teamCode, playerCode, 11] := reb;
  scoreCube[teamCode, playerCode, 12] := tur;
  scoreCube[teamCode, playerCode, 13] := ast;
  scoreCube[teamCode, playerCode, 14] := stl;
  scoreCube[teamCode, playerCode, 15] := blk;
  scoreCube[teamCode, playerCode, 16] := min;

  readln(f, separator);
  readln(f, teamCode, playerCode);
 end;
close(f);
```

図２－５　データベース参照プログラム（５）

```
 for i := 1 to numberOfTeams do
  begin
   for j := 1 to numberOfItems do
    begin
     for k := firstNumberOfplayer to lastNumberOfPlayer do
      begin
       scoreCube[i, lastNumberOfPlayerT, j] := scoreCube[i,
                 lastNumberOfPlayerT, j] + scoreCube[i, k, j]
      end;
    end;
  end;
```

図２－６　データベース参照プログラム（６）

7）問い合わせへの対応部

図２－７，８は問い合わせ要求に対する対応手順を記述した部分である。

図２－７はチーム全員のスコアが表示されるオプションに対応している。まずプロンプトに対して teamId がキーボードから入力されると(図３－１の上部)，実チーム数を越えていないかどうかがチェックされ，越えている場合には処理を終了するためにプログラム終端の 999 という空文が設定されているラベルに飛ぶ(図３－２の最下行)。teamId が実チーム数内である場合には，対応するチームの全選手名が一覧表の形で表示され，スコアを表示すべき選手のコード(4～15)を入力するようプロンプトが表示される(図３－１の中段部分)。それに対して 99 が入力された場合は全選手のスコアを表示する要求と解釈し，全選手とチーム合計のスコア一覧表が表示される(図３－１下段部分)。

```
teamMenu(namelistOfTeams, nameOfTeam, teams);
write('Enter TeamId   (end=99)  ');
readln(teamId);
if teamId > teams then
 begin
  goto 999
 end;
writeln;
writeln('Team name: ', namelistOfTeams[teamId]);
for i := firstNumberOfPlayer to lastNumberOfPlayer do
 begin
  writeln(i : 2, ' ', namelistOfPlayers[teamId, i])
 end;
writeln;
write('Enter PlayerId (all=99)  ');
readln(playerId);
writeln(scale2);
while teamId <= teams do
 begin
  if playerId > lastNumberOfPlayer then
   begin
    writeln('TeamId:', teamId : 3, ' - ', namelistOfTeams[teamId]);
    writeln(scale1);
    for i := firstNumberOfPlayer to lastNumberOfPlayer do
     begin
      write(i : 2, ' ', namelistOfPlayers[teamId, i] : 21);
      for j := 1 to numberOfItems - 1 do
       begin
        write(scoreCube[teamId, i, j] : 4)
       end;
      write(scoreCube[teamId, i, numberOfItems] : 5);
      writeln
     end;
    writeln;
    write('   ', namelistOfPlayers[teamId, lastNumberOfPlayerT] : 21);
    for j := 1 to numberOfItems - 1 do
     begin
      write(scoreCube[teamId, lastNumberOfPlayerT, j] : 4)
     end;
    write(scoreCube[teamId, lastNumberOfPlayerT, numberOfItems] : 5);
    writeln
   end
```

図２－７　データベース参照プログラム（７）

```
 1:CHN
 2:KOR
 3:JPN
 4:TPE
 5:HKG
 6:KAZ
 7:KGZ
 8:THA
 9:MAS
10:PHI
11:INA

Enter TeamId    (end=99)  3  ←――――――――――  3のJPN(日本)を指定



Team name: JPN
 4 Yuka_Harada                ←――――――――――  日本チームの全選手名一覧
 5 Kikuko_Mikawa
 6 Hiroe_Kakizaki
 7 Aki_Ichijo
 8 Chikako_Murakami
 9 Mikiko_Hagiwara
10 Takako_Kato
11 Takami_Takeuchi
12 Kagari_Yamada
13 Noriko_Hamaguchi
14 Taeko_Oyama
15 Akemi_Okazato

Enter PlayerId (all=99)  99  ←――――――――――  全選手のスコア表示を指定
```

TeamId: 3 - JPN

id	name	pts	3pm	3pa	2pm	2pa	ftm	fta	fl	rbo	rbd	reb	tur	ast	stl	blk	min
4	Yuka_Harada	16	1	4	4	6	5	6	5	1	6	7	9	27	4	1	98
5	Kikuko_Mikawa	32	10	18	1	1	0	0	3	1	0	1	1	2	3	0	49
6	Hiroe_Kakizaki	29	3	3	10	12	0	0	6	3	9	12	2	6	3	4	44
7	Aki_Ichijo	57	13	29	8	14	2	3	11	0	13	13	4	8	5	0	123
8	Chikako_Murakami	44	2	9	12	14	14	15	10	0	13	13	7	31	8	0	118
9	Mikiko_Hagiwara	94	7	18	31	58	11	11	13	6	12	18	7	5	1	1	164
10	Takako_Kato	98	1	1	37	65	21	26	13	9	24	33	16	10	5	2	145
11	Takami_Takeuchi	17	1	1	7	9	0	0	4	2	2	4	1	0	0	0	18
12	Kagari_Yamada	47	0	1	18	28	11	15	14	12	13	25	5	7	7	2	108
13	Noriko_Hamaguchi	16	0	1	7	9	2	4	4	2	3	5	0	1	0	0	23
14	Taeko_Oyama	43	3	4	13	21	8	12	6	2	15	17	2	9	4	2	92
15	Akemi_Okazato	18	4	6	3	6	0	0	2	1	0	1	2	2	2	0	18
	Team total	511	45	95	151	243	74	92	91	39	110	149	56	108	42	12	1000

図３－１　表示例（１）

図２－８は，選手コードの入力プロンプトに対して，特定の選手に対応するコードが入力された場合の手続に対応しており，その場合は，指定された選手のスコア記録のみが表示されることになる(図３－２)。

```
    else
     begin
      writeln('TeamId:', teamId : 3, ' - ', namelistOfTeams[teamId]);
      writeln(scale1);
      write(playerId : 2, ' ', namelistOfPlayers[teamId, playerId] : 21);
      for j := 1 to numberOfItems - 1 do
       begin
        write(scoreCube[teamId, playerId, j] : 4)
       end;
      write(scoreCube[teamId, playerId, numberOfItems] : 5);
      writeln
     end;
    teamMenu(namelistOfTeams, nameOfTeam, teams);
    write('Enter TeamId   (end=99)  ');
    readln(teamId);
    if teamId > teams then
     begin
      goto 999
     end;
    writeln;
    writeln('Team name: ', namelistOfTeams[teamId]);
    for i := firstNumberOfPlayer to lastNumberOfPlayer do
     begin
      writeln(i : 2, ' ', namelistOfPlayers[teamId, i])
     end;
    writeln;
    write('Enter PlayerId (all=99)  ');
    readln(playerId);
    writeln(scale2);

  end;
999:
end.
```

図２－８　データベース参照プログラム（８）

```
================================================================================

 1:CHN
 2:KOR
 3:JPN
 4:TPE
 5:HKG
 6:KAZ
 7:KGZ
 8:THA
 9:MAS
10:PHI
11:INA

Enter TeamId   (end=99)  1  ←――――――――――――  1 の CHN(中国)を指定


Team name: CHN
 4 Zhang_Wei_Juan
 5 Li_Xin
 6 Liu_Jun
 7 Wang_Fang
 8 Zheng_Dong_Mei
 9 He_Jun
10 Liang_Xin
11 Zheng_Hai_Xia
12 Xiang_Dong_Wen
13 Li_Dong_Mei
14 Ma_Zong_Qing
15 Zhan_Shu_Ping

Enter PlayerId (all=99)  11  ←――――――――――――  11 の Zheng Hai Xia  選手を指定

--------------------------------------------------------------------------------


TeamId:  1 - CHN
id name                 pts 3pm 3pa 2pm 2pa ftm fta  fl rbo rbd reb tur ast stl blk  min
11 Zheng_Hai_Xia         75   0   0  33  48   9  12  10  16  34  50  12   1   4   6   96

================================================================================

 1:CHN
 2:KOR
 3:JPN
 4:TPE
 5:HKG
 6:KAZ
 7:KGZ
 8:THA
 9:MAS
10:PHI
11:INA

Enter TeamId   (end=99)  99  ←――――――――――――  処理終了を指示
```

図３－２　表示例（２）

5．今後の方向について

今回紹介したプログラムは，極めて単純な参照機能のみを実現したものであり，実際の効用という点では，まだ不十分なプロトタイプの段階にあると言わなければならない。

今後の開発の方向としては，①表示項目の多様化と精選，②指定したキーによるソートや抽出，③表示のグラフ化，④ToolBox を用いた操作環境のGUI化，⑤リーグ戦のデータについては，標的チームの全選手を個体登録し，データファイル間でリレーショナルな参照・集計ができるようにすること，⑥ゲームの展開に即応できるスピーディーな入力方式の考案等を計画している。それらの経過については，順次報告することとする。

引用・参考文献

児玉善廣・鈴木敏明　1986　バスケットボールの競技力構造の分析—ユニバーシアード男子ソ連・アメリカ・日本の選手比較を基に。　仙台大学紀要，18，67-83.

児玉善廣・鈴木敏明　1990　バスケットボール・スコアの多次元構造。　Japanese Journal of Sports Science，9，5，272-279.

鈴木敏明・児玉善廣　1988　バスケットボール・パフォーマンスの構造的特性の分析—多次元尺度構成法のスコア分析への適用。　東北体育学研究，9，1，11-24.

鈴木敏明　1994　多変量解析。　寺田　晃・佐藤　怜「教育心理学統計・調査・実験」230-255，中央法規出版。