

# 舞踊譜 Labanotation に基づく 身体運動データ入力・編集・表示システムの開発

岡本賢一 八村広三郎

立命館大学 理工学部 情報学科

{okaken, hachi}@img.cs.ritsumeikan.ac.jp

中村美奈子

立命館大学 アートリサーチセンター

minako@fc.ritsumeikan.ac.jp

デジタルアーカイブでは、無形文化財としての、芸能や工芸における人間の身体動作もその対象として考えられている。本研究室ではこのような分野に応用可能な、人間の身体動作の入力、記述、記録、検索表示のための、総合的な身体運動処理システムを構築中である。ここでは、計算機システムによる身体動作の入力と記述の方法として、舞踊の分野で人間の動作を図形的な記号で記述するために使われている舞踊譜 Labanotation に基づいたデータ表現を用い、計算機システム内に人間の身体運動を記録し、再現する。本論文では、Labanotation 譜面のインタラクティブな入力・編集のためのシステム LabanEditor について述べる。LabanEditor によって、舞踊の身体運動の入力と編集、さらにはその結果を3次元CGで表示することが可能になる。

## System for the Processing of Human Body Movement based on Labanotation

Kenichi Okamoto Kozaburo Hachimura

Department of Computer Science

Ritsumeikan University

Minako Nakamura

Art Research Center

Ritsumeikan University

Today, intangible cultural asset like ballet and dance has been a target of digital archiving. Our laboratory is developing comprehensive data processing system to input, describe, record, search, and display human body movement. This system can record and reenact human body movement using the data format based on Labanotation, which has been used for recording human movement of dance with several types of graphical symbols. This paper describes LabanEditor, which is an interactive graphical editor for writing and editing Labanotation scores. By using LabanEditor, a user can input and edit human body movement of dance and also display animation of a human body model in 3D graphics.

### 1 はじめに

人間の身体動作を情報処理の対象として扱うことは、マンマシンインターフェースの高度化や、仮想現実環境の実現などの面で、ますます関心が高まってきている。身体運動の取り扱いには、身体運動の入力と記述（コード化）、データベース化、運動の表示の3つの課題があるが、複雑な身体形状の動きを入力し記述することが、特に問題である [1]。

また最近では、さまざまな歴史情報や美術品などの歴史文化資産をデジタル化し、マルチメディア技術によって、これらを後世のために保存し活用しようという、デジタルアーカイブが各所で構想されるようになってきている。このデジタルアーカイブでは、基本的には絵画や文様、歴史文書などの平面資料が対象となるが、焼物や工芸品などの立体物についても立体形状計測によりデジタル化して保存する試みがなされており、さらには、踊りや芸能、工芸などにおける無形文化、すなわち人間の身体運動についても対象にしようとしている。

我々は、以上のような観点から、人間の身体運動

の様子を情報処理の対象とし、これらをデジタル情報として保存・活用するための手法について、研究を行っている。

舞踊の分野では、ダンサーの体の動きを記述することが古くから検討されており、なかでも Labanotation と呼ばれる舞踊記述法はアメリカの舞踊界を中心に広く利用されている [2]。これは、音楽における楽譜のように、人間の動作を図形的な記号で記述するものである。Labanotation は、元来コンピュータへの入力のために開発されたものではない。また、複雑な身体運動にも対応できるようになっているため、誰にでもすぐに理解できるというものでもない。しかし、身体各部の動作が図形で表現されるため、一覧性があり、ある程度の学習により理解できるようになる。このような観点から、我々は、コンピュータによる身体運動の入力と記述の方法として、Labanotation に基づいたデータ表現を試みている [3][4]。

本報告では、Labanotation の考え方を身体運動表現の基礎として利用し、計算機システム内に人

間の身体運動を入力・記録するためのシステム LabanEditor と、その他の関連ソフトについて述べる。LabanEditor は Labanotation の譜面を、マウスを用いたインタラクティブなグラフィック操作で入力・編集を可能にするものである。作成・編集した譜面はプリンタに印刷出力できるのは当然であるが、このデータはファイルとして保存し、さらにこれを 3D-CG モデリングのための標準的な言語である VRML に変換して出力でき、これを利用してディスプレイ上で CG モデルによる身体運動として表示する。

Labanotation のコンピュータ化についての関連研究としては、アメリカの Dance Notation Bureau が開発した、Labanotation 譜面を作成するエディタ Laban Writer がある。しかし Laban Writer では譜面を作成し印刷する機能しかなく、また、実行環境も Macintosh 上に限定されているという制約がある。さらに、Laban Writer で編集した譜面を、舞踊の振付用ソフトとして開発された LifeForms のアニメーションへと変換して表示するという研究も現在行われている [5]。しかし、これには商用ソフト LifeForms を必要とするので、さまざまなプラットフォームでの動作が保証されているものではなく、表示機能も限定されたものになる。

本報告で述べる LabanEditor などのソフトウェアは JAVA で開発しているので、基本的に機種に依存せず、多様なプラットフォームで実行できるものとなっているところに特徴がある。

## 2 舞踊譜 Labanotation

### 2.1 Labanotation の譜面

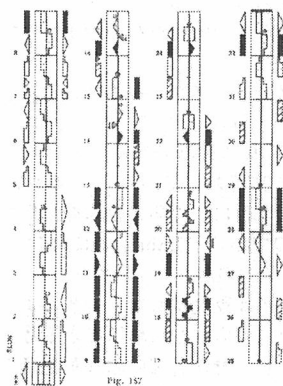


図 1: Labanotation の譜面の例 [2]

舞踊の分野では、振り付けの記録のための独自の記法が古くからいくつか考案され、利用されてきている。このような記法の中で、最も代表的なものが、1927 年に Rudolph von Laban によって発表された、

Labanotation である [2][6][7]。

図 1 は Labanotation の譜面の例である [2]。Labanotation は下から上へと読み進んでいく。譜面には時間の刻みを表す横線（小節線。楽譜では縦線に相当）が等間隔で記入されている。舞踊譜のそれぞれの列は、3本の縦線で区切られている。中央の線が体の中心を表し、中心線より右側が体の右側、左側が体の左側の各部分の動作を記述するために使われる。

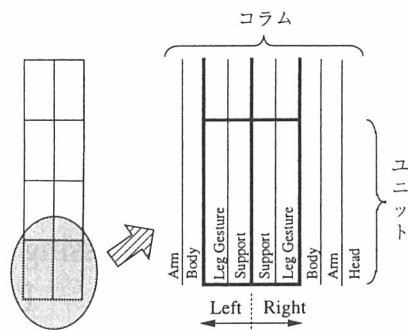


図 2: Labanotation のコラムの構成

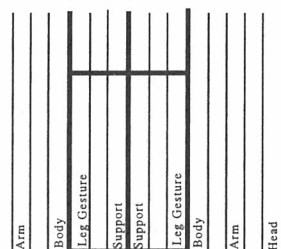


図 3: Expanded Staff のコラムの構成

3本の縦線によって構成される縦の列（コラムと呼ぶ）は、図 2 に示すように、人間の身体各部分に対応している。たとえば、左腕（Left Arm）に対応するコラムを下から上を読み進めていくことによって、左腕の動作が分かるようになっている。

コラムのうち、Support は体重を支えている体の部分（通常は足）の動きを表す。また、横線で区画された単位を、ここではユニットと呼ぶ。通常は図 2 のような構成の譜面（Standard Staff と呼ぶ）を用いるが、より複雑な運動を記述するために、Leg、Body、Arm、Head それぞれのコラム（総じて Gesture コラムと呼ぶ）の隣りに、より細かい動作表現を記述するためのコラムを加えた、Expanded Staff（図 3）と呼ばれる譜面が用いられる。

各部位の運動は、対応するコラムに書き込まれる図式記号（シンボル）の形状とその内部パターンの組み合わせで表される。

## 2.2 Direction シンボル

図4に Labanotation の図式記号の中で最も基本的な Direction シンボルを示す。シンボルの形状で水平面内の運動方向 (Direction という) を表し、垂直面内の運動方向 (Level という) はシンボル内部のパターンで表現される。また、運動が達成されるまでの時間がシンボルの長さによって表される。

Support コラムに書き込まれたシンボルによって体全体の移動方向を示すが、Support コラムが左右とも空白の場合は、体重が支えられていない、すなわち身体が空中にあることを意味する。

なお、Support コラムにおかれる、Direction シンボルは Support の移動方向を示しているが、これ以外のコラムにおいては運動の結果として最終的に達成される姿勢を示すという相違点がある。

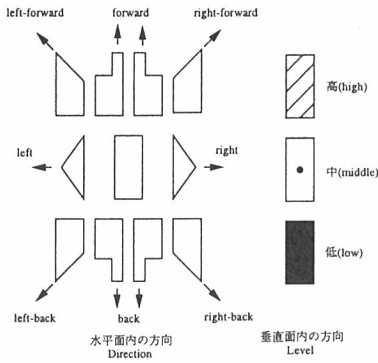


図4: Direction シンボル

## 2.3 付加記号

以上のような基本的なシンボルによって身体運動を記述するが、楽譜と同様に、Labanotation でも、さまざまな付加記号 (Sign) が使われる。ここでは、例として Rotation Signs についてのみ説明する。

Rotation Signs は、体の部分の回転を示すために使われ、Direction シンボルと同様に、シンボル形状とその内部パターンの組合わせで表される。Rotation Signs は、図5に示すように、シンボルの形状で回転方向を表し、シンボル内部のピンの方向で回転量を表す。

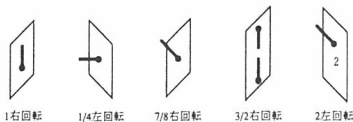


図5: Rotation Signs

## 3 Labanotation に基づく身体運動データ入力・編集・表示システム

Labanotation に基づいた身体運動データ入力・編集・表示システムの全体的な構成は、図6のようになる。以下にこの構成要素について述べる。

- 舞踊譜記述データ LND (LabaNotationData) システム内のデータ保存形式。Labanotation シンボルに対応する文字列によって記述される。
- 舞踊譜エディタ (LabaEditor) 本研究で開発している舞踊譜 Labanotation エディタ LabaEditor は、舞踊譜記述データ LND を入力し、主にマウスを用いてインタラクティブに Labanotation シンボルを編集し、LND を出力する。また、編集した譜面の LND を、表示システムに対応した VRML2.0 形式の身体運動データに変換する。
- 身体動作表示システム (MotionViewer) LabaEditor によって、LND から VRML2.0 形式に変換された身体動作データを表示するためのシステムである。Netscape などのウェブブラウザと CosmoPlayer などの VRML Plugin を用いることによって、様々なプラットフォームで動作を表示することができ、Java-EAI (External Authoring Interface) applet で身体動作表示のコントロールを行うこともできる。

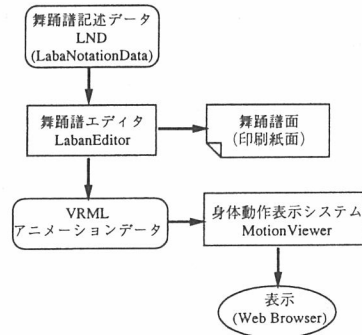


図6: Labanotation に基づく身体運動データ入力・編集・表示システムの概要

### 3.1 Labanotation に基づく舞踊譜記述データ LND

舞踊譜記述データ LND (LabaNotationData) は、Labanotation の記法を参考にした、システム内のデータ保存形式である。これは、Labanotation のシンボルに対応する文字列によって記述しようというものである。

LND は、コラムの1小節分のユニット単位で記述し、その中に各部位ごとのデータを記述する。以下に書式について説明する。

- 作成者や作成年月日などのファイル情報をコメントとして、“//”のあとに記述する。
- リズムを“#rhythm”のあとに記述する。リズムの種類としては、2/4、3/4、4/4、5/4、6/8をサポートしている。
- 1列あたり何ユニットの譜面であるかを“#unit\_per\_line”のあとに記述する。100を上限としている。
- ユニットの合計数を“#unit\_total”のあとに記述する。100を上限としている。
- 最初の姿勢を“#unit 0”の次の行から記述する。初期姿勢でなので開始時刻、終了時刻はない。
- “#unit (ユニット番号)”の次の行から各シンボルに相当するデータを記述していく。シンボルデータを表1に示す<sup>1</sup>。

表 1: LND のシンボルデータ

シンボルデータ	対応するシンボル
direction	Direction シンボル
rotation	Rotation Signs
body	Body Signs
pin	Relationship Pins
space	Space Measurement Signs
hold	Hold Sign

- シンボルデータの次に身体運動の部位(コラム)を記述する(図2参照)。
- 部位を記述したあとに、それぞれのシンボルデータに対して、方向・高さ・回転量などのデータを記述する。
- 開始時刻・終了時刻は、各リズムの1拍分を1.0としている。つまり、リズム2/4ならば1/4を1.0、リズム6/8ならば1/8を1.0にしている。
- 最後にそのシンボルのあるコラムが2分割されている(divided column)か、そうでないかを記述する。2分割されていないなら“standard”、divided columnで、シンボルが分割されたコラムの右側なら“divided.r”、左側なら“divided.l”と記述する。

<sup>1</sup> 舞踊譜のシンボルは種類が多く、細かく規定されている。そこで、Labanotation エディタや舞踊譜記述データ(LND)では使用できるシンボルを制限している。ここでは、参考文献[2][6]を参照し、これらの中で使用頻度の高いシンボルや重要と思われるシンボルに限定した。

舞踊譜記述データ LND の書式と記述例を以下に示す。

### 書式

```
// コメント
#リズム
#1列あたりのユニット数
#ユニットの合計数

#ユニット番号
direction コラム 方向 高さ 開始時刻 終了時刻 staff
rotation コラム 回転方向 回転量 開始時刻 終了時刻 staff
body コラム 関節 開始時刻 終了時刻 staff
pin コラム 方向 高さ 開始時刻 終了時刻 staff
space コラム 大小 大きさ 開始時刻 終了時刻 staff
hold コラム 開始時刻 終了時刻 staff
```

### 例

```
// Notated by K.Okamoto , Dec 1 2001

#rhythm 4/4
#unit_per_line 7
#unit_total 56

#unit 0
direction l_support place mid standard
direction r_support place mid standard
direction l_arm left mid standard
direction r_arm right mid standard

#unit 1
direction r_arm right high 0.00 1.00 standard
direction l_arm left high 0.00 1.00 standard
direction l_support place mid 0.00 1.00 standard
direction r_support left low 0.00 1.00 divided_r
direction r_support right low 0.00 1.00 divided_l
```

## 4 舞踊譜編集システム LabanEditor

舞踊譜 Labanotation エディタ LabanEditor を立ち上げると、図7のようなウィンドウが開く。これは編集する舞踊譜面の表示や各種コマンドを選択できるメインウィンドウである。このウィンドウ内にあるキャンバスにシンボルを置いたり、動かしたり、編集することによって舞踊譜面を描いていく。LabanEditor の機能の説明を以下に述べる。

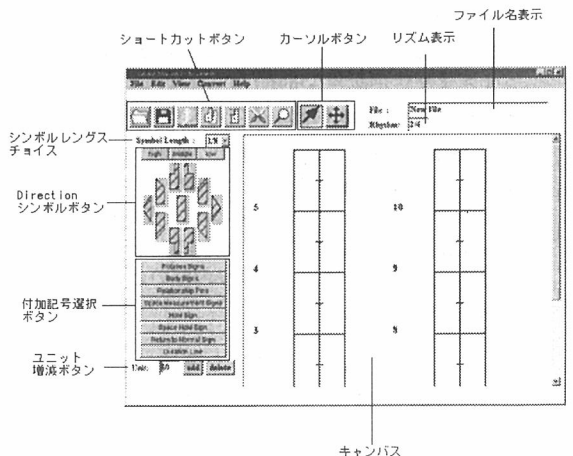


図 7: 舞踊譜エディタ LabanEditor の画面

## 4.1 LabanEditor のインターフェース

### 4.1.1 ファイル入出力

ファイルの新規作成は、メインウィンドウ (図7) にある File メニュー内の New File を選択する。この際、キャンバスのレイアウト設定ダイアログが現れるので、リズムを選択し、編集する譜面の1列あたりのユニット数を決定する。表示された譜面は、メインウィンドウのユニット増減ボタンでユニット数を増減させることができる。

ファイルのオープン、セーブはメインウィンドウにある File メニュー内の Open、Save、Save As を選択して行う。

### 4.1.2 シンボルの入力

各種シンボルの入力は、Direction シンボルボタン、付加記号選択ボタンの操作により入力する。

Direction シンボルの入力には、メインウィンドウ内の Direction シンボルボタン (図8) を使用する。

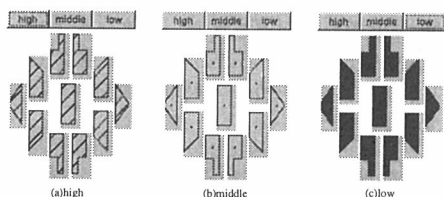


図8: Direction シンボルボタン

まず、シンボलगラスチョイスで入力したいシンボルの長さを選択する。次に Direction シンボルボタン上部の Level 選択ボタンをクリックして、Direction シンボルの Level (高さ) を選択する。Level ごとにシンボルボタンが用意されているので、その中から入力するシンボルを選択し、さらにキャンバス内の舞踊譜の適当な位置をクリックすると、そこにシンボルが入力される。

付加記号を入力するには、メインウィンドウ内の付加記号選択ボタンをクリックすることによって現れる付加記号ダイアログ内からシンボルを選択し入力する。付加記号は、Rotation Signs、Body Signs、Relationship Pins、Space Measurement Signs、Hold Sign がある。

また、各シンボルは重ね合わせて入力することによって、図9のように重ね合わされたシンボルが各々半分になり、コラムが分割される。これにより divided column として編集が可能となる。

### 4.1.3 譜面の編集

キャンバス内の譜面で、それぞれのシンボルについて、コピーやカット/ペースト、デリート、シンボルの移動、Undo や Redo などの編集操作をする

ことができる。これは、一般の Drawing ソフトとほぼ同様の機能である。また、編集中の譜面のレイアウト (1列あたりのユニット数) も変更することができる。

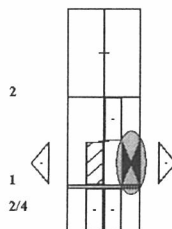


図9: divided column

## 4.2 譜面の印刷

入力・編集した舞踊譜面は、プリンタによって印刷することができる。図10は入力した Labanotation 譜面を印刷した例である。

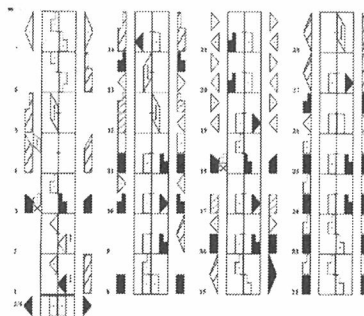


図10: 印刷した Labanotation 譜面

## 4.3 LabanEditor の内部構造

LabanEditor は図11に示すような部分から構成されている。

ユーザインターフェース部は、Java の AWT<sup>2</sup> を用いて実装している。ユーザは GUI 部品やイメージをマウスで操作し入力を行う。ここでは、それらのユーザの要求を、シンボルデータ解析部、ファイル入力部、ファイル出力部に渡す。

ユーザインターフェース部から送られてくるシンボルイメージに関するデータを解析し、シンボル名、開始時間、終了時間などのデータに変換したのち、シンボルイメージを作成する。キャンバスにはグリッドが設定してあり、決められた座標にのみシンボルを入力することができる。

舞踊譜表示部では、シンボルデータ解析部より作成されたシンボルイメージをキャンバス内のグラ

<sup>2</sup>Abstract Window Toolkit. AWT はボタンやメニューなどの GUI 部品、イメージ表示と、マウスやキーボードからの入力 (イベント) を処理する機能を持つ。

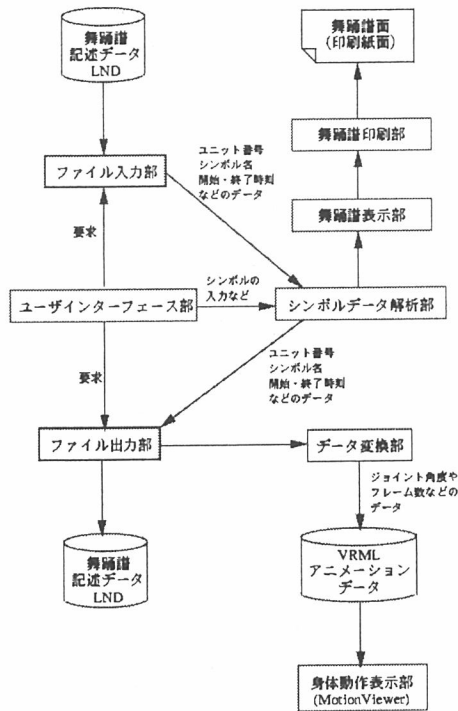


図 11: LabanEditor の構成

フィック画面上に表示し、舞踊譜印刷部では画面上のイメージを印刷する。

データ変換部では、現在キャンバスに表示されている舞踊譜面のデータを VRML アニメーションデータに変換する。身体動作表示部 (MotionViewer) は、この VRML アニメーションデータを読み込むことで身体動作を表示する。

#### 4.4 データ変換

入力・編集した Labanotation 譜面を身体動作として表示するために、VRML2.0 形式のアニメーションデータに変換する。システムメニューのポップアップメニュー Convert にある into VRML を選択すると、表示用のセーブダイアログがポップアップされる。ここでファイル名を入力するなどしてセーブファイルを決定する。

VRML ファイルに書かれる内容は、システムメニューのポップアップメニュー Convert 内 Options を選択することで様々な変更を加えることができる。変更項目には以下のものがある。

- 使用する身体モデルデータの指定  
身体モデルは身体構造を定義したファイルのほか、各セグメントの形状データのファイルなど

複数のファイルからなり、それらは一つのディレクトリにまとめられている。使用するモデルを変更する場合、このディレクトリ名を指定する。

- LND 変換テンプレートファイルの指定  
LND から身体動作データに変換するためのテンプレートファイルを選択する。デフォルトとして用意したものより精密な動きに変えたり、対応シンボルを増やしたりする場合、それを読み込んで変換に使用することができる。
- VRML スタンドアロン実行モードの設定  
ウェブブラウザでのアプレットの実行環境を利用できない場合には、VRML2.0 Plug-in のみで表示可能なファイルとして出力する。通常は再生速度の変更やループの ON / OFF はアプレットから操作できるようになっているが、スタンドアロン実行モードに設定すると、初期値があらかじめ身体構造ファイル (human.wrl、図 15 参照) に記述される。スタンドアロン実行モードに設定すると、ループを常に ON にし、1 サイクルにかかる時間を固定の値に設定する。

変換の処理の過程を図 12 に示す。変換には動作変換テンプレートファイルを読み込み、その内容にしたがって舞踊譜記述データ LND から身体動作データに変換する。

動作変換テンプレートファイルは舞踊譜のシンボルに対し、動作する関節名と pitch (x 軸回りの回転) roll (z 軸回りの回転) yaw (y 軸回りの回転) の回転角度とが、一対一に対応している。以下に書式と例を示す。

書式	例
#シンボル・サイン名	#direction
#コラム名	#r_arm
#舞踊譜記述データ LND	#forward high
関節名 回転角度	R_Shoulder -135,0,0
関節名 回転角度	R_Elbow 0,0,0
	⋮

上の例は、LND の記述

```
direction r_arm forward high
```

に対して、R\_Shoulder の pitch、roll、yaw の回転角度を  $-135^{\circ}, 0^{\circ}, 0^{\circ}$ 、R\_Elbow の pitch、roll、yaw の回転角度を  $0^{\circ}, 0^{\circ}, 0^{\circ}$ 、と設定することを表している。

#### 変換処理の過程

1. 指定されたテンプレートファイルを読み込み配列に格納する。
2. 舞踊譜記述データ LND ファイルをバッファリングしながら 1 行ずつ読み込む。
3. 読み込んだ一行に対してトークン化を行い、構文解析する。
4. トークンとテンプレート配列を比較し、対応する動作を関節数・回転角・フレーム数からなる配列に格納する。
5. 配列を元に VRML2.0 のアニメーションデータを生成する。
6. 生成したデータを指定したファイルにバッファリングしながら書き込む。

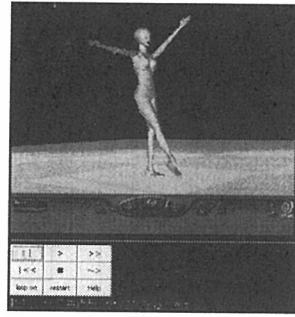


図 13: 身体動作表示システム MotionViewer

### 5.1 人体モデル

データ変換部で生成された VRML2.0 の動作データを、あらかじめ用意した身体モデルデータを用いて表示する。これは 3 自由度を持つ 21 のジョイントとセグメントからなる人体モデルで、図 14 のような階層構造を持っている。

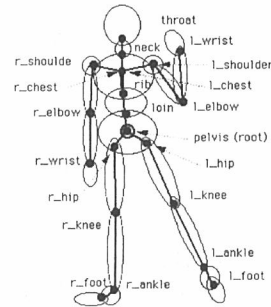


図 14: 人体モデルの階層構造

### 5.2 システムの構成

身体動作表示システムの構成を図 15 に示す。身体動作の表示のためのプラットフォームにはウェブブラウザと VRML2.0 対応プラグインを用いる。

図 15 内の実線矢印で示したように、関節定義、セグメント形状のデータ、アニメーションデータを外部ファイルから読み込む。このうち、アニメーションデータを定義している aPROTO.wrl が、データ変換部で生成されるファイルである。aPROTO.wrl は human.wrl から呼び出されるが、さらに、human.wrl からは関節定義ファイル jPROTO.wrl や各セグメントの形状を記述した VRML ファイル (\*.wrl) も読み込まれる。以上で一つの身体モデルのデータと、そのそのアニメーションデータとからなる、ひとつの VRML ファイルを形成する。

人体モデルの動作 (アニメーション) 制御は、Java

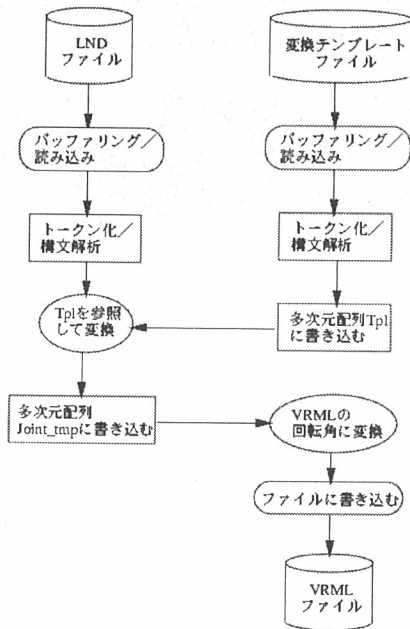


図 12: データ変換処理

ここで、VRML のアニメーションデータは、VRML の Interpolator (補間) ノード<sup>3</sup>のデータとして生成される。

## 5 身体動作表示システム MotionViewer

MotionViewer は、LabanEditor のデータ変換部によって LND から変換・出力された VRML2.0 形式のデータ読み込んで、身体動作を表示する (図 13)。

<sup>3</sup> ノードは VRML における命令の単位で、その構成要素をフィールドという。



アプレットを使用する。このため、ウェブブラウザに Java-EAI 対応プラグインを組み込んで利用する。

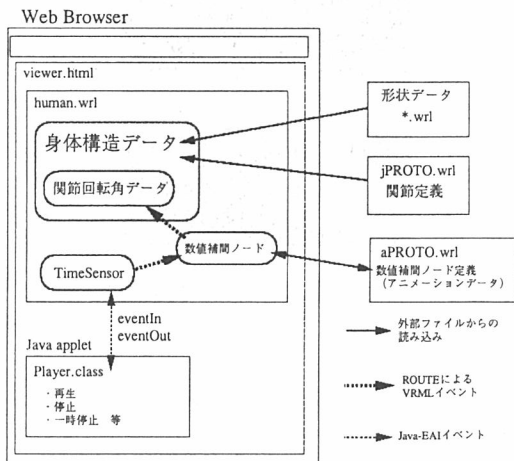


図 15: MotionViewer の構成

図 15 に示したように `viewer.html` から `Player.class` (フレームコントロールアプレット) と `human.wrl` (身体構造を記述した VRML ファイル) が呼ばれ、ブラウザ上に表示される。

Java applet から VRML ヘデータを送る場合は EAI の `eventIn` メソッドを、VRML の値の変化を applet が受け取るには、EAI の `eventOut` メソッドを使用する。

`TimeSensor` は、時刻を制御するノードで、`human.wrl` に書かれた `TimeSensor` は、はじめは停止した状態である。`TimeSensor` の変化は、数値補間ノードに伝えられ、数値補間ノードで計算された値を関節回転角データの変化として伝えることで、アニメーションを行う。フレームコントロールアプレット (図 16) は Java-EAI を用いて、`TimeSensor` に `eventIn` 命令を送ることでフィールドを操作し、再生、停止、早送り再生など、アニメーションを制御する。

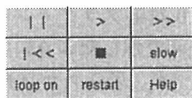


図 16: フレームコントロールアプレットのパネル

### 5.3 身体動作表示結果

LabanEditor で入力・編集した舞踊譜面を MotionViewer による 3DCG で身体動作として表示した例を図 17 に示す。

## 6 おわりに

本報告では、舞踊譜 Labanotation を身体運動の記述法の基礎として利用した身体動作情報処理シス

テムについて報告した。LabanEditor では、Labanotation の記述法の全てには対応していないものの、簡単な身体動作については入力・編集・表示が可能であり、初心者でも容易に舞踊譜面を編集できることが確認された。

本システムは、Java 開発環境 JDK (Java Development Kit) によって構築されており、多くのプラットフォームで実行できる。現在のところ、SGI (IRIX)、SUN (Solaris) および Windows マシン上での動作が確認された。このように本システムは広く普及している PC で利用できるので、Labanotation の教育、振り付けの指導など、マルチメディア教材としても利用できると考えている。

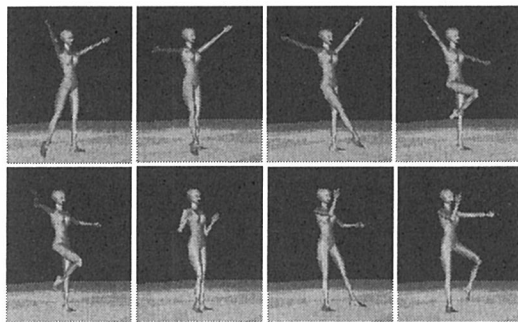


図 17: 身体動作表示結果

## 参考文献

- [1] 平松尚子、八村広三郎「身体動作の記述と表示のための実行制御環境の開発」、情報処理学会論文誌 第 40 巻 第 3 号, pp.939-948, 1999
- [2] Ann Hutchinson, "Labanotation", Theatre Arts Books, 1977
- [3] 吉田康行、松岡洋介、八村広三郎、「舞踊譜 Labanotation に基づく身体運動の処理-譜面読み取り LabanReader と譜面エディタ LabanEditor-」、情報処理学会人文科学とコンピュータ研究会報告 38-6, pp.61-68, 1998
- [4] Matsumoto, Hachimura, Nakamura, "Generating Labanotation from Motion-captured Human Body Motion Data", Proceedings of the International Workshop Recreating the Past-Visualization and Animation of Cultural Heritage, pp.118-123, 2001
- [5] Ilene Fox, "Documentation Technology for the 21st Century", World Dance 2000 Academic Conference: Papers and Abstracts, pp.137-142, 2000.
- [6] Peggy Hackney, Sarah Manno, Muriel Topaz, "Study Guide for ELEMENTARY LABANOTATION", A Dance Notation Bureau Press, 1977
- [7] Peggy Hackney, Sarah Manno, Muriel Topaz, "ELEMENTARY READING STUDIES", A Dance Notation Bureau Press, 1977