CHOREOSCRIBE:

AN "INTELLIGENT" EDITOR FOR CHOREOLOGISTS

Rhonda S. Ryman
Detlef O. K. Dransch
John C. Beatty

Introduction

The work described in this article is the extension of a project begun in September 1981 to produce a prototype system with which choreologists unfamiliar with the wizardry of computers can produce simple Benesh scores. The resulting system has been named "ChoreoScribe."[1]

Workstation Environment

At present, "ChoreoScribe" runs on a mainframe computer. The choreologist sits down at a workstation consisting of a colour *Display Monitor* (that looks much like a television set), a *Terminal* (a typewriter-like device with a keyboard and a screen), a *Puck* with four buttons, and a *Graphics Tablet* (a flat, sensitive surface on which the puck is moved, and which continuously transmits the puck's movement to the computer). (See Figure 1.)

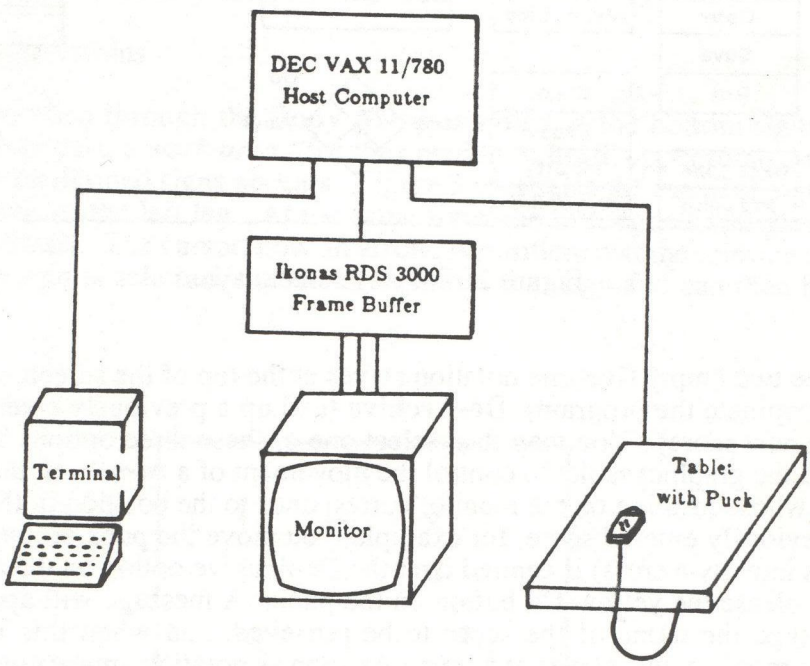

Figure 1. Workstation setup.

A Description of the "ChoreoScribe" Program

"ChoreoScribe" has been designed to produce, edit, and perform limited verification of Benesh dance scores. Since the system is intended for use by notators unfamiliar with the jargon and intricacies of computers, it has been designed to guide the user, step by step, through the workings of the program. To start the program, you use the keyboard to type a command which is seen on the terminal screen and which generates the initial display on the colour monitor. Figure 2 illustrates the general layout of the display screen, although the entire display is never visible at any given time. Only appropriate options are shown.

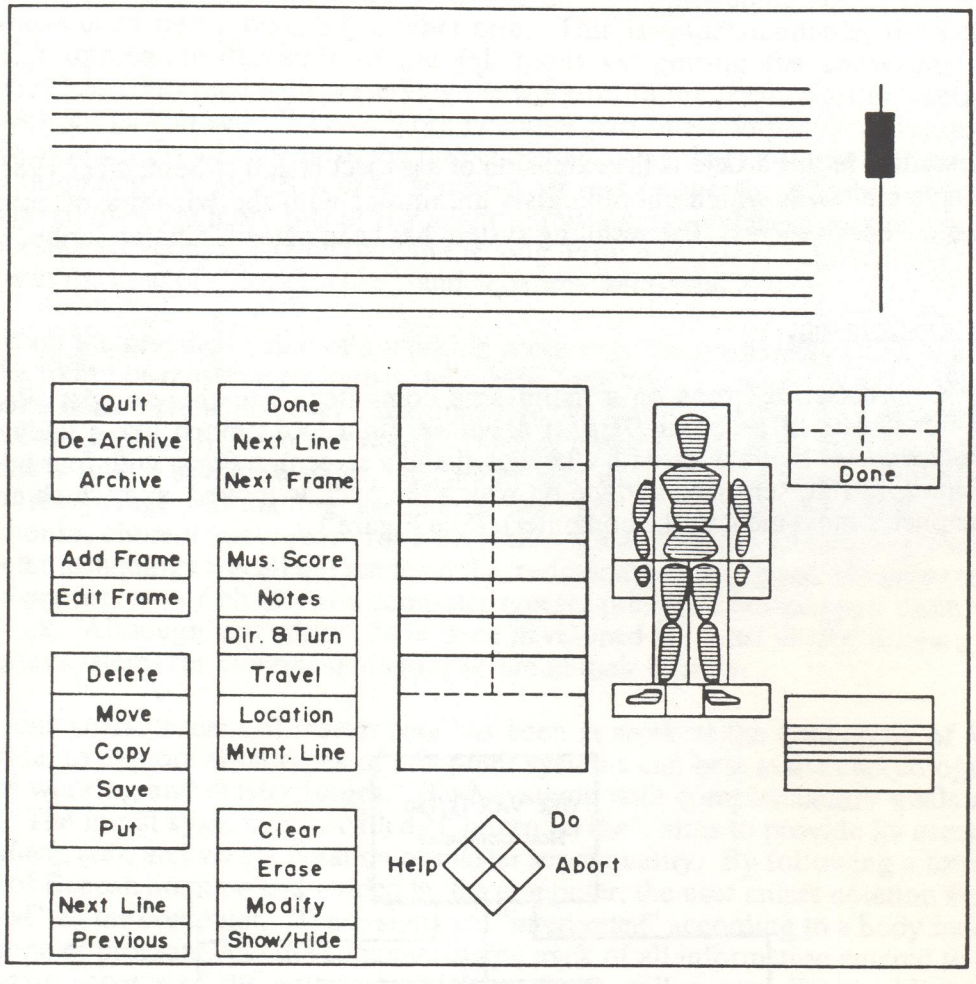

Figure 2. Display screen layout.

Initially, you see two empty five-line notation staves at the top of the screen, and three commands below: **Quit** (terminate the program), **De-Archive** (call up a previously entered score), and **Add Frame** (enter a new score). You may then select one of these three options by manually moving the puck around the graphics tablet to control the movement of a *cursor* on the monitor screen. A cursor is a sign whose location on the monitor corresponds to the position of the puck on the tablet. To call up a previously entered score, for example, you move the puck around the tablet until the cursor (which is initially a cross) is centred over the De-Archive option on the monitor screen. You then press and release the yellow **Do** button on the puck. A message will appear on the terminal asking you to type the name of the score to be retrieved, and when this is done the score is displayed on the monitor, two staves at a time. As soon as notation appears on the two staves, you can perform the editing operations displayed in the lower left portion of the screen. You may **Add**, **Modify** (edit), **Delete**, **Move**, **Copy**, or **Save** notation frames. Frames may be saved, for example, when a theme is to be repeated later in the score. When more than two staves of notation have been entered, two new options, **Next Line** and/or **Previous Line**, are displayed. These

allow the user to scroll forward or backward through the score, one stave at a time. A *vertical scroll bar* is displayed to the right of the staves, to provide visual feedback showing where the two lines displayed are located within the whole score. Each score or version of a score which the notator wishes to save may be archived under a separate name (eg. "Napoli_Jan_82" or "Napoli_July_85"), allowing the original version to remain intact when subsequent revisions are made. If you wish to revert to the previous level of the program before completing the current one, you simply press the green **Abort** button on the puck. If you require assistance at any point, pressing the white **Help** button calls up a series of relevant instructions on the terminal.

To enter a new score you select **Add Frame**, and the screen displays a new set of appropriate *menus*. Each menu lists a number of appropriate options which you may chose to pursue. Although the editor has been designed to allow notation of body positions and movements in any context, specific details have been incorporated to allow its application to ballet. That is, musical scoring (**Mus. Score**), word notes (**Notes**), and staging details (**Dir. & Turn, Travel, Location**) have been included. The enlarged *work-frame* appears in the centre bottom of the screen, allowing the choreologist to achieve finer discrimination in positioning notation signs. A *dotted centre line* as well as top and bottom *leger lines* have been added to provide additional frames of reference enhancing the accurate placement of signs.

When the notation of a work-frame is complete, you may select **Next Frame, Next Line** or **Done** to transfer the contents of the work-frame to the stave. The Next Frame option places the current work-frame at the indicated position in a stave, and the new work-frame on the same stave. The Next Line option places the current work-frame at the indicated position in a stave, and the new work-frame on the next stave. The Done option, after placing the work-frame at the indicated position, takes you to the *stave-menus* listed at the bottom left of the screen, allowing you to manipulate the two staves displayed (scroll, Add, Edit, etc.), or the entire score (Quit, De-Archive and Archive). This gives the choreologist considerable control over the format of each stave: although each Benesh frame is theoretically 16 mm. wide, narrow body positions (eg. 5th position, arms 5th en haut) are notated in less space than wide body positions (eg. arms and legs à la seconde). In practice, more narrow frames than wide frames fit on a stave, but the choreologist has some freedom to choose a more or less dense format. "ChoreoScribe" does this automatically: the computer calculates the greatest horizontal distance of any sign from the centre of the frame (this value is doubled to obtain the *data width* of the frame), sums up the data width of all frames on a stave line, and spaces them evenly. If wide spacing is desired, the notator selects Next Line after a few frames have been added. If many frames per stave are preferred, the user continues to select Next Frame until the desired density has been achieved.

Body positions and movements

Body position is specified through the **Body Menu** diagram at the bottom right of the screen. After you select a body part, a *work area* (for the creation of head, upper torso, and pelvis signs) or a *pop-up menu* of predefined signs appears. Figure 3 illustrates the menu of signs available for specifying the position of the left leg. At the same time, the left leg is highlighted on the Body Menu to provide feedback. The cursor, now an arrow, is positioned at the relevant sign by moving the puck. When the sign is selected, the cursor becomes that sign and can then be dragged into position.

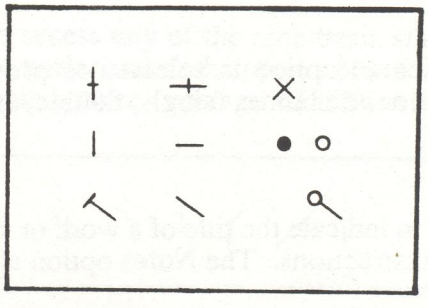

Figure 3. Menu of left leg signs.

Figures 4 and 5 illustrate menus for balletic positions of the feet, accessed by selecting the box between the two feet on the Body Menu. Figure 4 presents the menu of 1st, 3rd, and 5th position signs.

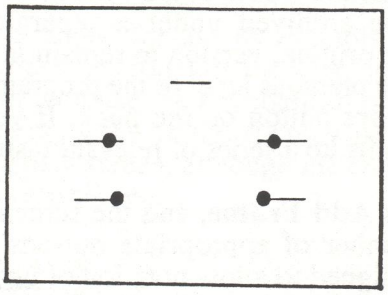

Figure 4. Menu of balletic foot positions: 1st, 3rd, and 5th.

Figure 5 presents the hierarchical menu of "foot closing" signs and corresponding "cou-de-pied" and "coupé" signs that appear upon selection of the sign ⊷ in Figure 4. These menus allow easy access to 100 signs with three or four button pushes.

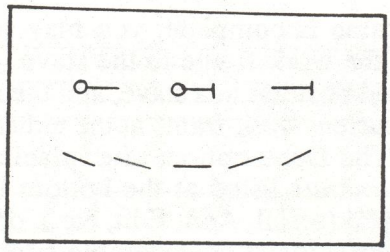

Figure 5. Menu of balletic foot positions: 5th right front--closing, cou-de-pied, and coupé.

A unique feature of "ChoreoScribe" is the **Show/Hide** option. Benesh notation records changes in position: body parts are understood to remain in the last position specified until a new indication is given. The choreologist must mentally keep track of the total body position at all times. When the Show option is selected, the total body position is displayed in the work-frame, providing a helpful double-check. This is not possible in the "MacBenesh" editor (described in the following article) due to the fundamentally different way in which the information is stored in the computer.

Locomotor and non-locomotor transitions are accessed through the Movement Line (**Mvmt. Line**) option. Locomotor transitions, such as steps, jumps, and slides, are indicated by means of lines which span individual frames. When you select such a movement, you are guided through a series of questions pertaining to the type of movement selected and based on the syntax and semantics of Benesh Movement Notation. Non-locomotor transitions which can be shown within a frame are treated differently. Since an infinite number of transitions is possible, these lines are created through the **Free-Hand** option (in which the cursor acts like a pencil), rather than selected from predefined menus. You press and release the Do button when the cursor points to the start of the movement line, sketch the path of the curve, and press and release the Do button when the cursor points to the line's end. Slight deviations in the line are automatically smoothed.[2] You may then modify the shape of the curve by moving control vertices.

Musical scoring

The Musical Scoring (**Mus. Score**) option is selected to provide access to menus for Time Signature, Benesh Rhythm Notation, Bar Lines (single, double, and dotted), and Repeat Signs.

Word notes

Strings of text are often required to indicate the title of a work or section, musical tempo or quality, props, or special choreographic instructions. The **Notes** option allows you to type the needed text and then position it via the puck.

Body orientation is indicated by selecting one of eight basic direction signs via the **Dir. & Turn** option, as illustrated in Figure 6. The cursor is positioned at the desired sign, and the Do button pressed twice. A direction sign is automatically positioned centrally under the work-frame, since this is the position at which it is placed most often. If needed, its position can then be altered via the Modify option. This same menu is also used to access over 600 basic turn signs (1/8, 1/4,...x-7/8) derived from turning clockwise or counterclockwise, starting and ending in any of the basic directions faced. The cursor is positioned on the starting direction sign, the Do button is pressed and released, the turn is manually traced, and the Do button is pressed and released on the final direction sign. If needed, you may then select a numeral indicating the number of multiple turns--doubles (2), triples (3), or unspecified (x)--and drag the resulting sign or composite-sign to the appropriate position under the work-frame.

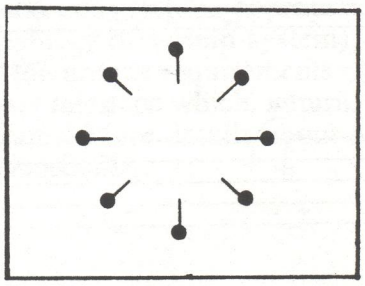

Figure 6. Menu of basic direction signs.

General travel is indicated by selecting one of the nine basic travel signs via the **Travel** option, as illustrated in Figure 7.

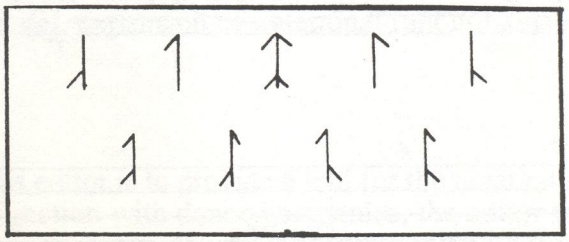

Figure 7. Menu of basic travel signs.

Figure 8 illustrates the forward travel sign selected in its eight basic orientations with respect to the stage. Once the orientation is selected, the sign is dragged to the appropriate position below the work-frame.

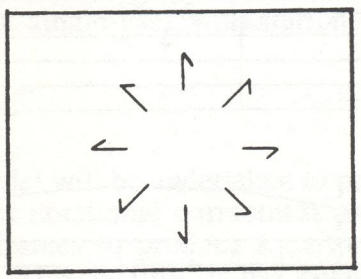

Figure 8. Menu of basic orientation of travel forward.

The **Location** option is selected to access any of the nine basic stage location signs illustrated in Figure 9. Once a sign is selected, a horizontal dash appears below it, ready to be positioned along that sign to indicate the dancer's depth between upstage and downstage.

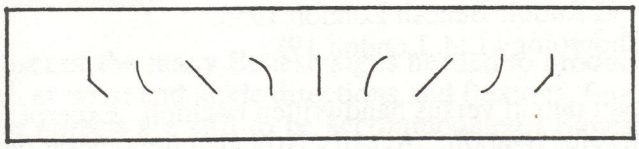

Figure 9. Menu of basic stage location signs.

Typeset Output

Printed copies of scores produced through "ChoreoScribe" are available from a number of devices[3] in differing qualities. A sample, produced on an Autologic APS-Micro5 typesetter, is provided in Figure 10.

Choreography: Beck after Bournonville
Music: Paulli
Notation: R. Ryman

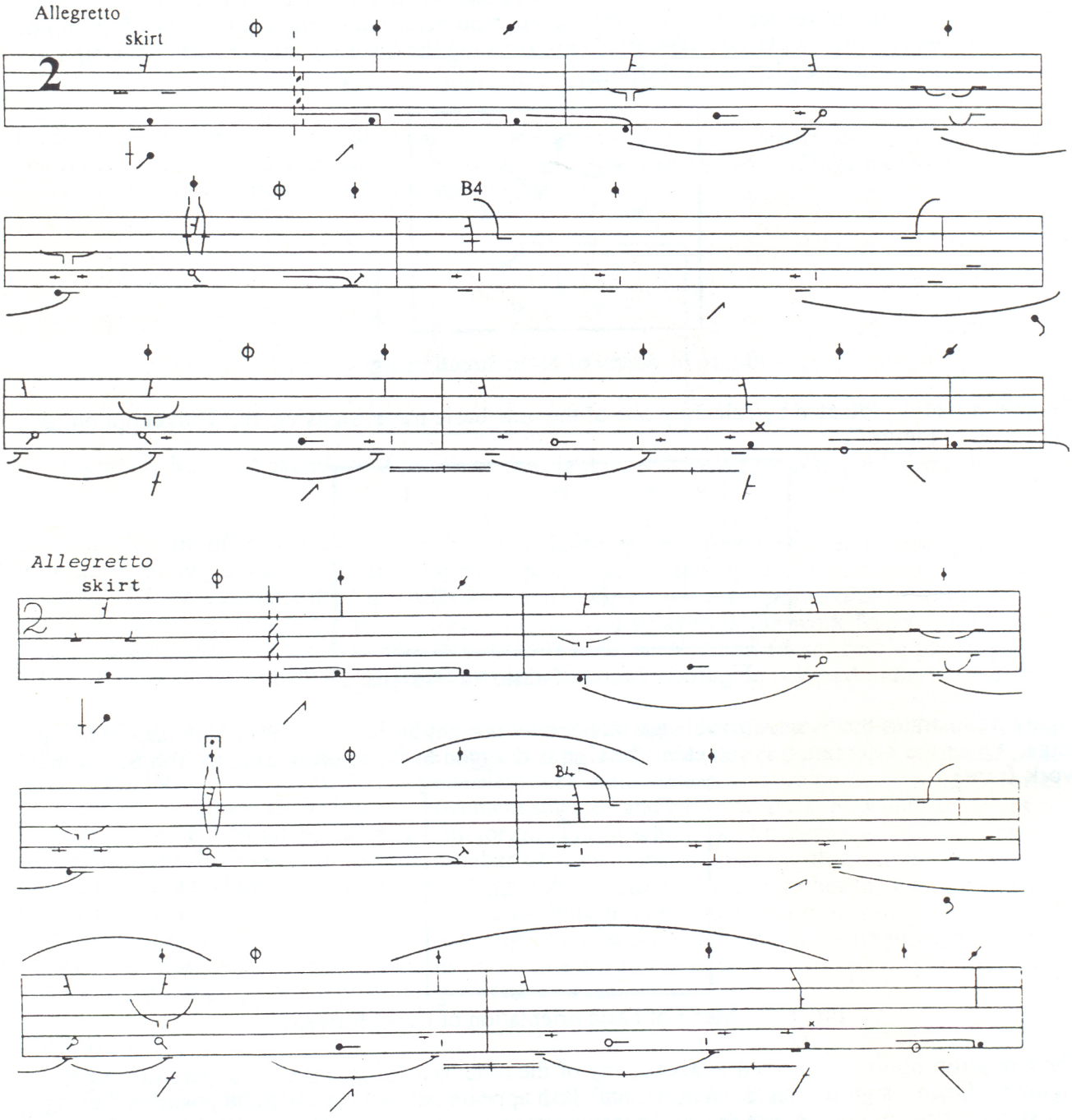

Figure 10. Typeset output versus handwritten notation. Excerpt from the 2den Damesolo, "Napoli," Act III. Note that the typeset output does not accommodate all Benesh signs (eg. legato lines, support signs).

Human Interface Aspects

Human factors play an ever-increasing and critical role in the acceptance and ease-of-use of any system today, and have been of prime concern in the development of "ChoreoScribe." The role of the choreologist has been given great emphasis in the design of its *human interface*. In lay terms, the human interface of a computer system consists of all inputs from the user to the system (inputs produced by motor activity), all outputs from the system to the user (information seen or heard, and understood), and the sequencing of these inputs and outputs. These determine the way in which the user and the system interact and communicate with one another. In Foley's words, "Interaction with a computer involves three types of basic human processes: perception, cognition, and motor activity. The system designer's job is to design interaction techniques which minimise the work required by these processes, both individually and in combination."[4] Recent research has focussed on a number of relevant issues including ergonomic factors (in the design of work environments), the use of appropriate input devices (eg. keyboards versus pucks), factors affecting user productivity (eg. simplicity and consistency, appropriate use of feedback), and speed of learning to use a system (eg. availability of a Help system). The result of the process of task analysis (in our case, determining the unique requirements of a choreologist entering a Benesh score) influence the design of the user interface which, in turn, influences the overall design of the *software* (i.e. the computer program). More details about human interface issues relevant to "ChoreoScribe" are provided in the Appendix .

Future Extensions

The work described here is but a small subset of a system that may eventually find acceptance as a useful tool for the notation of dance using Benesh Movement Notation. The current prototype has already illustrated its usefulness in some error correction of scores as they are entered. Further work will centre on the following areas: implementing the editor in a workstation environment, its extension to a full Body Model, expansion of notational functionality, and enhancement of editor functionality.

Workstation environment

The primary objective of this editor is to provide a tool for the notation of dance works. Since most choreologists work in conjunction with dance companies, the editor must move from a high-cost research environment to an inexpensive stand-alone workstation housed on the company premises. At present, users of "ChoreoScribe" must share computer resources with numerous other users, with the result that production time is often increased. Program development has taken place in the existing time-sharing environment, with little attention to the creation of a new environment ergonomically designed for this particular application. In the future, it is expected that a version of this editor will be available in a single-user workstation environment for the professional choreologist.

Body Model

Extension of the current Body Model will be undertaken to provide *verification* of body postures and movements (i.e. checking for notational correctness and biomechanical possibility), and *animation* (i.e. using notation frames to produce a cartoon-like display of a human figure performing the dance notated). This facility, it has been suggested, could be used in the choreographic process, or in non-dance applications such as cinema (to control the motions of animated movie characters) and robotics (to specify mechanical movements).[5] The reverse process--manipulation of a computer-drawn figure on a screen or a puppet model--could generate a Benesh score.[6]

Expansion of notational functionality

To date a substantial subset of the many Benesh signs needed to produce a ballet solo have been implemented. Items such as wrist and ankle directions and flexions, finger positions, legato lines, and support of inanimate objects are still to be accommodated. Benesh Movement Notation is dynamic in that its syntax is evolving as the notation is applied to new dance styles and other

movement forms. To facilitate an ever-expanding vocabulary, the concept of a *Sketchpad* has emerged: if the user requires a sign not available in the current editor, an implemented sign may be customised or an entirely new one created by manipulating and combining shapes selected from a "library" of signs. Signs created with the Sketchpad, however, are not compatible with the Body Model concept since their form and therefore their meaning are not predefined.

When scores of solo dances have been accommodated, it is planned to add facilities for notating *multiple performers* in a single score. In addition to requiring an expanded vocabulary, this creates interesting spacing problems, since simultaneously performed movement sequences are vertically aligned. The method of determining spacing must ensure that corresponding frames, bars, and score lines for different dancers appear in correct relation to one another.

Enhancement of editor functionality

As the task of expanding notational functionality nears completion, more time will be spent in improving the ease of use. The following enhancements of the human interface are seen as important. The current method of scrolling through the score is a time-consuming task which should be supplemented by a method for directly calling up the frame, bar, or stave of interest, perhaps via a **Find** command. The addition of a mode in which the entire screen is used to display staves would be useful, especially when scrolling through a finished score. Another useful feature would be a **Symmetry** function, since classical ballet exhibits many positions and movements with symmetry about the mid-sagittal plane, eg. second position of the arms and feet. Ballet *ports de bras* are often notated with curved movement lines whose symmetricality is difficult to accurately produce free-hand. In addition to producing more accurate and aesthetically pleasing notation, such a function would simplify the interaction required. The process of notating repeated themes could also be facilitated if more than the currently available single *save-buffer* were available. A buffer is a region of computer memory for temporary storage of data. Numerous buffers would make it possible to save a library of set steps (eg. pas de basque, grand jeté en tournant) and thematic sequences (eg. steps which characterise a particular choreographer's style), and to work on a number of scores simultaneously, copying repeated sequences from one score to another. In this way, a library of often used *macros* (short names which call up whole sets of commands) could be developed to define common movements. It may also be helpful to provide a facility for storing segments of a score via a **Partial Archive** command.

Finally, a valuable direction for this project lies in the teaching and refinement of the Benesh system itself. The design of the user interface for "ChoreoScribe" has required a great deal of specification and analysis of the logical concepts of Benesh notation. The value of this process and the resultant computer system have been recognised as methods which can assist in the presentation of Benesh Movement Notation. As the full Body Model is implemented, its use can logically and systematically be extended to a number of human movement applications including sports, anthropology, ergonomics, and clinical studies.

Acknowledgements

This work has been done under a two-year grant from the Social Sciences and Humanities Research Council of Canada, with further support from the National Science and Engineering Research Council, the University of Waterloo, and the Computer Graphics Laboratory. The authors gratefully acknowledge the assistance of Adrian Grater, Assistant Director of the Institute of Choreology, who has served as a consultant over the past year.

Appendix:

Human Interface Issues

Use of appropriate interaction devices

The editor currently makes use of *select* (place the cursor arrow at the desired sign and press the Do button), *position* (select a sign, a control vertex, or a text string, drag it to the appropriate location, and press the Do button), and *path* operations (position the cursor arrow, press the Do button, trace a curve or circular path, and press the Do button). Limited use is made of the *text* operation (entered via the keyboard). Future enhancements to the editor are expected to include the *orient* operation, since many Benesh signs are semantically differentiated according to their orientation (eg. direction signs), and improved facilities for inputting text (eg. title and background information, props, zoning).

Factors affecting user productivity

"ChoreoScribe" is designed primarily as a *mode-oriented* system in order to accommodate the Body Model feature. That is, the user first selects a particular mode, such as Add Frame, then proceeds systematically through a sequence of operations allowed within that mode. This allows the system to keep track of information as it is specified, and enables the editor to enforce certain constraints relating to the syntax and semantics of Benesh notation. Such constraints include the automatic crossover of an extremity sign when it is not located in its own domain. In this way, the system can relieve the choreologist of some of the burden of notation and help in the verification of what has been notated. For some operations in the editor, a *mode-less* operation has been implemented. For instance, the Modify and Erase options allow the user to alter or delete any of the following: body position signs and movement lines, musical scoring signs, text strings, and staging signs.

For most functions of "ChoreoScribe", the development of the user interface is an *iterative* process. That is, the general concept of the user interface for a function is first implemented and then used by choreologists, in order to isolate aspects of the interface which could be improved. Some improvements involve 'fine-tuning' such as reshaping or reordering a menu, or changing its placement on the screen. Others involve redesigning, reimplementing, and retesting, as was the case for locomotion lines. In each case, consistency and simplicity are fundamental criteria--but criteria which are not always compatible.

A degree of *simplicity* has been achieved by adhering to Benesh notation's *conceptual model*, i.e. the set of ideas which are used to explain the behaviour of the system by the user. By employing concepts already familiar to the user, the system becomes easier to learn and appears simpler. Computerisation, however, may provide the means for improvements in methodologies currently employed by choreologists. This was the case, for instance, in the creation of head, upper torso, and pelvis signs. Rather than manually drawing the sign, the choreologist specifies the degree and type of displacement, and the appropriate sign is automatically positioned in the work-frame. In general, if the notation does not predefine a position within the work-frame for a particular type of sign, the user must drag it into the work-frame and place it appropriately. If, however, the notation does predefine the position of a sign, that sign is placed automatically by the system in order to simplify the operation. *Consistency* has been sought by using operations in the same manner throughout the system, allowing the user to apply knowlege and experience gained about one aspect of the system to other aspects.

Another fundamental principle of "ChoreoScribe" is *error avoidance* to minimise the amount of time spent by the user in error recovery activities. An example is the deletion of an existing sign when an item is redefined (eg. three feet cannot be specified in one frame). Other features include the automatic crossover of an extremity sign which is outside its normal domain, and the automatic placement of signs which have predefined positions, as previously described. As future development is done, particularly the implementation of a full Body Model, further constraints will be applied to enforce the proper syntax and semantics of Benesh Movement Notation.

Minimising the amount of effort for motor movement is one of the objectives of a good user interface. We have attempted to keep puck movement and button pushes to a minimum, and to automate tasks wherever feasible, as in the automatic placement of signs for which the notation predefines a position. The placement of pop-up menus is generally in the vicinity of the cursor, minimising the movement of the user's centre of attention as well as the puck.

Speed of learning to use a system

"ChoreoScribe" has been developed for use by persons with limited computer experience. To assist in the *learning and use* of the system, the user is constantly provided with feedback on the system state and activity. A system of "Help" menus has also been implemented which, when required, give a brief textual description of the current operation. As previously mentioned, since "ChoreoScribe" is geared to the untrained, novice computer user, it is an interactive system with primarily *computer-initiated dialogue*, whereby the user is prompted or guided by menus or questions presented by the system. The use of a *menu-driven system* reduces the load on the short-term memory. Instead of remembering commands to issue to the system (as is required in user-initiated dialogue), the user is guided through the steps of the task by the system, and at the completion of each step feedback is employed to keep the user informed of progress toward task completion. However, *user-initiated dialogue* is appropriate for simple tasks, and has been implemented in "ChoreoScribe" for the Abort and Help functions. These can be selected at any point in the program by pressing the green and white puck buttons respectively.

Static menus, those which are displayed at a fixed location on the screen, allow the user to anticipate the menu and direct the cursor toward it even before it appears. Such menus, however, require a shift in visual attention, and occupy display space which is at a premium. *Pop-up menus*, those which appear near the current cursor location, minimise both hand and eye motion, but have the disadvantage of sometimes overlaying and occluding other screen data. "ChoreoScribe" employs both types of menus, organised hierarchically due to the number of functions available. The Stave-Menu, Frame-Menu, and Body-Menu levels are static menus, while further menus (eg. extremity signs) are pop-up or a hierararchy of pop-up menus (eg. feet position signs). Future extensions of the editor may involve the use of more than the current number of menus; it may therefore become feasible to provide experienced users with some form of more direct access to the desired functions.

The *number of options* at each level in the hierarchy of menus has been limited by the common rule of thumb: "Seven plus or minus two".[7] When more than nine menu items are displayed at a time, they are clumped into logical clusters containing no more than nine items each. Attention has been given to the *shape and ordering of menu items*. All menus composed of textual options are organised in a vertical fashion, clustering among groups as mentioned.

For non-textual pop-up menus, the order and shape are determined by the information contained. For example, stage location signs are listed from left to right indicating stage columns from stage left to stage right.

In any form of conversation, be it between two persons or between man and machine, *feedback* plays an important role. In "ChoreoScribe" feedback is provided to keep the user informed of the system state, the current status of the score, and the current status of the work-frame. The three levels of feedback to user actions have been categorised as *lexical* (pertaining to the words or vocabulary of a language), *syntactic* (pertaining to the formal structures of symbol systems, especially languages), *and semantic* (pertaining to the relationship of signs or symbols and that which they represent, i.e. meaning)[8]. In "ChoreoScribe" lexical feedback is word or symbol output corresponding to the same action or input, and includes the appearance of characters on the terminal screen as they are typed via the keyboard. Syntactic feedback is a response indicating that the system has accepted an input action completed by the user, and includes the change of the cursor arrow to the sign selected. Semantic feedback is information that the requested task has been completed, and is usually provided by displaying the results of the operation just performed. Since feedback should be provided immediately, placebos are provided for operations taking more than a few seconds in order to tell the user that the system is working on the requested task. "ChoreoScribe" uses a Buddha icon adapted from Newshole.[9]

Feedback regarding *system state* is done in two ways. When an option is selected from a static menu, that option is highlighted and all other options disappear (i.e. are switched off). Another method is to alter the shape of the cursor: it is a cross when a command is to be selected, an arrow when an object is to be selected, and a Benesh sign when that sign is to be placed in the work-frame. The *current status of the score* is shown in various ways. For example, the position of the two displayed staves within the entire score is indicated by the vertical Scroll Bar on the right of the screen, and the insertion point of the work-frame being added or edited is indicated by an arrow on the stave display. The *current status of the work-frame* is indicated via the Show/Hide option: when the option is selected it is highlighted, and the work-frame is in the Body Model mode which displays information carried from previous frames as well as newly specified information. Otherwise the work-frame is in notation mode displaying only those signs differing from the previous frame. Different background colours are used to differentiate between menu items not defined (black), items defined in and carried from a previous frame (grey), and items specified in the current work-frame (same colour as the work-frame).

Colour versus resolution

Although the use of *colour* provides helpful feedback in this implementation of the editor, it has been achieved at the expense of high *resolution.* Resolution is a measure of the granularity of the display, and is described as the number of *pixels* (picture elements or phosphor spots) on the monitor screen. Current development has been done on a medium resolution (512 X 512 pixels per inch) frame buffer. This is approximately the same as a domestic television set. Greater resolution produces more clearly distinct shapes and more smoothly curved lines. Currently, only a monochromatic display will provide appropriately high resolution (1024 X 1024) while keeping the projected cost of a workstation within reason. In discussion with choreologists, the general consensus is that increased resolution should take precedence over the use of colour.

Notes

[1] The editor runs on a DEC VAX 11/780 running the Unix operating system. It uses an Ikonas RDS 3000 graphics system attached to a colour display monitor on which most of the output and feedback from the program is shown. A Summagraphics Bit Pad tablet and puck are used as the primary input device. A terminal is used to start the program, for entry of text strings, to display error messages and to display any Help information requested by the user. All code for the editor is written in the C programming language. Programmers are referred to: Singh, Baldev, "A Graphics Editor for Benesh Movement Notation," M. Math. Computer Science, University of Waterloo,1982; and Dransch, Detlef O. K., "ChoreoScribe: A Graphics Editor to Describe Body Position and Movement Using Benesh Movement Notation," M. Math. Computer Science, University of Waterloo, 1985.

[2] After experimenting with various representations, we have chosen to least squares fit a cubic B-spline curve to the initial sketch of a movement line. Editing of the line is accomplished by moving control vertices. End point interpolation is accomplished by using knots at the beginning and end of the curve, for which the knot spacing is otherwise uniform. The number of control vertices may be varied by the user, depending on the complexity of the movement line.

[3] A separate program is under development which reads as input a score produced by the editor, and then reproduces it on a number of devices including a Versatec V-80 raster plotter (200 dots per inch), an Imagen/Canon 10/240 laser printer (240 dots per inch) and an Autologic APS-Micro5 typesetter (723 dots per inch).

[4] Foley, James D., Victor L. Wallace, and Peggy Chan, "The Human Factors of Computer Graphics Interaction Techniques," Computer Graphics and Applications, 4(11) pp. 13-48 IEEE Computer Society, November 1984.

[5] Badler, N. I., J. O'Rourke, and B. Kaufman, "Special Problems in Human Movement Simulation," Computer Graphics (Proceedings Siggraph '80) 14(3) pp. 189-197, July 1980.

Calvert, T. W., J. Chapman, and A. Patla, "The Simulation of Human Movement," Proceedings Graphics Interface '82, pp. 227-234, NCGA, 1982.

Herbison-Evans, D., "A Human Language for Computer Animation," pp. 117-128 in Language Design and Programming Methodology, ed. Jeffrey M. Tobias, September 1979.

Herbison-Evans, D., "Real-Time Animation of Human Figure Drawings with Hidden Lines Omitted," IEEE Computer Graphics and Applications, pp. 27-34, November 1982.

[6] Calvert, T.W., J. Chapman, and A. Patla, "The Integration of Subjective and Objective Data in the Animation of Human Movement," Computer Graphics (Proceedings Siggraph '80) 14(3) pp. 198-203, July 1980. This paper describes a system using goniometers (devices which measure body joint angles) connected to a person in order to record movement data and produce Labanotation.

[7] Miller, G., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity For Processing Information," Psychological Review 63(2), March 1956.

[8] Foley, J. D., "The Design and Implementation of User-Computer Interfaces," Siggraph '82, Tutorial No. 7, ACM, July 1982.

[9] Tilbrook, D., "A Newspaper Layout System," M. Sc. Thesis, Department of Computer Science, University of Toronto, Toronto, 1976.