

TIME MANAGEMENT IN INTERACTIVE SCORE EDITING

*Przemyslaw Prusinkiewicz
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada*

ABSTRACT

Problems related to interactive score editing are surveyed. Four aspects of editing: the editing operations, time control, sound synthesis control and score viewing are distinguished. The time factor, perceived as the distinctive feature differentiating score editors from text editors, is given particular consideration. General discussion is illustrated with examples based on the design of an interactive score editor for microcomputers, called INTERSCORE.

RESUMÉ

Cet article présente une analyse des problèmes liés à l'édition musicale interactive. Quatre aspects de l'édition y sont discutés: les opérations d'édition, le contrôle du temps, le contrôle de la synthèse du son et la représentation visuelle de la musique. On attache une attention toute particulière au facteur temps, perçu comme attribut caractéristique de l'édition musicale l'opposant à l'édition du texte. La présentation générale est illustrée avec des exemples fondés sur l'INTERSCORE — un éditeur destiné particulièrement aux microordinateurs.

1. INTRODUCTION

In computer music, a score can be specified either as a program or as a data file. In the first case the score is a sequence of statements in a sound synthesis language, for example Music V. The process of score specification is similar to computer programming in other languages. In contrast, if the score is perceived as a data file, its preparation is similar to text editing. In this case, a text file is created on a virtual strip of a paper of potentially indefinite length. A user can view a selected portion of the file on the screen. In WYSIWYG (What-You-See-Is-What-You-Get) editors, special consideration is given to make the image on the screen similar as possible to the paper copy of the edited document. Consequently, at any moment the user can completely evaluate all aspects of his work. This feature is an essential element of the man-machine interaction while editing.

Maintaining a comparable level of interaction in musical score editing is by far more complicated than in text

editing, because a score is not a static text or picture, but a sequence of events in time. Hence, a fully interactive "WYHIWYG" (What-You-Hear-Is-What-You-Get) editor must provide real-time acoustic feedback. This can be achieved by controlling one or more synthesizers in real time. Thus, the editor must incorporate process control functions in addition to the various editing operations. This real-time aspect differentiates interactive score editors from text editors.

This paper surveys problems related to interactive score editing. The time factor is given particular consideration. A score editor for microcomputers, called INTERSCORE, is referred to for examples.

2. AN OVERVIEW OF INTERSCORE

INTERSCORE [Prusinkiewicz (1984)] is an interactive score editor for microcomputers, loosely inspired by GROOVE [Mathews & Moore (1970)]. It is written in C, in a modular and structured way. It is therefore relatively easy to introduce modifications for the purpose of testing various aspects of the man-machine interface as well as for studying alternative system structures. INTERSCORE is implemented on the Apple IIe micro-computer (*), with a five-octave alphaSyntauri(**) keyboard and Mountain Computer Music System (+) synthesizer boards. A joystick and two pedals (used to control the progress of time) complete the hardware configuration.

The editor provides the composer with both audio and visual feedback. An example of the screen display when editing is shown in Fig. 1. The central part of the screen is thought of as a window in which the selected portion of the score is visualized using the piano-roll notation [Buxton, Sniderman, Reeves, Patel & Baecker (1979), Krasner (1980)]. A non-continuous line indicates two or more instruments playing in unison. Wide lines near the top and the bottom of the screen show the current position of the time cursor. The notes corresponding to this position are being played by the

(*) Apple IIe is a trademark of Apple Computer Inc.

(**) alphaSyntauri is a trademark of Syntauri Corp.

(+) Music System is a trademark of Mountain Computer Corp.

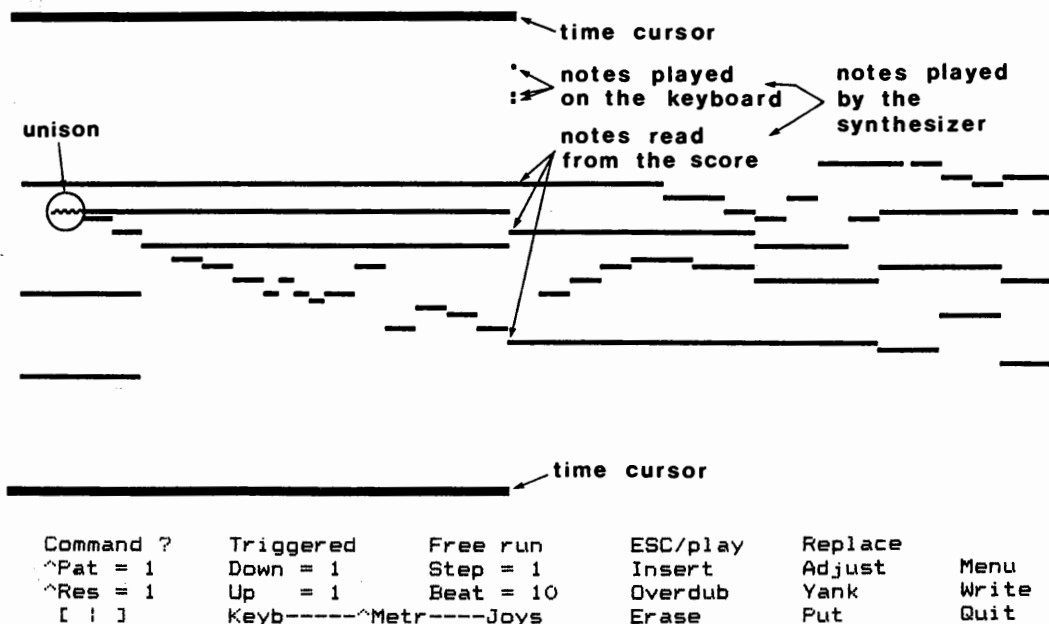


Fig. 1. An example of the INTERSCORE screen display.

synthesizer. Under the score, the text portion of the screen is used to display the main menu of operations and to show the current state of the editor.

The available operations fall into the following categories:

- (i) **Editing:** Operations used to modify the score.
- (ii) **Time control:** Operations used to control the progress of time while editing and consequently to travel through the score.
- (iii) **Sound synthesis control:** Operations used to control the audio feedback while editing.
- (iv) **Viewing:** Operations used to select the portion of the score to be shown on the screen and to control the resolution of presentation.
- (v) **Other:** For example, operations for specifying default settings, or for printing the score on paper.

3. EDITING OPERATIONS

Unlike letters in text, which can be manipulated (inserted, deleted and replaced) independently from each other, events in a musical score may form associations which should be preserved by the editing process. From this viewpoint, the following classification of events can be introduced:

- (i) Single events; for example, corresponding to short percussive sounds.
- (ii) Event n-tuples; for example pairs: the beginning and the end of a note, or five-tuples specifying an

envelope: the beginning of a note, the attack time, the decay time, the sustain time and the release time.

- (iii) Event sequences: representing successive samples of a continuously changing parameter.

In INTERSCORE all events are assumed to form pairs: the beginning and the end of a note. Consequently, the editing operations are grouped into the following categories:

- (i) Operations affecting one event. Examples are: the insertion of the beginning of a note, the insertion of the end of a note, the modification of the time when a note starts or ends, the modification of articulation (corresponding to how hard a key is pressed or how fast it is released), and the labeling of a particular place in the score (for instance, to specify the scope of the subsequent yank-and-put operation). Note that these operations may have a limited domain of applicability. For instance, it makes no sense to move the beginning of a note past its end.
- (ii) Operations affecting one note. Examples are: the replacement of the instrument assigned to a note, the deletion of a note, and the change of the pitch of a note.
- (iii) Operations affecting a portion of the score. Examples are: the deletion of a portion of the score, and yank and put operations. The specification of the scope of these operations involves two or more positions of the time cursor.

Operations belonging to class (i) or (ii) affect only the score point currently indicated by the time cursor, or a small time interval. Arguments of these operations can be specified using only one value of time. The effect of an operation can be instantly evaluated. In contrast, operations of class (iii) affect longer portions of the score, and a longer time is needed to evaluate their effect. This distinction is important in a microcomputer-based environment. If in the main memory there is not enough space for the whole editor, class (iii) operations can be implemented as overlays brought into memory only when needed. The increased time necessary to perform the desired operation will not significantly degrade the man-machine interface.

Treating a score as a totally ordered sequence of events is very restrictive. A composer must be able to freely manipulate concurrent melody lines [Balaban (1981)]. In INTERSCORE this is achieved by providing two basic editing modes, called insert and overdub. In the insert mode the score is virtually spliced at the point determined by the current position of the time cursor, and new notes are inserted between the spliced parts. As a result, the overall duration of the composition is affected. Thus, the insert mode can be thought of as an equivalent of tape editing by splicing [Howe (1975)]. Similarly, the overdub mode is analogous to the magnetic tape procedure of building up a composition one track at a time on a multi-track tape recorder. The added notes are to be played concurrently with the notes previously specified. Notes removed in the overdub mode disappear from the score without affecting the overall timing of the composition.

4. TIME MANAGEMENT

Flexible control of the progress of time is a necessary component of the interactive score editing. The following notions [cf. Joffe (1984)] are useful when describing the time-related effects.

- (i) The composition time t_c is the "real" time passing while entering data.
- (ii) The performance time t_p is the "real" time passing while listening to the music.
- (iii) The score time t_s is a numerical parameter actually entered to the score.

The score time can reflect either the actual duration of each note during the performance, or the rhythmical value of the notes. In the first case, complete information on time relationships within a composition, including changes of tempo, *accelerandi*, *ritardandi* etc., is included in the score. During a performance, the score time is linearly mapped into the performance time. This is equivalent to playing the score at a constant speed. In the second case, the score contains information on the rhythmical value of each note or rest. Nuances of tempo result from an appropriate mapping of the

score time into the performance time. Consequently, this mapping is no longer linear, and it must be separately specified. This separation has several advantages:

- (i) It is much more natural to interpret changes of tempo as changes in the speed of time progress through the score rather than as changes of note durations. For instance, *ritardando* means "play slower", not "make notes longer".
- (ii) Only rhythmical values of notes have adequate graphical representation in the common musical notation (CMN).
- (iii) The flexible mapping of the score time into the performance time can be thought of as a component of the interpretation rather than as a component of the score itself. Consequently, the same score can be used by different performers to create individual interpretations.
- (iv) If the mapping of the score time into the performance time can be controlled in real time, a predefined score can be interpreted during a live performance.

Unfortunately, a consistent separation of the interpretation from the score presents serious problems. For example, it seems very difficult (if it is at all possible) for the editor to determine rhythmical values of notes played *rubato* while entering data.

For the sake of simplicity, in INTERSCORE the interpretation is not distinguished from the score, and the linear mapping of the score time into the performance time is assumed. Still, there is a high level of flexibility in mapping the composition time into the score time. This mapping ranges from identification of these times to their complete disassociation.

In the simplest case, the relationship between composition time t_c and score time t_s is in essence linear: $t_s = kt_p$. The computer simulates a multi-track tape recorder. A score is developed by "recording" successive instruments with "playback." Techniques for score modification are analogous to the punch-in/punch-out facility found in tape-recorders.

Due to the nature of digital computers, the score time t_s is incremented in discrete intervals of composition time Δt_c . Parameter Δt_c controls the time resolution of the score. Thus, for small values of Δt_c (milliseconds) the quantization of time is negligible, and all nuances of interpretation (while entering data) are reflected in the score. Large values of time intervals (fractions of a second) introduce a significant nonlinearity into the mapping of the composition time into the score time. This can be used to eliminate some imperfections which may occur while entering data from the keyboard. However, the composer has to enter data in synchronization with the sampling process. Otherwise, the low time resolution may degrade rather than improve the score.

	no parameters controlled
autonomous	time direction controlled with pedals
	time direction and playspeed controlled with a joystick
TIME MODES	
	triggered from the organ-like keyboard
triggered	triggered from the pedals
	triggered from the alphanumeric keyboard

Fig. 2. INTERSCORE time modes.

By controlling coefficient k in the equation $t_s = kt_c$, the composer can stop the progress of score time t_s ($k = 0$) or reverse its direction ($k < 0$). This allows for the instantaneous correction of errors while entering data. Dynamic manipulation of coefficient k is also convenient when scrolling a score in order to access a particular fragment. Big values of k allow for fast scrolling past irrelevant portions of the score, while small values allow for accurate positioning of the time cursor. In INTERSCORE parameter k is dynamically controlled using pedals and a joystick. They are used to stop the progress of time or reverse its direction. The joystick allows also for continuous changes of parameter k . This is particularly useful when scrolling the score.

In the modes described so far, time progresses autonomously. The external devices only control the actual value of parameter k . These autonomous time modes can be contrasted to the triggered ones. In a triggered mode, each change of the performance time is directly caused by an external signal. One possible source of this signal is the organ-like keyboard itself. In this case time progresses when a key is being pressed or released. The time intervals associated with the pressed or released keys, $\Delta t_s(\text{down})$ and $\Delta t_s(\text{up})$, need not to be the same. If $\Delta t_s(\text{down}) \gg \Delta t_s(\text{up})$, the entered notes will be performed *legato*; if $\Delta t_s(\text{down}) \ll \Delta t_s(\text{up})$ — they will be performed *staccato*.

Pedals are another useful source of triggering signals. While the organ-like keyboard is continuously active, its state affects the score only when pressing or releasing a pedal. Thus, the composer can try, for instance, a few possible chords, before entering the final one into the score. The duration of this chord will be determined by the number of times the pedal is pressed while the keys are down. Instead of pressing the pedal repeatedly, the composer can also specify the duration of each note or pause by entering appropriate values of parameters $\Delta t_s(\text{down})$ and $\Delta t_s(\text{up})$ from the alphanumeric keyboard. A variant of this approach makes use of symbols 1, 2, 2, ..., to denote the duration of the whole note, the half note, the dotted half etc. in a predefined tempo.

A summary of time modes available in INTERSCORE is given in Fig. 2.

5. SOUND SYNTHESIS CONTROL

An audio feedback is one of the essential features of interactive score editing. This feedback can be provided by synthesizers or by real-time sound synthesis programs. Ideally, a score editor should incorporate a complete sound editor/mixer. This would allow for changing instrument definitions within the context of a particular composition, and for adjusting their relative volume levels. This goal, however, may be difficult to achieve for the following reasons:

- (i) In order to change parameters interactively, while playing a score, the sound editor has to run concurrently with the score editor. This requires a main memory space large enough to accommodate both editors. Moreover, the CPU must be fast enough to run these editors concurrently in real time.
- (ii) Parameters to be controlled vary among the different synthesizers and sound synthesis techniques presently in use. Consequently, the development of a device-independent sound editor is a non-trivial task.
- (iii) Concurrent manipulation of the score and the sound can be confusing. Therefore, human factors must be considered with particular care while designing the man-machine interface.

The present version of INTERSCORE allows for the selection of waveforms and the control of instrument volumes from the editor. Sound editing capabilities are implemented as an overlay to the score editor. As a result, instantaneous switching between the score editor and the sound editor is not possible.

6. SCORE VIEWING

The objective of score viewing is to complement the acoustic feedback with a visual representation of the score. The relative importance of the visual *versus*

acoustic feedback depends on the particular editing mode and time mode in use. Generally, in time modes closely associating the score time to the composition time ("recording on the pseudo-tape"), the acoustic feedback is predominant. In contrast, if the score time is dissociated from the composition time (as it may be in the case of triggered time modes), the acoustic feedback is heavily distorted in the time domain, and the visual representation becomes more useful. This is also true for the case of editing operations affecting large portions of the score; for example, yank and put.

Visual feedback involves the following elements:

- (i) Selection of the graphical representation of the score. A number of notations, usually derived from either the common musical notation or the piano-roll notation, were proposed and studied in the past [Buxton *et al.* (1979)].
- (ii) Selection of the portion of the score to be presented on the screen. This involves positioning of the virtual window with respect to the score, and changing the window size.
- (iii) Control of the time resolution of the presentation, i.e. the smallest score time interval represented on the screen.

The most important question is related to the choice of musical notation. The common musical notation has the following shortcomings:

- (i) It represents only rhythmical values of notes, and neglects nuances of interpretation.
- (ii) There is no straightforward way of representing the score with various time resolutions. CMN is not adequate, when the composer needs a visual representation of long portions of the score.

The piano-roll notation does reflect nuances of interpretation, and — conceptually — allows for the presentation of the score in any time resolution. However, it also has deficiencies:

- (i) The piano-roll notation is less legible than the CMN.
- (ii) For small time resolutions (long score time intervals per pixel) the score is undersampled, and its graphical representation can be misleading (short notes missing, notes of equal duration represented with bars of nonequal length, etc.)

In INTERSCORE the piano-roll notation is used. Three special keys ([, | ,]) make it possible to position the window in such a way that the time cursor appears near the left edge, in the middle, or near the right edge of the screen. The composer specifies resolution of the presentation as the number of score time units per pixel. Wide-scope low-resolution views help analyzing the general structure of the composition and are particularly useful when scrolling the score. High resolution views are of value when making fine modifications. A special command: freeze, is used to switch off the visual feedback. This feature is useful when "recording" par-

ticularly fast portions of the score, because screen updating can be then too time-consuming to be performed in real time. The screen is also automatically frozen if the editor cannot update it fast enough or if the time cursor leaves the window.

A number of questions related to visual feedback in interactive score editing remain open for further research. The essential problem is the choice of musical notation. A notation used in interactive score editing is a means of communication between a human being and a machine. This objective is significantly different from the traditional functions of musical notations, the storage of a score and its communication between humans. Consequently, new notations intended specifically for interactive score editing might improve the man-machine interface.

7. CONCLUSIONS

This paper describes problems of interactive score editing. The time factor, perceived as the distinctive feature differentiating score editors from text editors, is given particular consideration. General remarks are illustrated with examples based on the design of a score editor for microcomputers, called INTERSCORE. While the ultimate goal of INTERSCORE is to provide a user-friendly environment for the creation of computer music, successive versions of the editor have been used mainly as an environment for studying the time component of the man-machine interaction.

ACKNOWLEDGMENT

This research was sponsored by grants from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

- Balaban, M. (1981). The set of tonal-music-strings — a structurally unambiguous representation for tonal music pieces. *Weizman Institute of Science Rep.* CS81-28.
- Buxton, W., Sniderman, R., Reeves, W., Patel, S. & Baecker, R. (1979). The evolution of the SSSP score editing tools. *Computer Music Journal* 3 no. 4, pp. 14-24.
- Foley, J. D. & van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*. Reading, MA: Addison-Wesley.
- Howe, H. S. (1975). *Electronic Music Synthesis*. New York, NY: Norton.
- Joffe, D. A. (1984). Ensemble possibilities and problems in computer music. *Abstracts, International Computer Music Conference 1984*, p. 186, Paris, France.
- Krasner, G. (1980). Machines tongues VIII: The design of a Smalltalk music system. *Computer Music Journal* 4, no. 4, pp. 4-14.

Mathews M. V. & Moore, F. R. (1970). GROOVE — A program to compose, store, and edit functions of time. *Communications of the ACM* 13 no. 12, pp. 715-721.

Meyrowitz, N. & van Dam, A. (1982). Interactive editing systems: Part I and II. *ACM Computing Surveys* 14 no. 3, pp. 321-415.

Music System Operating Manual (1981). Scotts Valey, CA: Mountain Computer.

Prusinkiewicz, P. (1984). INTERSCORE — An interactive score editor for microcomputers. *Proceedings, 4th Symposium on Small Computers in the Arts*, pp. 58-64, Philadelphia, PA.