INTERSCORE - AN INTERACTIVE SCORE EDITOR FOR MICROCOMPUTERS

Przemyslaw Prusinkiewicz

Technical Report CS-84-12

COMPUTER SCIENCE DEPARTMENT

University of Regina

INTERSCORE - AN INTERACTIVE SCORE EDITOR FOR MICROCOMPUTERS

Przemyslaw Prusinkiewicz

Technical Report CS-84-12

Department of Computer Science
University of Regina
Regina, Saskatchewan CANADA S4S 0A2

July 1984

INTERSCORE - AN INTERACTIVE SCORE EDITOR FOR MICROCOMPUTERS

Przemyslaw Prusinkiewicz

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2

ABSTRACT

Since 1970, the GROOVE system has been the classic example of an interactive editor of time functions. It has inspired the design of an interactive score editor for microcomputers, named INTERSCORE. A composer is provided with a wide range of easy to use editing operations, and with both audio and visual feedback (in piano-roll notation). An organ-like keyboard is extensively used as an input device. Time relationships are given particular attention.

INTRODUCTION

Buxton et al. [2] divide the composer's tasks in the context of computer music as follows:

1. Definition of the palette of timbres to be available: This is analogous to choosing the instruments which are to comprise the composer's orchestra. The main expansion on the analogy is that the composer also has the option to "invent" his own instruments.

2. Score definition: Definition of the pitch-time structure of a composition. In conventional music, this task would be roughly analogous to composing a piano version of a score.

3. The "orchestration" of the score: Attachment of instruments to a score.

4. The performance of the material being developed.

The conceptual framework induced by the above classification is observed in this paper. Thus, instruments are defined separately from the score, and the possibility of setting up timbres individually designed for each note is limited. In return, the removal of the instrument definition from the score makes it potentially easier to construct long, complex pitch-time structures.

At the present time, music software available on microcomputers makes it possible to define scores using either one of the following approaches:

1. A score is defined as a sequence of statements which explicitly specify the pitch and duration of each note. Definition of a score is conceptually similar to text editing. It is also possible to edit a score using a symbolic musical notation. The score is then specified, for example, by picking notes of the appropriate duration from the menu and positioning them on the staff [12].

2. A score is thought of as a record on a "pseudo-tape," simulated in the memory of a computer. The sequence of events is defined in real time, by playing an organ-like keyboard used as the input device. The user's model of the system is that of a multi-track tape recorder. Thus, a score is developed by "recording" successive instruments with "playback" on separate "tracks." An unsatisfying track can be "erased." Smaller corrections can be made using a "punch-in/punch-out" facility [8].

The essential difference between these approaches lies in their rapport to time. In the first case the composition time is dissociated from the performance time. Time characteristics of a note are defined as numerical parameters. Consequently, the order in which the notes are specified may be different from the order in which they will be played during a performance. The score can be easily modified by inserting, deleting, or changing appropriate statements. However, since it has to be substantially processed by the computer before the performance, effects of the modifications cannot be instantly evaluated.

In the second case the situation is diametrically opposite. The performance time is a linear function of the composition time. A composer has immediate audio feedback: He hears what he plays. Modifications of the score are made in real time, by rerecording unsatisfactory parts. As a result, the whole composition process occurs under real-time pressure. A mastery of the keyboard is necessary. Small adjustments are difficult to make.

An interactive score editor should provide a composer with a wide range of easy to use editing operations combined with good audio and visual

feedback. This objective was first met in the GROOVE system [10]. The nature of interaction implemented in GROOVE was described as follows:

> One of the most important features of GROOVE is the flexible control of "program time" which may be used both to edit and to alter the generation of the output functions.... We may slow down the progress of program time by reducing the frequency of the interrupt oscillator. Or we may stop the progress of time altogether by throwing a switch which essentially tells the computer: "Don't progress time normally at all, but, instead, use the value of a knob to give the current position of time within one disk buffer...." The user may essentially "redraw" any portion of any disk function using any input device he likes, such as the (X,Y) axes of the 3-dimensional wand or a knob value. While he is doing this, not only can he see what he is doing on the oscilloscope display, but he can also observe its effect on the controlled process. So it is quite possible to stop in the middle of a run and "tune up the chord...." Given the appropriate commands, the system will allow any functions of time to be altered in any conceivable manner.

The idea of the GROOVE system has been widely recognized as a model example of interactive score editing [6]. However, the system itself was not portable and is not available since the unique hardware was dismantled.

INTERSCORE is an interactive score editor for microcomputers, inspired by GROOVE. It is written in C, in a modular and portable way. At present INTERSCORE is implemented on the Apple //e (*), with a 5-octave alphaSyntauri (**) keyboard and Mountain Computer Music System (+) synthesizer boards [12]. Two pedals and a joystick complete the hardware configuration. INTERSCORE makes use of the Syntauri programs Quickwave and Wave to define instruments. INTERSCORE files are compatible with the multitrack recording system METATRAK (**) [8] and the Composer's Assistant (**) [13] can be used to transcribe the score using conventional notation.

This paper presents INTERSCORE from the user's (composer's) perspective. Time problems related to interactive score editing are given particular attention.

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

(**) alphaSyntauri, Metatrak, and Composer's Assistant are trademarks of Syntauri Corp.

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

GENERAL DESCRIPTION OF THE SYSTEM

From the user's perspective, the editor is embedded in a menu-driven control program. The functions of this program fall into three categories:

1. Selection of the instruments (timbres) to be used for audio feedback. These instruments can be, but do not have to be the same as the instruments used for the final performance.

2. File manipulation. This category includes concatenation and merging. Concatenation of files results in longer scores. Merging increases the number of instruments playing concurrently [1].

3. Transfer of control to utilities, such as programs to define new instruments, the screen dump program, etc.

An example of the screen while editing is shown in Fig. 1. The central area of the screen is thought of as a window, in which the selected portion of the screen is visualized as a plot in pitch-time coordinates (scroll-bar notation [3, 9]). A non-continuous line indicates two or more instruments playing in unison. For comparison, Fig. 2 shows the same piece of music using conventional notation.

Wide lines at the top and the bottom of the score (Fig. 1) show the current position of the time cursor. The notes corresponding to this position are being played by the synthesizer. Under the score, the text portion of the screen is used to display the menu of editing operations and to show the current state of the editor. The selected mode is displayed in inverse video. Special characters warn about particular situations, such as the end of the score being reached. The space in the upper right corner of the text portion of the screen is used to display additional information, for example parameters of editing operations, error messages, etc.

SCORE VIEWING

Even relatively short scores cannot be represented in their entirety on the limited surface of a screen. Thus, an interactive score editor must provide viewing operations which make it possible to select the portion of the score to be seen. In INTERSCORE this selection is thought of in terms of moving a virtual window over the scroll-bar representation of the score. Three special keys ([, |,]) are used 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 window. The composer can also specify the resolution of the presentation (number of time units per pixel). Wide scope - low resolution views help in analyzing the general structure of the composition and are particularly useful when browsing through the score. High resolution views are of great value when making fine modifications.

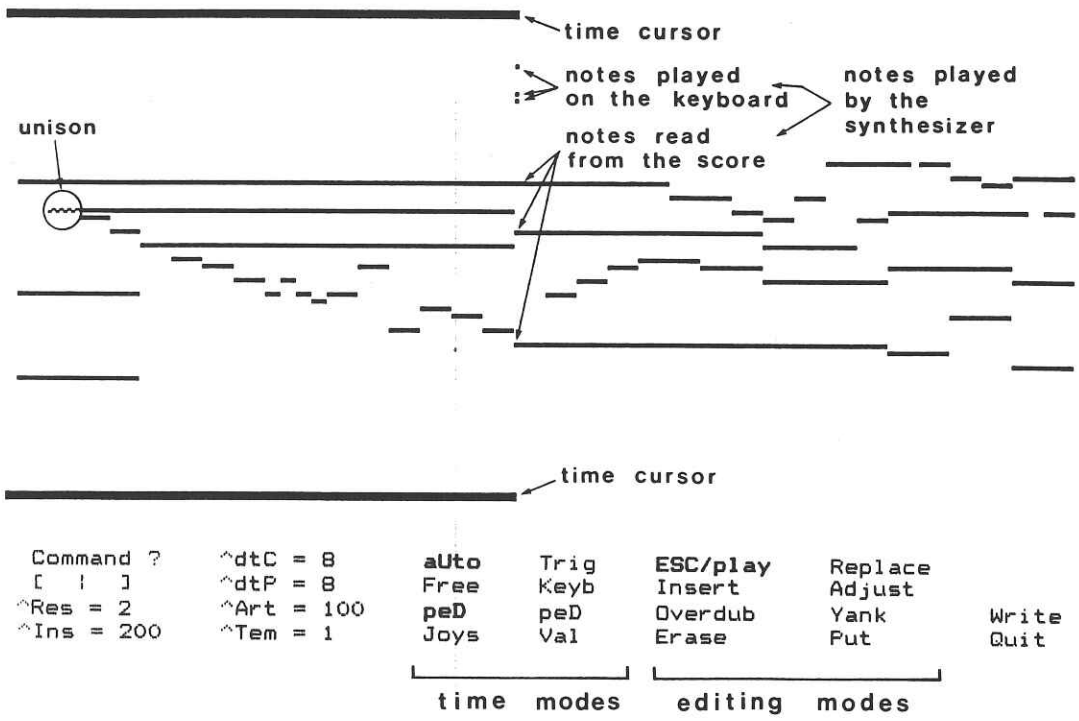

Fig. 1. A dump of INTERSCORE screen.

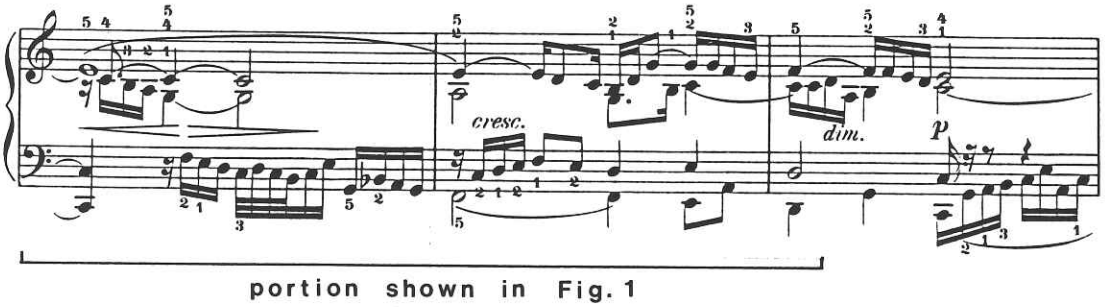

Fig. 2. The score from Fig. 1 in conventional notation.
An excerpt from J. S. Bach, Das Wohltemperirte Klavier, Part II, Prelude I.
Transcription by C. Czerny and A. Ruthardt, Leipzig: C. F. Paters.

EDITING MODES

The two basic editing modes are called insert and overdub (Fig. 3). 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. Extraction of a portion of the score is implemented as the insertion of a segment of negative duration. In both cases the overall duration of the composition is affected. Thus, the insert mode can be thought of as an equivalent of tape editing by splicing [7]. Likewise, the overdub mode is analogous to the magnetic tape procedure of building up a composition one track at a time on a multitrack tape recorder. The added notes are to be played concurrently with the notes previously specified.

The operation complementary to overdub is called erase. The composer can either erase all notes played by a given instrument within some time limits or can select the notes to be erased by

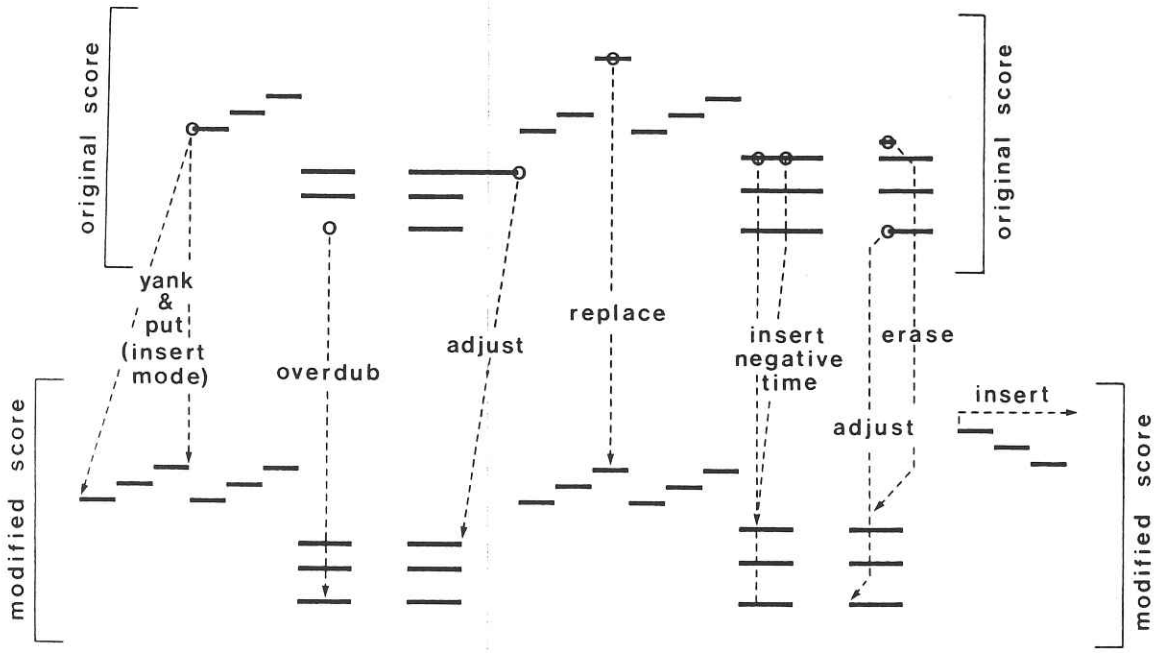

Fig. 3. INTERSCORE editing modes.

pointing to them with the time cursor (x coordinate) and the keyboard (y coordinate). By playing a chord, several notes can be erased simultaneously.

The replace mode is used to change the pitch or the instrument assigned to the selected note.

The adjust mode makes it possible to move the beginning or the end of a note in time.

The remaining two modes: Yank and put are patterned on similar operations found in text editors [11]. Yank fetches a portion of the score, played by the specified instrument, delimited by two positions of the time cursor. Put, in essence, places the yanked portion at another point in the score. This can be done in several ways. First, either the insert or the overdub mode must be selected together with put in order to determine, how the added notes shall be matched with the existing portion of the score. Second, the yanked fragment can be repetitively put, several times in sequence, transposed by an interval specified with the keyboard, and possibly with the instrument changed.

All editing modes are illustrated in Fig. 3.

TIME MANAGEMENT

Time is the essential component of both the process of the composition of a piece of music and its performance. Therefore, the relationship between the composition time and the performance time should be easily manageable by the composer. For this purpose several time modes have been introduced into INTERSCORE.

The simplest relationship between the composition time and the performance time is illustrated in Fig. 4a. During the composition time the keyboard is sampled and the score is updated in equal intervals denoted by Dtc. During the performance the score is read and data to the synthesizer are passed in intervals Dtp. The ratio Dtp/Dtc controls the "playspeed:" If it is different from one, the performance will be faster or slower with respect to the composition. Absolute values of intervals Dtc and Dtp control the "time resolution" of the composition. For small values of these intervals (milliseconds) the discretization of time is negligible and the sequence of events entered into the score is repeated with all nuances during the performance (Fig. 4b). Large values of time intervals (fractions of a second) let smaller time differences disappear. This can be used to eliminate some imprecisions which may occur when entering data from the keyboard (Fig. 4c). The

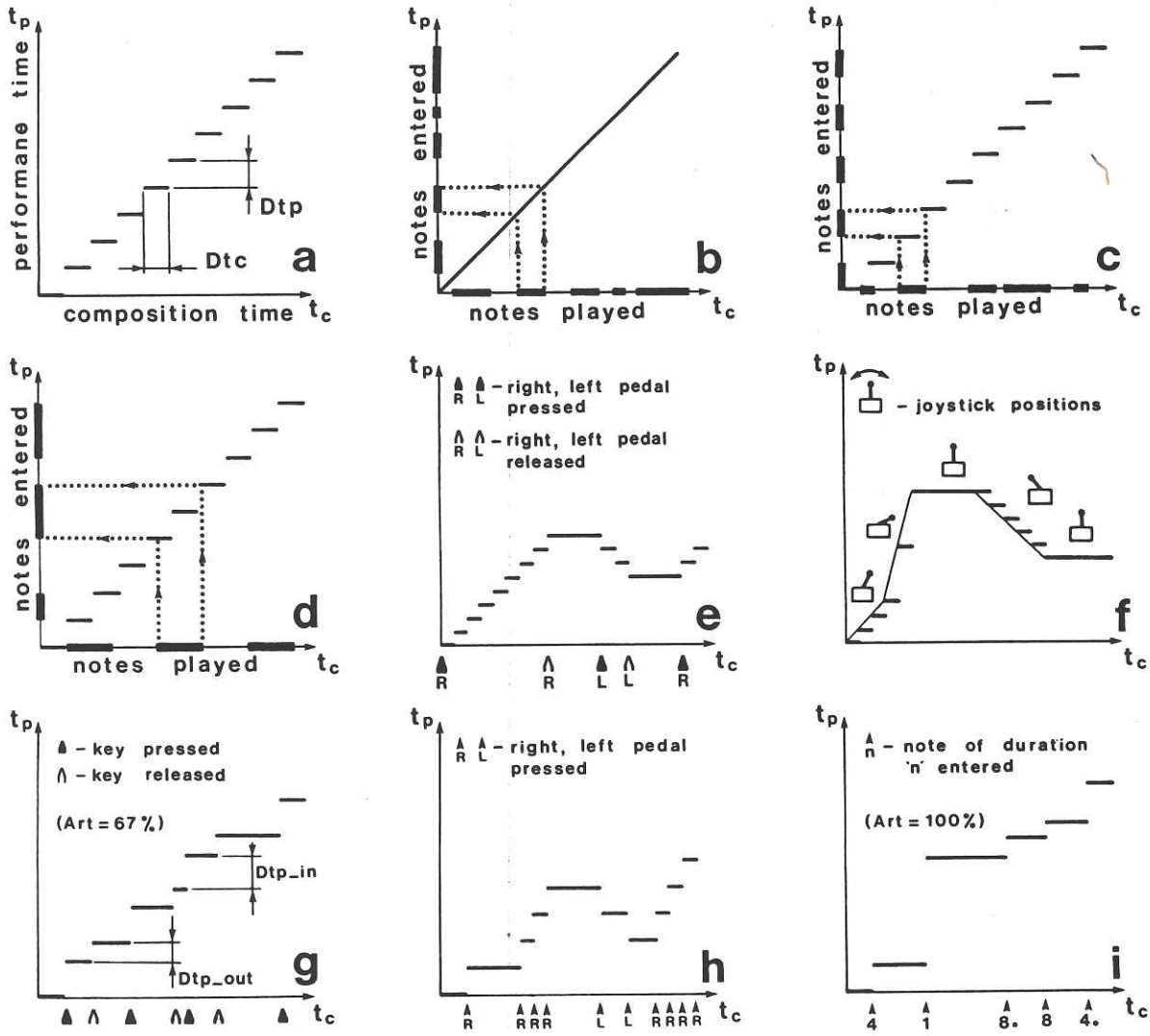

Fig. 4. Composition time - performance time relationship in INTERSCORE. Explanation in the text.

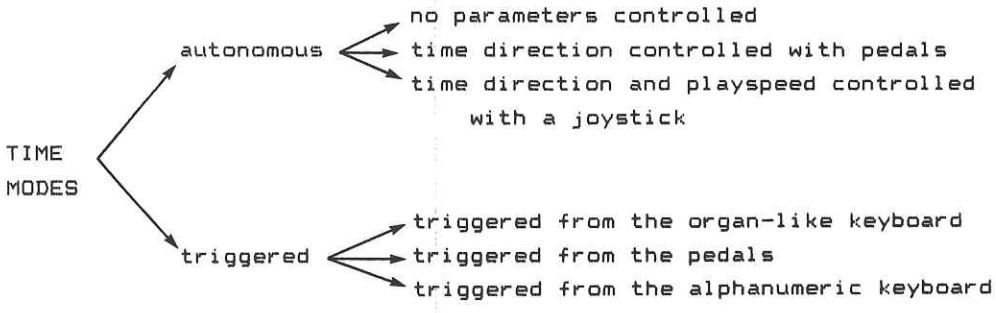

Fig. 5. INTERSCORE time modes.

composer has to enter data in synchronization with a metronome generated by the editor. Otherwise low time resolution may deteriorate rather than improve the score (Fig. 4d).

Abstracting from their finite resolution, the above modes define linear mappings of the composition time into the performance time. A modification introduces a piecewise linear mapping. At any moment the composer can stop the progress of time or reverse its direction by using one of two special pedals (Fig. 4e). While the composition process is still very similar to real-time recording, errors can be instantly deleted before the recording resumes.

When scrolling a score to access a particular fragment, it is convenient to dynamically control the playspeed in addition to the time direction. The device used for this purpose in INTERSCORE is a joystick. See Fig 4f. for an example.

In the modes described so far, time has progressed autonomously. The external devices have only controlled the actual value of parameter Dtp. These autonomous time modes can be contrasted to the triggered modes. 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, Dtp_in and Dtp_out, need not be the same (Fig. 4g). If

Dtp_in >> Dtp_out,

the entered notes will be performed legato; if

Dtp_in << Dtp_out

they will be performed staccato. Instead of controlling parameters Dtp_in and Dtp_out directly, it seems more convenient to specify the overall duration of a note:

Dtp = Dtp_in + Dtp_out

and its "articulation" (or duty cycle):

Art = Dtp_in / Dtp.

Another source of triggering signals are pedals used to increment or decrement performance time by a predefined value (Fig. 4h). While the organ-like keyboard is active all the time, its state affects the score only when a pedal is pressed. Thus, the composer can try, for example, 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 the parameter Dtp 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 and with a predefined articulation (Fig. 4i). This mode is particularly convenient when entering a score given in conventional music notation.

A summary of time modes provided by INTERSCORE is shown in Fig. 5.

CONCLUDING REMARKS

Design and experimentation with INTERSCORE has yielded some observations:

1. Scroll-bar notation is a convenient means for visual communication between the composer and the computer. The difficulty in perceiving the exact pitch of each note (with no staff) is irrelevant because of the audio feedback. Moreover, it is easy and intuitive to find the pitch of a note by matching it with the displaceable bars corresponding to the keys pressed on the keyboard. Due to the audio feedback, precise positioning of the time cursor (up to a single pixel) is trivial. The visual and the audio feedback are complementary.

2. When a long score of a repetitive structure is edited as a single file, the composer may confuse similar parts of the score. It is therefore preferable to build long scores using separate files, which are concatenated at the end of the editing session.

3. Although the description of INTERSCORE was given in terms of physical devices (joystick, pedals etc.,), the software was written in a device independent way, patterned on the methodology developed in computer graphics [5]. Thus, only the appropriate device drivers have to be rewritten when replacing the joystick with a different valuator, or pedals - by any other buttons. Apparently, the organ-like keyboard does not fit into the existing classification and introduces a new class of logical input devices. The problem is, however, that the main purpose of the device independent design (i.e. the possibility of simulating various logical devices using the available physical devices) is questionable in highly interactive musical applications. Substitutions of physical devices (for instance, hand manipulated buttons for pedals, let alone a replacement for the organ-like keyboard) tend to deteriorate the man-machine interface to the point of uselessness, even if they are perfectly feasible from the viewpoint of software design.

INTERSCORE has not yet been extensively tested by musicians, therefore an objective evaluation is not available. However, the main expectation related to its creation seems to be fulfilled: Using INTERSCORE it is easy to quickly create complex, error-free scores. Moreover, if desired, nuances of articulation giving a feeling of "real performance" can be preserved in the editing process.

ACKNOWLEDGEMENT

This research was supported in part by grants from the National Science and Engineering Research Council of Canada.

REFERENCES

[1] M. Balaban, "The set of tonal-music-strings - a structurally unambiguous representation for tonal music pieces," Weizmann Institute of Science Rep. CS81-28, 1981.

[2] W. Buxton, W. Reeves, R. Baecker, et al., "The use of hierarchy and instance in a data structure for computer music," Computer Music Journal, vol. 2, no. 4, pp. 10-20, Dec. 1978.

[3] W. Buxton, R. Sniderman, W. Reeves, et al., "The evolution of the SSSP score editing tools," Computer Music Journal, vol. 3, no. 4, pp. 14-24, Dec. 1979.

[4] P. Casella, AlphaPlus Tutorial Manual. Palo Alto, CA: Syntauri Corp., 1982.

[5] J. D. Foley and A. van Dam, Fundamentals of Interactive Computer Graphics. Reading, MA: Addison-Wesley, 1982.

[6] S. Haynes, "The musician-machine interface in digital sound synthesis," Computer Music Journal, vol. 4, no. 4, pp. 23-44, Dec. 1980.

[7] H. S. Howe, Electronic Music Synthesis. New York, NY: Norton 1975.

[8] R. J. Jigour, Metatrak II User's Manual. Palo Alto, CA: Syntauri Corp., 1982.

[9] G. Krasner, "Machines tongues VIII: The design of a Smalltalk music system," Computer Music Journal, vol. 4, no. 4, pp. 4-14, Dec. 1980.

[10] M. V. Mathews and F. R. Moore, "GROOVE - A program to compose, store, and edit functions of time," Communications of the ACM, vol. 13, no. 12, pp. 715-721, Dec. 1970.

[11] N. Meyrowitz and A. van Dam, "Interactive editing systems: Part I and II," ACM Computing Surveys, vol. 14, no. 3, pp. 321-415, Sep. 1982.

[12] Music System Operating Manual, Scotts Valley, CA: Mountain Computer, 1981.

[13] K. Reynolds and R. Jigour, Composer's Assistant User's Manual. Palo Alto, CA: Syntauri Corp., 1983.