

# Improvements to a Pen-Based Musical Input System

Elizabeth Ng, Tim Bell, and Andy Cockburn  
Department of Computer Science  
University of Canterbury  
Christchurch, New Zealand  
eliza, tim, andy@cosc.canterbury.ac.nz

## Abstract

*This paper describes an improved design for pen input of music into a computer system. A usability analysis of an existing system is performed, and improvements based on the analysis are implemented and tested. New capabilities are also added. The resulting system is tested for both usability and speed of input, with good results in both areas.*

## 1. Introduction

Computers are widely used as an aid to musicians, including for composing, arranging, editing and printing scores. However, many musicians find the interface (typically a mouse, computer keyboard and/or synthesiser keyboard) less natural than the traditional pencil and manuscript. The work reported here explores ways to reduce this gap by using a pen computer with digital ink to simulate the traditional music writing environment. Not only is the environment closer to handwriting, but we are able to accelerate data entry by using short-hand pen gestures.

A set of gestures for pen entry of music was reported in 1996 [2], and this paper describes significant enhancements and refinements to the system. The reader is referred to the earlier paper for a summary of related work, and for an analysis of the frequency of the symbols in samples of music.

The original Presto system, *Presto1*, was developed by studying videos of musicians writing music on paper manuscript, and then designing gestures that mapped closely to the sequence of gestures observed. In particular, it was found that musicians often drew partial symbols, and came back to complete them. For example, it was common for a number of note-heads to be drawn, and then the stems to be added later. The Presto1 gesture set was then implemented on a pen computer, and further iterative refinement was applied after user evaluation. The final system appeared to be

comparable in speed to hasty handwriting, and up to three times faster than other methods of data entry.

The *Presto2* system reported here is a further refinement of Presto1. Four main areas have been addressed:

- The recognition of gestures has been improved by changing the gesture set slightly, and improving the recognition algorithms.
- Several important editing functions have been improved or added, including deletion and undo.
- Presto1 had some problems with gestures for *beaming* — linking several notes together with a horizontal beam — and this has been addressed by surveying users about potential gestures, and evaluating a new approach.
- The audible and visual feedback from Presto has been improved to help users with data entry.

This paper discusses each of these four improvements, and reports on an evaluation of the usability and speed of the improved system. More details about the Presto2 system can be found in Ng [18].

## 2. The Presto system

The Presto gesture set has been designed to enable fast input of music, but is also intended to be quick to learn. It uses shorthand gestures that are generally mnemonic of the symbols that they produce. The main gestures are shown in Table 1. For example, the gesture for a crotchet note is a single dot with the pen, representing the head of the note. This is the most common gesture [2], which is why it is designed to be very simple.

To produce a beamed group the user first draws the notes as crotchets (with the dot gesture), and then draws any line through the stems of the crotchets. This sequence is similar to the way most musicians write music on paper. Allowing

this sequence means that in the middle of the gesture the music may not be semantically correct (there may be too many beats in a bar), so checking must be done later, if at all.







Because the gestures are generally shorter than those required when using pen and paper, and less accuracy is needed (for example, beams can be drawn anywhere across the stems), the Presto system is faster than writing on paper.

### 3. Simplifying gesture recognition

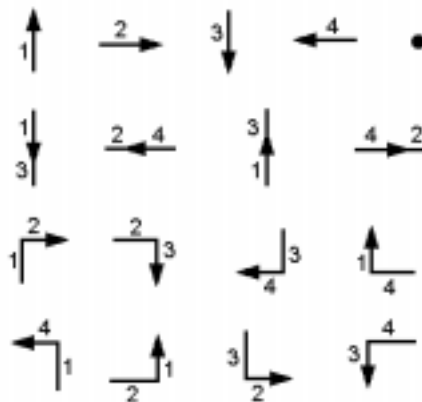
In the original Presto system (Presto1) there were some problems with recognising gestures accurately, which meant that users were sometimes frustrated by nothing appearing after a gesture was made, or even worse, the wrong symbol appeared. This was partly due to the design of some gestures, and partly due to the attributes of the gestures used to recognise them. In pen systems, the following are the main attributes of a gesture that are available:

- direction,
- context (for example, whether it started on a note-head or on a blank area),
- speed (for example, a “flick” of the pen compared with normal writing speed),
- length,
- shape (for example, straight, curved, or wavy), and
- modifications (such as pressing a button on the pen barrel or a key on the keyboard while doing the gesture).

Presto1 used the first four of these attributes. The direction was quantised to the nearest octant (north, northwest, west, etc.), which made diagonal gestures possible, but it meant that the user had to be able to draw a line within a segment of the intended direction. The more common

	Symbol	Gesture	Context
Note			Dot on pitch
Beam			Line through stems
Barline			Vertical line through staff

**Table 1. Example gestures from the Presto system**



**Figure 1. Seventeen gestures available using not more than two directions (an arrow in the middle of a line represents the second direction of a double stroke)**

directions (N, S, E, W) in Presto1 had segments that subtended 52 degrees each, and the other segments subtended 38 degrees each. Drawing within these angles is not too difficult, but errors occurred sufficiently often that it could be frustrating to use. In Presto2 the direction was limited to just four directions, and the few gestures that relied on diagonal lines have been redesigned. Although this seems like a crude quantisation, it is very effective because it gives a very large tolerance for recognising gestures, and users only have four possible directions to remember instead of eight. Similar observations on the importance of maximising “sloppiness space” are made by Goldberg and Richardson [6] in the design of “Unistrokes”.

The other attribute that caused difficulty in Presto1 was the distinction according to the speed of the stroke; a normal stroke from a proficient user might be mistaken for a fast one, and all users found fast strokes harder to draw accurately. These problems were solved by not using this attribute, again simplifying recognition and making the system more predictable to users.

Even though fewer attributes are used, there are still more than enough gestures available, particularly if gestures are allowed to have one direction change. Figure 1 shows the 17 gestures available based only on direction of the strokes in four quadrants (up to two directions per gesture), plus the dot gesture.

Table 2 shows the gestures that were changed to allow for the different attribute recognition; it also shows new gestures that were added for rests and double bars.

Symbol/effect	Gesture	Context
 default note		Dot on pitch
 default note, stem up		Start on pitch, draw up then down
 default note, stem up		Start on pitch, draw up then down
 double default duration		Double dot on pitch
 quaver rest		On staff
 crotchet rest		On staff
 minim rest		On staff
 semibreve rest		On staff
 barline		Over staff
 double bar-line		Next to single bar-line
 sharp,  double sharp		Start on note-head
 flat,  double flat		Start on note-head
Change stem up		Start on note-head, draw up then down
Change stem down		Start on note-head, draw down then up
Add durational dot		Start on note-head or rest
Remove durational dot		Start on note-head or rest
Double duration		Start on note-head or rest, draw right then left
Halve duration		Dot on note-head or rest
		Start on note-head or rest, draw left then right
Beam		Over note stem or rest
		Through stems
Delete symbol		Dot with button on symbol
Delete whole beam		Double dot with button on beam
Delete double barline		Double dot with button on barline
Undo		Off staff
Redo		Off staff
Select symbols		Draw around symbols with button to select; draw out of staff with button to unselect

**Table 2. Gestures revised or added for Presto2**

## 4. Enhancing editing operations

Presto1 concentrated on gestures for entering symbols, but it became apparent that editing is an important part of data entry, as slips and errors must be allowed for. Five aspects of editing were added or improved in Presto2: deletion, undo, selection, automatic scrolling, and selection of staff size.

In Presto1 the deletion gesture was a wavy line (to represent scribbling out). This gesture was difficult to recognise reliably, as users sometimes had the waves too shallow (so it looked like a straight line) or had too few (so that it looked like an up/down gesture).

A number of deletion gestures from other systems were considered, and they are listed in Table 3. The shortest deletion gesture in the table is the dot gesture in the Schoolchildren system [10]. In this system, users draw a dot on an object to delete it. Goldberg and Goodisman [5] did not assign a specific deletion gesture to the text entry system, but to imitate pencil and paper techniques of editing text, they suggested clicking the button on the pen barrel and “rubbing” the object off.

Most of the deletion gestures in the table are long, slow, and difficult for users to draw, or difficult for the system to recognise. Tapping with the pen, or drawing the dot gesture, as in the Schoolchildren system, is the shortest gesture but it is already used in Presto1 to draw a crotchet. This gesture can be modified to require pressing the button on the pen barrel or a key on the keyboard for deletion. In addition to the advantage that it is short, this gesture will not cause accidental deletions because users have to press the button deliberately.

An undo facility is also useful during data entry; in fact, often users are tempted to use delete when the real purpose is to undo the last action. Many systems enable users to execute undo through menus, but that is not desirable in pen systems because gestures should be the main input method, with menus being used for less common commands. Only some pen systems provide gestures to execute undo, while others do not have this command at all. Undo gestures for other systems are shown in Table 4.

The absence of the undo command made editing music in Presto1 laborious, especially when music representation has to be precise. When the user cannot undo a regretted action, editing in the music system becomes frustrating. Exploratory learning is encouraged when undo is available.

An undo gesture itself should not be easily misrecognised because the user will not be able to undo the previous error as well as the undo gesture that is misrecognised (assuming a simple undo/redo model). The button-and-dot gesture is not used for any command when it is drawn in a blank area, thus it could be used to execute undo. This is fast and simple but may create problems with deletion since

both deletion and undo are important and common.

Another simple and suitable candidate is a horizontal line. There is no action associated with the gesture if it is drawn on empty space, that is, not on a musical symbol. A line to the left could be used as an undo gesture, and a line to the right could be used as a redo gesture. These gestures are suitable and mnemonic, because the left stroke is similar to a left arrow for going backwards, and the right stroke is similar to a right arrow for going forwards. This is the gesture that was chosen for Presto2.

A selection gesture was added so that operations could be performed on multiple notes. As well as traditional cut/copy/paste operations, musicians often want to perform various modifications to music, such as transposition, augmentation, and stem direction changes.

There are two methods that could be used to select objects. The first one is the same as that in text editors. That is, the user selects text by dragging over them with a mouse or a pen. This method is not suitable for selection in music, because music is two dimensional and the area being selected can be an unusual shape.

The second method is to enclose the selected objects with a rectangle or an enclosure (closed curve). The gesture drawing a rectangle is used in Landay’s interface design editor [14], and the enclosing gesture is suggested by Buxton et al. in their music system [3]. GEdit [12], the Air traffic control system [4], and Tivoli [13, 20] also used the enclosing gesture to select objects. In these editors, the objects are not fixed in their two dimensions; the square and the enclosing gestures are very suitable for such objects.

The selection gesture cannot be a straight line to select a rectangle, since it will conflict with other gestures like the beam, and the add and remove dot gestures. Thus, the enclosing gesture is a better option.

Presto2 has an automatic scrolling function that is not provided in Presto1. This function prevents the musical symbols from flowing off the screen if the user does not scroll, and assists the user while entering music so that they do not need to stop to scroll manually.

In Presto2, users can change the size of the staff to small, medium, or large according to their preferences. At the same time they can also choose to remove the cursor if it is obstructing their work in the system.

## 5. User-oriented beaming techniques

Beams are used to group notes together in music. Figure 2 shows an example of how beams are used to make the notation more readable (by chunking notes) and also to show the main beat (beamed groups generally make up a whole beat). They are also used in vocal music to show how syllables map onto multiple notes.













System	Effect	Gesture
Air traffic control system [4]	delete	
Apple Newton [16]	delete	
GEdit [12]	delete	
Gesture Mosaic [1]	delete	
GO Penpoint [17]	delete	 or 
Graffiti [9]	delete	
	backspace	
Network design system [15]	delete	
Schoolchildren system [10]	delete	dot on object
Text entry system [5]	delete	click pen button and rub off
Unistrokes [6]	backspace	
Windows for Pen Computing [19]	delete	
	delete word	

Table 3. Gestures used for deletion in other systems




System	Gesture
Air traffic control system [4]	
Network design system [15]	
Windows for Pen Computing [19]	

Table 4. Gestures used for undo in other systems



Figure 3. Using the logical beaming rule



Figure 2. Using beaming to group notes together

The beaming gesture in Presto1 involves drawing a line through the notes to be beamed. A line through a group of crotchets (each counting as a single beat) would turn them into beamed quavers (each worth half a beat). The general rule used for the gesture in Presto1 was that a line through a note stem would halve the length of the note, and beam it with the others that had their stems crossed. However, it turns out that this general rule creates anomalies in some situations. Figure 3 shows three situations where the results of the beam gesture are not what the user might expect.

We observed that musicians using the system thought of the beaming gesture *visually*, that is, as if it was drawn on paper. The *visual* model of beaming is that a beam is added between every pair of notes crossed, compared with the *logical* rule, that every note crossed has its length halved. For example, the same situations that are in Figure 3 (using the



**Figure 4. Using the visual beaming rule**

logical rule) have very different outcomes when they use the visual rule in Figure 4.

In order to verify the perceived preference for the visual rule, we surveyed ten musicians about drawing beams. Each musician was given a demonstration of Presto1 so that they had an understanding of the shorthand pen-based input method, and they were then given a questionnaire that had two parts to it. The first part required them to draw beam gestures over notes to produce a given effect, and the second part required them to show the effect that they expected certain beaming gestures to have in various contexts. The subjects were not made aware of the distinction between the visual and logical rules.

The questionnaires were analysed by categorising the replies according to whether they followed the visual rule, the logical rule, or neither. No subject used only one rule exclusively, but all used the visual rule more often than the logical rule. On average, the subjects used the visual rule in 63.2% of their answers, the logical rule in 12.2%, and some other rules in 24.6%. This confirmed that the logical rule, while appealing to computer scientists, is not natural for people who have been writing music on paper, and Presto2 was modified to use the visual rule.

Another problem with beaming was the tolerance allowed for the gesture. Many first-time users drew beams very carefully, starting close to the top of the first stem and finishing at the top of the last one. Experienced users realised that any line drawn roughly through the stems would do, but if they were too careless the gesture might continue on to the next group.

Samples from the beam survey were used to observe where users started and ended their beam gestures. These samples showed that the tolerances for both ends of the beam gesture should allow more for overruns than under-runs. The samples also show that the length of both ends of the beam could be related to speed; that is, when users draw gestures slower they tend to be more careful, but when they draw faster the gestures may overlap with the next beamed

group. This suggests allowing more tolerance for faster gestures, although we did not implement this. However, if the results are not what users expect, they would usually do it again slowly.

## 6. Enhanced audio and visual feedback

Feedback is an important part of any user interface, and because music is inherently auditory it is particularly appropriate to use this kind of feedback. Sound has the particular advantages [11] that the user's direct attention is not required for it to be noticed and it can reduce the bandwidth required to send information through the visual channel. Because the system is being used by musicians, they are more likely to be able to distinguish subtle differences in sound (such as notes that are close together) and so we would expect to be able to communicate more information through sound than one might normally. Sound also has problems, including the lack of privacy (unless headphones are used), and the lack of persistence. Because of the potential annoyance from sound (for the user and others nearby) it is important that the sound can be switched off or controlled by the user.

Usually when musicians enter music they “hear” the notes in their head, and so playing notes as they are entered provides a good mechanism for detecting errors, since the user will hear a conflict between the imagined note and the actual note. The Presto2 system was connected to a synthesiser so that it could play notes as they are entered.

The system also provides audible feedback on deletion of a note, errors, and inserting a rest (unpitched percussive sounds are used), and the ability to play back part or all of the tune. The user is provided with a control to switch off the sound for selected actions.

The visual feedback from Presto has also been reviewed. To make it easier to draw notes on ledger lines (above and below the staff), Presto2 provided users with “scaffolding” for drawing these notes in the form of temporary lines when the pen was in those areas. A simple help system was also implemented to suggest alternative gestures to use and inform users what kinds of errors have occurred, such as gesture errors and system errors. This system simply displays messages at the bottom of the window, which the user can ignore if desired. The playback feature also gave visual feedback in the form of a pointer that identifies which part of the music is currently being played.

## 7. Usability evaluation

The usability of the improved system was evaluated by having eleven musicians use it and then fill out a questionnaire that covered the main parts that had been modified.

Subjects were given at least 15 minutes to become familiar with the system, although some of the more enthusiastic users spent significantly more time on it. The subjects were composition students at the University of Canterbury.

Overall the subjects were enthusiastic about the system, and all were positive about using it. Comments about the system included “very ingenious,” “great,” and “definitely useful.” One subject felt that it was “so much easier and quicker and more flexible than a mouse,” and another described it as “what you are used to.”

The users also identified a number of problems with the system. Three subjects encountered parallax errors in Presto2. One subject kept entering notes at the wrong pitch, and the other two had difficulty deleting beams because they could not aim accurately on the beams. Further work in Presto could design features that help users aim more accurately, such as a system to calibrate the pen on the staff.

Five subjects either could not understand the concept of context-sensitive gestures or found it inconvenient to apply. They expressed that it was awkward to have the same gesture with different commands, and preferred to draw a gesture anywhere on the screen which should be interpreted as the intended command.

There are some gestures that five subjects found difficult to draw and were not intuitive, especially the gestures that require the users to retrace the lines. Further work in Presto could change the gestures to simpler ones that are more mnemonic.

Most subjects liked or needed the tracker during playback in Presto2, but three subjects could follow the music along the score very well without it. Further work in Presto could let users choose to have the tracker move from symbol to symbol or bar to bar, or not have a tracker at all during playback. The played symbol could also be highlighted with a different colour.

The suggestions and error messages at the bottom of the window were not helpful for six subjects, but one other subject wanted them to be more visible. Further work on the help system could expand it into an intelligent help system, like Tivoli [13, 20], with icons and animation to better illustrate the suggestions, and with an option to switch off.

Six subjects preferred Presto to operate without the mouse or the keyboard. Subject 1 explained that one hand would be holding the pen, and the other hand on the piano keyboard. Thus, the mouse and keyboard should be optional devices in Presto.

The subjects provided a large list of features that are not in Presto2; these include missing functions, missing feedback, and missing symbols.

## 8. Speed of input

One of the main goals of Presto is to accelerate the speed of music input. This was measured for Presto2 using a copying task, although the relative speed of different input methods may vary with the nature of the task. For example, for proofreading there may be only occasional minor corrections, and the real measure of speed is the time taken from noticing an error to its correction.

Only three users tested the system for speed, since we were interested in the maximum performance possible for an “expert” user<sup>1</sup>. The test task involved copying a selection of Haydn’s String Trio “Divertimento No. 15” (Robbin Landon edition) [7]. This piece was chosen because speed performance results were already available for entering the same piece in Presto1, and work on Presto1 made these results indirectly comparable with other methods of music data entry, such as optical music recognition (OMR) and a conventional computer music editing system [8].

The average time for the three subjects entering music in Presto2 is 72% of that required for the same subjects to copy by hand; from the results reported with Presto1 [2] this implies that entry time on Presto2 is 30% of the time using OMR or 22% of that using a keyboard system. In other words, Presto2 can be about three times as fast as OMR and more than four times as fast as conventional computer music entry methods. The best time achieved for the copying task in Presto2, which is 6 minutes and 53 seconds, is about 27% faster than the best time in Presto1, which is 9 minutes and 24 seconds.

It has been observed that in pen-based systems more time can be spent between gestures than doing them [6], indicating that the cognitive load imposed by a gesture set needs to be kept low. One reason that pen-based input (whether on paper or computer) is faster is that the user is manipulating the visual representation of the music directly, rather than using a keyboard or palette of musical symbols. Also, gestures indicate the note’s pitch and duration at the same time, rather than forcing the user to decompose the note for data entry. This maps more directly to both the visual and aural representation of music, which means that the more “natural” interface can be faster because the musician has a lower cognitive load.

## 9. Conclusion

The user surveys have shown that the Presto2 system has strong user acceptance, and most of the modifications to Presto1 have been worthwhile. In particular, the simpler gesture recognition has made the system more reliable to use, and the enhanced editing operations improve input

---

<sup>1</sup>The three users were all involved in developing the system.

by allowing corrections to be made efficiently. The beaming gestures have been improved significantly by using a visual rule instead of the logical rule, and the use of audible feedback has increased the back-channel by exploiting musicians' aural ability.

The speed tests have demonstrated that Presto2 appears to be faster than writing on manuscript, or at least the same speed but with better quality and more flexible output.

A related system, called "Pen-Based Music Editor" is described by Silberger [21]. This editor originally used a palette of symbols that users could select with the pen, but it was improved by having the user make some form of gesture at the point where a note is to be inserted. In an early version the gesture was simply the length of time the pen was held down, while a more recent approach uses a pop-up menu that offers the user a selection of symbols. The system is also able to "learn" new gestures. With a menu-based system, gestures only use two possible pen directions, and the mapping between the gesture shape and the symbol obtained is more abstract than Presto's. Presto must also rely on pop-up menus for unusual symbols, but most symbols can be entered without the user having to recognise entries in a menu.

The Presto system is likely to be most useful when integrated with other forms of music input, including OMR, and keyboard (piano and computer) and mouse input.

## References

- [1] American-Tele-Systems, available from: <http://www.amtelcom.com/Mosaic/gmgif/quickref.gif>. *Gesture Mosaic [online]*.
- [2] J. Anstice, T. Bell, A. Cockburn, and M. Setchell. The design of a pen-based musical input system. In *Proceedings of the Sixth Australian Conference on Computer-Human Interaction*, pages 260–267, Los Alamitos, California, 1996. IEEE Computer Society.
- [3] W. Buxton, R. Sniderman, W. Reeves, S. Patel, and R. Baecker. The evolution of the SSSP score editing tools. *Computer Music Journal*, 3(4):14–25, 1979.
- [4] S. Chatty and P. Lecoanet. Pen computing for air traffic control. In *Human Factors in Computing Systems: "Common Ground" CHI '96 Conference Proceedings*, pages 87–94, New York, 1996. ACM.
- [5] D. Goldberg and A. Goodisman. Stylus user interfaces for manipulating text. In *Proceedings of the Fourth Annual ACM Symposium on User Interface Software and Technology*, pages 127–135, New York, 1991. ACM.
- [6] D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Human Factors in Computing Systems: "Bridges Between Worlds" CHI '93 Conference Proceedings*, pages 80–87, New York, 1993. ACM.
- [7] J. Haydn. Divertimento No. 15 C-Dur für 2 Violinen und Violoncello Hob. In H. C. R. Landon, editor, *Diletto Musicale DM915: Joseph Haydn Streichtrios*, volume 16. Doblinger, Vienna, 1981.
- [8] W. B. Hewlett and E. Selfridge-Field. How practical is optical music recognition as an input method. In W. B. Hewlett and E. Selfridge-Field, editors, *Computing in Musicology: An International Directory of Applications*, pages 159–166. Center for Computer-Assisted Research in the Humanities, Stanford, California, 1994.
- [9] Hewlett Packard. *HP OmniGo 100 Quick Start and User's Guide*, 1995.
- [10] T. D. Kimura. A pen-based prosodic user interface for schoolchildren. *IEEE Multimedia*, 3(4):48–55, 1996.
- [11] G. Kramer. An introduction to auditory display. In G. Kramer, editor, *Auditory Display: Sonification, Audification, and Auditory Interfaces*, pages 1–77. Addison-Wesley, Reading, Massachusetts, 1994.
- [12] G. Kurtenbach and B. Buxton. GEdit: A test bed for editing by contiguous gestures. *SIGCHI Bulletin*, 23(2):22–26, 1991.
- [13] G. Kurtenbach, T. P. Moran, and W. Buxton. Contextual animation of gestural commands. In *Proceedings of Graphics Interface 94*, pages 83–90, Toronto, 1994. Canadian Information Processing Society.
- [14] J. A. Landay and B. A. Myers. Interactive sketching for the early stages of user interface design. In *Human Factors in Computing Systems: "Mosaic of Creativity" CHI '95 Conference Proceedings*, pages 43–50, New York, 1995. ACM.
- [15] K. V. Mardia, N. M. Ghali, T. J. Hainsworth, M. Howes, and N. Sheehy. Techniques for online gesture recognition on workstations. *Image and Vision Computing*, 11(5):283–294, 1993.
- [16] A. Meyer. Pen computing: A technology overview and a vision. *SIGCHI Bulletin*, 27(3):46–90, 1995.
- [17] D. Millen. Pen-based user interfaces. *AT&T Technical Journal*, 72(3):21–27, 1993.
- [18] E. P. Y. Ng. Development of the Presto pen-based music editor. Master's thesis, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1998.
- [19] D. Ouellette. *The Pen Connection: A Guide to Pen Computing*. McGraw-Hill, New York, 1995.
- [20] E. R. Pedersen, K. McCall, T. P. Moran, and F. G. Halasz. Tivoli: An electronic whiteboard for informal workgroup meetings. In R. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, editors, *Readings in Human-Computer Interaction: Toward the Year 2000*, pages 509–516. Morgan Kaufmann, San Francisco, California, 1995.
- [21] K. Silberger. Putting computers in control. *IBM Research*, 34(4):14–15, 1996. also available from: [http://www.research.ibm.com/resources/magazine/1996/issue\\_4/musiceditor496.html](http://www.research.ibm.com/resources/magazine/1996/issue_4/musiceditor496.html).