An integrated, multimedia environment for the analysis of human-computer interactions in Logo

Ross Purves
School of Arts and Humanities
Institute of Education, University of London
20 Bedford Way, London WC1H 0AL, United Kingdom
rmpurves@ntlworld.com

Abstract

This paper discusses the development of an integrated, multimedia environment for the collection and analysis of dribble data. Based around *SONAR* music sequencing software, this system allows the researcher to review audio and video recordings of *MSWLogo* users' discussions and gestures concurrently with synchronised dribble data. The latter is captured from Logo, not via the traditional text file, but through its encryption into MIDI (Musical Instrument Digital Interface) meta-event messages that are recorded with time-stamps in *SONAR*. The resulting data can not only be triangulated with the concomitant audio and video, but also exported for further analysis. This system is being used to analyse potential epistemological synergies emerging from the use of Logo-based tools to bring together learning in music and mathematics.

Keywords

Dribble files, human-computer interaction, (MIDI) musical instrument digital interface, data triangulation, multimedia

1. Introduction

1.1. The dribble file approach

Many Logo implementations feature the ability to generate composite transcripts of all text-based human-computer interactions through the use of 'dribble files'[3]. Such files can provide programmers and researchers with valuable sources of data on computer users' actions. Some of the advantages of using dribble files to collect data include:

- the opportunity to study evolving, dynamic records of user activity. Dribble files have the potential to offer 'a window into the *process* of learning' (Barab, Fajen, Kulikowich and Young, 1996: 186 emphasis in original). In contrast, pre- and post-test assessments only provide snapshots of user performance at particular stages (Barab, Bowdish, Young and Owen, 1996);
- the ability to collect data in an unobtrusive and transparent manner (Barab, Fajen, Kulikowich and Young, 1996; Scherly, Roux and Dillenbourg, 2000). Dribble files

[3] Dribble files are also variously referred to as 'transcript files' (Harvey, 1997), 'log files' (Barab, Fajen, Kulikowich and Young (1996) and 'computer tracking systems' (Mills, 2001).

can thus be used to log user activity in real-world learning situations as opposed to contrived lab conditions (Mills, 2001);

- data is already in electronic form. This simplifies any processing or additional coding prior to analysis (Gay and Mazur, 1993).

Hoyles and Noss (1992) offer an indicative example of how the analysis of Logo dribble files can assist researchers. They describe the experiences of a pair of thirteen-year-olds playing the 'target game'. The aim of this game was to use recursive multiplication to obtain a number as close as possible to 100 from a starting point of 13. The pedagogical objective was for users to understand the effect of multiplying by numbers between 0 and 1 and by negative numbers. The researchers provide a commentary on the pair's dribble file, plotting mistakes and perceived misconceptions before subsequently highlighting evidence of increasing aptitude and confidence. For Hoyles and Noss, the dribble file provided a means of assessing the children's unfolding mathematical strategies as the game progressed.

1.2. Extending the Logo dribble file

Unfortunately, over a decade on from this study by Hoyles and Noss, the facilities to generate dribble files from Logo remain rudimentary. As then, most implementations of the language work on the simple principle of logging copies of command-line interactions in ASCII[4] text files. Though this method affords a precise record of all interactions with the computer, it cannot provide investigators with any insight into users' associated contextual experience (Gay and Mazur, 1993). In contrast, work in the wider field of educational human-computer interaction research outside Logo has developed the dribble file concept much further. Time-stamping dribble file entries has been shown to be particularly effective in locating difficult or poorly designed aspects of user interfaces (e.g. David, 2003). A number of researchers have also supplemented dribble data with parallel audio-visual recordings (e.g. Gay and Mazur, 1993). This paper discusses the development of an integrated, multimedia environment for the collection and analysis of Logo dribble data that incorporates these additional features. The intention is to offer a possible solution for Logo researchers who wish to extend the range and depth of recorded dribble data in order to better evaluate the user experience.

The present study forms part of a much larger project. This is exploring the potential to engender epistemological synergies by bringing together learning in music and mathematics through the use of Logo-based tools (Purves, 2001; 2005). The need for such an analysis environment arises from a concern to capture, moment-by-moment, the actions of pairs of computer users as they work with these tools. In order to track users' conceptual development as accurately as possible, audio, video and text-based dribble data is recorded simultaneously. Considered individually, the recordings within each of these domains offer distinct analysis possibilities, but such possibilities are enhanced significantly when the three domains are triangulated (Mills, 2001). Gay and Mazur (1993) refer to the composite data arising from this type of triangulation as a 'thick description', and assert that this can 'provide multiple levels of integrated information about users' attitudes, needs, and use of program features' (p. 46). By logging data in these domains simultaneously and in real-time, researchers have the facility to replay them together, thus offering a 'virtual' real-time re-creation of users' actions.

Some developments in Logo dribbling techniques have recently taken place. Gorman (2001; 2003; 2005) added procedures to *MSWLogo* (Mills, 2002) that time-stamp dribble file entries

---

[4] 'American Standard Code for Information Interchange'. A system in which alphanumeric characters are represented by numbers between 0 and 127.

and log mouse coordinate data. Unfortunately, these individual innovations do not appear to have been influential on the design of publicly-available Logo implementations. Thus, like Gorman, this study supplants the dribble facilities already present within *MSWLogo*. However, dribble data is captured not via the traditional text file, but through its encryption into Musical Instrument Digital Interface (MIDI) messages.

2. Developing an integrated, multimedia analysis environment for Logo dribble data

Based around *SONAR* music sequencing software (TTS, 2003), this environment allows the researcher to review audio and video recordings of *MSWLogo* users' verbal discussions and physical gestures concurrently with synchronised text-based dribble data. Once stored in this way, the potential exists for dribble data to be exported for further, time-referenced analysis using statistical software. Markers can be placed on the sequencer's timeline to aid navigation between points of interest and create coding strategies.

2.1. The Musical Instrument Digital Interface and the type one text meta-event

MIDI (IMA, 1983) is the standard networking protocol for the real-time exchange of musical performance data between electronic musical equipment (such as synthesizers, samplers, recording equipment and, of course, computers). The messages transmitted over a MIDI network provide a very detailed description of a musical performance, but do not convey any actual digital audio data. Sequencing software is used to record, edit and playback MIDI messages in real-time. Modern sequencing packages such as *SONAR* can also record digital audio data alongside MIDI and these provide musicians and sound engineers with powerful 'virtual' recording studios.

The Standard MIDI file (SMF) allows users to save MIDI messages in a non-proprietary format. In addition to MIDI performance data, the SMF supports a series of special messages termed *meta-events*. These messages provide a means of conveying important non-real-time information about a piece of music (such the tempo, key and time signatures). Another of these special messages, termed the *type one text meta-event*, allows users to insert free-text comments within SMFs. Since they are intended to hold ASCII-encoded data, type one meta-events are perfectly suited for storing dribble data generated from Logo. In this system, each line of text from the Logo command line is stored as a separate meta-event message. Furthermore, since *SONAR* time-stamps all MIDI messages, each line of text is automatically stored in the SMF with a unique, sub-second generation time-code. Figure 1 shows a hexadecimal 'dump' of a short SMF containing a type one text meta-event.

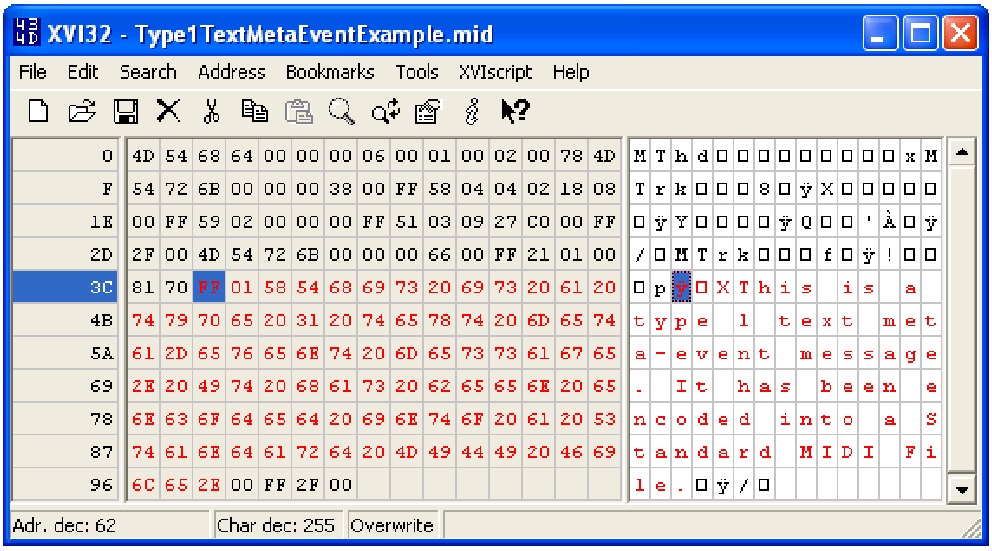


Figure 1. Example of a Standard MIDI File containing a comment as a type one meta-event message. The left-hand window contains a hexadecimal 'dump' of the file, whilst the right-hand window contains an ASCII character conversion. The cursor has been placed at the beginning of the message. `FF` is the hexadecimal code used to indicate a meta-event message, `01` is the type and `59` (89 in decimal) is the number of subsequent bytes comprising the comment. The two bytes preceding the highlighted FF form the message's time-stamp, recorded using the delta-time format (see Kientzle (1998) for a description of delta-timings).

Unfortunately, meta-event messages are only used within SMFs and do not form part of the MIDI real-time networking protocol (Kientzle, 1998). This means that it is not possible to transmit them from one piece of equipment to another. Thus in this system, lines of dribble data are first encoded into ASCII and then transmitted as *system exclusive* MIDI messages[5]. Unlike meta-events, system exclusive messages *can* be sent in real-time over a MIDI network. Once recorded by *SONAR* and saved in an SMF, a specially-written, self-authored utility program - *sysex2meta* – is used to convert the system exclusive messages into text meta-events. The processed SMF is then reloaded into *SONAR*.

For the system to operate, *MSWLogo* must be configured to output copies of all typed commands and computer feedback as ASCII-encoded system exclusive MIDI data. This is accomplished by using the `midimessage` primitive. The specially-written procedure `display` is presented below as an example of this method. This replaces `print` in user-defined procedures and is used to send copies of screen feedback over MIDI. To aid subsequent analysis, all such feedback is preceded with the string 'feedback:' when output over MIDI.

```
to display :text
    print :text; Print the text to screen before sending over MIDI
    localmake "outlist [70 101 101 100 98 97 99 107 58]; 'feedback:'
    ignore (midiopen 6)
    foreach :text [
           queue "outlist 32
```

[5] System exclusive messages allow manufacturer-specific information to be sent in real-time over a MIDI network. Typically, this allows a computer to control the internal configuration of MIDI equipment such as synthesizers. Since they are manufacturer-specific, they have no one particular format (beyond starting with the hexadecimal code F0 (240 in decimal) and ending with F7 (247 in decimal)) and so can easily be used to carry ASCII-encoded text.

```
            foreach ? [
                  queue "outlist (ascii ?)
            ]
      ]
      midimessage (se 240 :outlist 247)
      midiclose
   end
```

The `midiopen` command used in the fourth line of `display` is used to establish MIDI communications between *MSWLogo* and the receiving MIDI sequencing software. The latter could be running on a second computer connected via an external USB MIDI interface. In the present setup, however, *MSWLogo*'s MIDI output is configured to use a virtual, 'loopback' MIDI device provided by a third-party freeware utility[6]. This allows all MIDI messages sent from *MSWLogo* to be looped back into *SONAR*, which runs in parallel on the same laptop computer.

2.2. Capturing audio and video data

At the same time as capturing MIDI-encoded dribble data, *SONAR* also records audio data emanating from small, lapel-mounted microphones warn by both members of the pair. The output signals from the two microphones are recorded onto separate digital audio 'tracks' within *SONAR*, thus enabling a useful degree of isolation during subsequent analysis[7].

Like other sequencing packages, *SONAR* can import a pre-recorded video file and play it back concurrently with MIDI and audio tracks. In the present setup, video is first recorded using a discrete digital video camera. As figure 2 shows, the camera is set up to capture participants' non-verbal information such as facial expression and body movement (Goldman-Segal, 1991). The camera's field of view also includes the computer's mouse and keyboard so as to facilitate post-hoc analysis of individual users' appropriation of these controls (Windschitl, 2001).

Figure 2. Framing the video image. Gaussian blur has been applied to the faces in order to protect the participants' anonymity.

Once an experimental session is complete, video data is transferred from the camera to the computer via the IEE1394 interface, stored as an AVI format video file and imported into *SONAR*. Since the video data is recorded separately from the MIDI and audio data, it is

---

[6] The MIDI loopback device used is *LoopBe1*, available from http://www.nerds.de/

[7] Though inevitably, this separation is compromised slightly as a user's microphone will also pick up the voice of their partner in the background.

necessary to sync it up manually before triangulation. Figure 3 summarises the equipment setup used in the analysis environment.

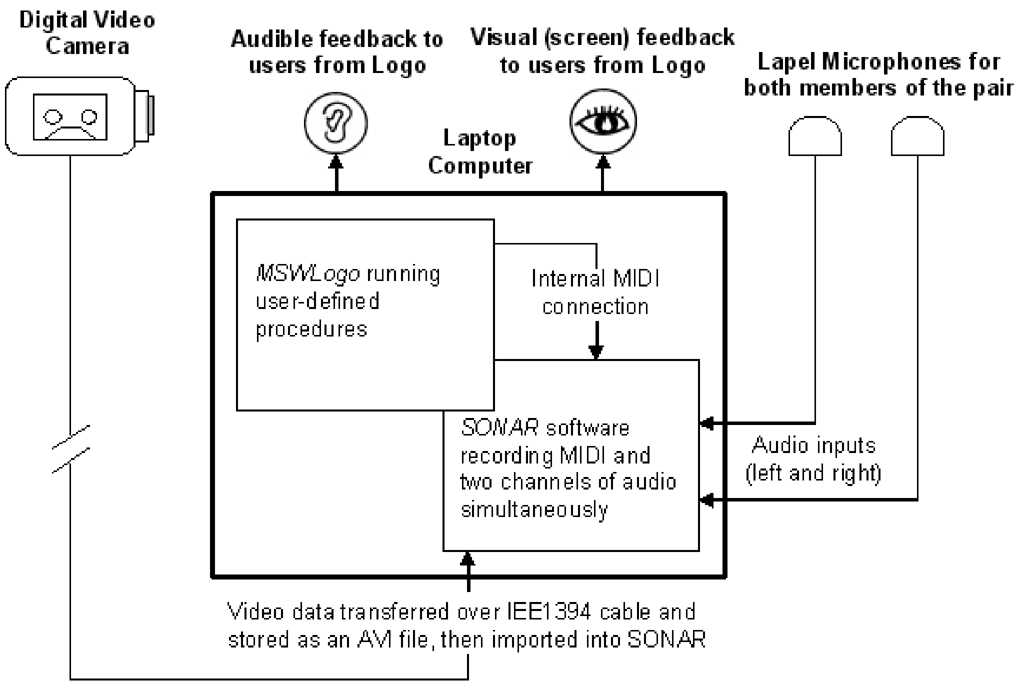


Figure 3. The equipment setup for two users, as used in the pilot experiment.

2.3. Reviewing data in *SONAR*

Once dribble, audio and video data have been recorded and processed, *SONAR*'s user interface is configured as in figure 4 in order to facilitate analysis.

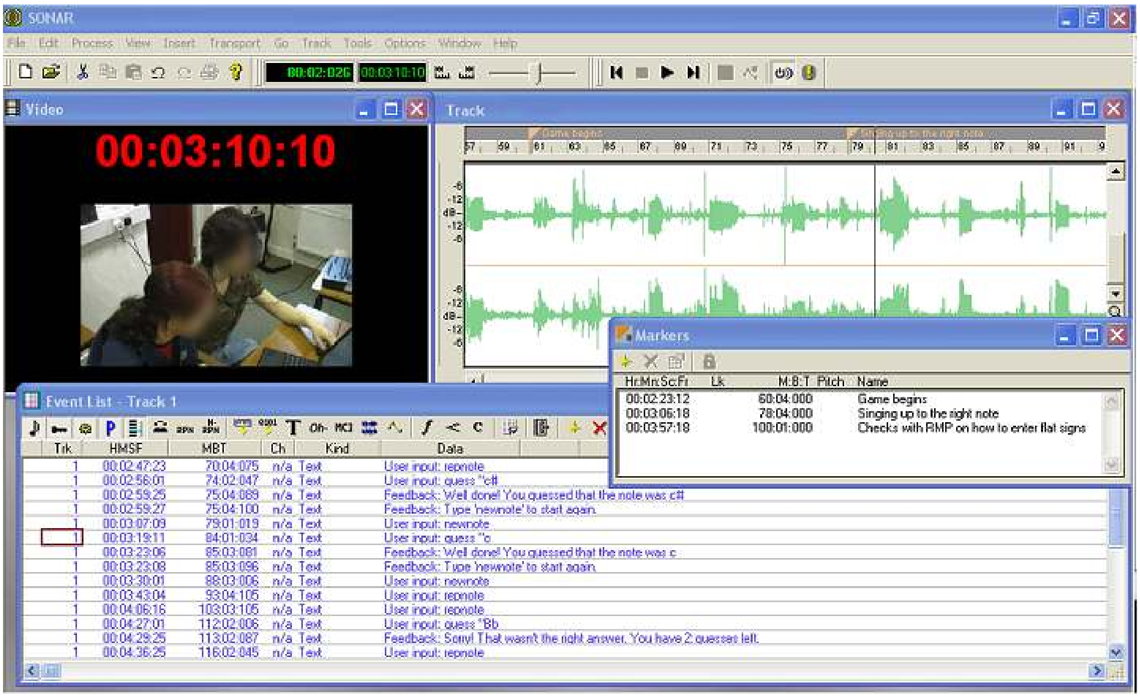


Figure 4. Displaying dribble, audio and video data using *SONAR* software.

The onscreen windows show data from the three recorded domains. Real-time video and time-code are displayed the top left of the screen. The waveform representations of the two digital audio tracks can be seen top-right. The MIDI messages containing the dribble data are visible in *SONAR*'s 'Event List' window (at the bottom of the screen). This also displays a time-stamp for each line of text. As playback progresses, the red square on the left of this window highlights each line of dribble data in turn. A fourth window lists location markers added during analysis. Markers are also displayed in yellow on the sequencer's timeline above the audio tracks.

3. Evaluating the Analysis Environment

3.1. The Pitch Identification Game

A small-scale pilot experiment was designed to evaluate the analysis environment. This was based around a specially-developed game intended to challenge users' musical listening skills. Specifically, the task was to identify by ear randomly-selected pitches between C^4 (261Hz) and B^4 (508Hz) (i.e. one of twelve semitones within a single octave). Though continuing the musical theme, this game did not in itself form part of the larger project discussed in section 1.2. The intention was simply to get a pair of users to discuss a tightly-defined problem. As a musical task, it was hypothesised that the game might elicit specialised forms of non-verbal communication, thus facilitating the collection of a rich multimedia dataset.

The game involved relatively few specially-written Logo procedures. These were accessible directly from *MSWLogo*'s command line. On typing `newnote`, users heard a note lasting for 1.5 seconds and played using a flute sound. Notes could be repeated twice more per game by using the command `repnote`. A maximum of three guesses were available per game by typing `guess` followed by the chosen note. On providing a correct answer, users were rewarded with a congratulatory on-screen message, the sound of a trumpet fanfare and an invitation to start a new game. If unsuccessful, they heard a 'wrong answer' sound (reminiscent of television game shows) and, if unused guesses remained, were encouraged to guess again. Each of these procedures was configured so as to output all textual feedback to `display`. User input was also transmitted over MIDI, but was preceded with the string 'user input:' to distinguish it from the computer feedback.

For the pilot experiment, the pitch game was played by a pair of seventeen-year-old Advanced Level[8] music technology students (one male, one female).

3.2. Evaluating the analysis environment

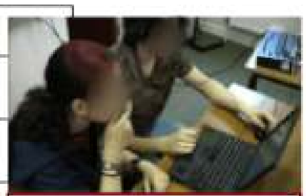
A short excerpt from the data obtained during the pilot experiment is presented in figure 5. Some of the participants' physical gestures, verbal, and non-verbal communications have been highlighted and these serve to demonstrate how raw dribble data can be enriched by audio and video. At several points during the pilot, for instance, the female participant can be heard singing a scale in an attempt to find the correct pitch. Unfortunately, on the first such occasion, she stopped one note short and this led to the pair's first incorrect guess: a *Bb* rather than a *B*. In response to the consequent 'wrong answer' sound, the participants shook their heads and began again. This time, the female participant attempted to 'play' her sung scale on an imagined piano keyboard. Again, the strategy failed, with the male participant exclaiming that their guess was "…well out". Nonetheless, these initial, incorrect guesses appeared to have narrowed down the range of options left and, towards the end of the extract, the pair

---

[8] *Advanced Level* qualifications, or *A Levels* for short, are post-compulsory courses studied by students between the ages of sixteen and nineteen in England and Wales.

used them to deduce the correct answer more methodically ("Was it higher or lower than a *Bb*?", "It was definitely higher than a *G#* as well"). Through discussion, each of the remaining possibilities was eliminated until only the correct *B* remained. As the computer responded with a fanfare, the male participant enthusiastically shouted "Winner!"

Hr:Mn:Sc:Fr	Dribble text
00:03:15:00	Feedback: Type 'newnote' to start again.
00:03:21:23	User input: newnote
00:03:34:26	User input: repnote
00:03:58:08	User input: repnote
00:04:18:23	User input: guess "Bb
00:04:21:17	Feedback: Sorry! That wasn't the right answer. You have 2 guesses left
00:04:28:17	User input: repnote
00:04:28:19	Feedback: Sorry, I've already repeated this note twice!
00:05:00:22	User input: guess "g#
00:05:03:16	Feedback: Sorry! That wasn't the right answer. You have 1 guess left
00:05:51:14	User input: guess "B
00:05:55:09	Feedback: Well done! You guessed that the note was B
00:05:55:12	Feedback: Type 'newnote' to start again.



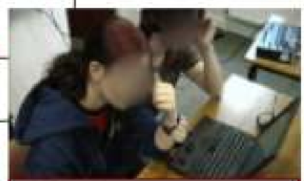


Figure 5. An excerpt from the data obtained during the pilot session. Dribble data has been annotated with partially transcribed audio. Each entry in the dribble file is shown with its time-stamp (hours, minutes, seconds and frames).

Had such an analysis of this episode used the raw text dribble data alone, many of the subtleties would have been lost. It would have been impossible to gain insights into the modelling strategies used by the female participant (singing, playing an imagined keyboard). We would also know practically nothing about the 'elimination' strategy that ultimately proved successful. Moreover, the role of the initial, unsuccessful guesses in this strategy could easily have been underestimated.

4. Future Directions

The results of this small-scale pilot experiment suggest that the composite display of dribble, audio and video data has much to offer Logo researchers who wish to investigate the real-world complexities of human-computer interaction. Furthermore, the use of MIDI sequencing software provides a readily-available, multimedia platform for data collection and analysis in these domains.

The analysis environment presented is still in active development. In the near future, it is hoped to increase the amount of data that is dribbled via MIDI. This will include the contents of variables at each stage of a Logo session (Mills, 2001). When referenced against time-stamps and the 'feedback' and 'user input' labels, it is anticipated that these values will provide useful quantitative data sets suitable for importing into tools such as *SPSS* and *Microsoft Excel* (Gay and Mazur, 1993). Another possibility is to import this data into specialist applications that map out users' decision pathways in graphical form (Shaw *et al*, 1997).

5. Acknowledgements

I am extremely grateful to Bob Gorman for his technical advice and encouragement in the preparation of this article.

References

Barab S, Bowdish B, Young M, and Owen S (1996), *Understanding kiosk navigation: using log files to capture hypermedia searches*, Instructional Science **24**, 377-395

Barab S, Fajen B, Kulikowich J and Young M (1996), *Assessing hypermedia navigation through Pathfinder: prospects and limitations*. Journal of Educational Computing Research **15** (3), 185-205

David H (2003), *What can you get from a dribble file?*, McCabe (ed), Contemporary Ergonomics. London: Taylor and Francis, 277-282

Gay G and Mazur J (1993), *The utility of computer tracking tools for user centered design*, Educational Technology **33** (4), 45-59

Goldman-Segal R (1991), *A multimedia research tool for ethnographic investigation*, Harel and Papert (eds.), Constructionism: Research, Reports and Essays: 1985-1990. Ablex, New Jersey, 467-497

Gorman B (2005), Personal email correspondence between Ross Purves and Bob Gorman, 16th May.

Gorman B (2001), Email posting to the comp.lang.logo newsgroup on 25th May. Retrieved 12 June 2005 from http://groups.google.co.uk/group/comp.lang.logo/msg/f616dea1887998c6?dmode=source

Gorman B (2003), Email posting to the LogoForum discussion group on 19th March. Retrieved 12 June 2005 from http://groups.yahoo.com/group/LogoForum/message/7897 (access to this page requires LogoForum membership)

Harvey B (1997) *Computer Science Logo Style Volume 2: Advanced Techniques*. MIT Press, Massachusetts

Hoyles C and Noss R (1992), *A pedagogy for mathematical microworlds*, Educational Studies in Mathematics **23** (1), 31-57

IMA (1983), *MIDI Musical Instrument Digital Interface 1.0*. International MIDI Association, Los Angeles

Kientzle T (1998), *A Programmer's Guide To Sound*. Addison-Wesley, Massachusetts

Mills R (2001), *Analyzing Instructional Software Using a Computer-Tracking System*, Information Technology, Learning and Performance Journal **19** (1), 21-30

Mills G (2002), *MSWLogo Software version 6.5b*. Softronix, Inc, Massachusetts

Purves R (2001), *A Computer Program for Investigating Transformation Geometry Concepts in Musical and Visual Form: From Theory to Realisation*, Unpublished Masters Degree Dissertation, Institute of Education, University of London. Retrieved 12 June 2005 from http://homepage.ntlworld.com/ross.purves/geomuse/RossPurves_MA_Diss.pdf

Purves R (2005), *Exploring conceptual parallels between music and mathematics education using computer tools: a report on a preliminary study*, presentation at the Society for Education, Music and Psychology Research Conference, 16th April

Scherly D, Roux L and Dillenbourg P (2000), *Evaluation of hypermedia in an activity learning environment*. Journal of Computer Assisted Learning **16**, 125-136

Shaw R, Effken J, Fajen B, Garrett S, Morris A (1997), *An ecological approach to the online assessment of problem-solving paths: principles and applications*, Instructional Science **25** (2), 151-166

TTS (2003), *SONAR 2.2 Sequencing Software*. Massachusetts, Twelve-Tone Systems

Windschitl M (2001), *Using simulations in the middle school: does assertiveness of dyad partners influence conceptual change?*, International Journal of Science Education **21** (1), 17-32