

Technical Report No: 2004/15

***Analysis of the synthesizer user interface: cognitive
walkthrough and user tests***

Allan Seago

14th May 2004

***Department of Computing
Faculty of Mathematics and Computing
The Open University
Walton Hall,
Milton Keynes
MK7 6AA
United Kingdom***

<http://computing.open.ac.uk>



Analysis of the synthesizer user interface: cognitive walkthrough and user tests - Technical Report TR2004/15

Allan Seago,
Dept of Computing
The Open University
Milton Keynes MK7 6AA,
a.seago@londonmet.ac.uk

1. Background

The aim of this report is to analyse the user interfaces of two electronic synthesizers and to report the results of user tests conducted on these interfaces.

Work on interface design in the sound synthesis domain here has tended to focus on the development of experimental input devices which capture physical gestures, and which are mapped to synthesis parameters. However, relatively little attention has been given to the analysis of existing audio and music related hardware and software (e.g. electronic synthesizers) from the HCI perspective. In the context of HCI, sound has, for the most part, only been considered as a means of providing warning or monitoring feedback to users of systems in which it otherwise does not play a role [1]. Auditory interfaces have sought to present data, which would normally be presented visually, in aural form, sometimes for the benefit of users with visual impairments [2] [7] [13]. Studies of software/hardware interfaces in which the creation, editing and storage of audio material is the prime focus are scarce, however [8] [15].

Ruffner and Coker's review of synthesizer interface design [15] focused on the control surfaces of four contemporary instruments, and commented on the degree to which they conformed to design principles identified by Williges *et al.* [16] They concluded that the demands placed on the user by the interfaces made them far from ideal for the purpose: noting that, in general 'user interface principles have been, at best haphazardly applied' in the design of the synthesizer interface, the authors also suggested a number of issues that should drive future research in this area. Another more recent study [5] has applied an heuristic evaluation to an electric guitar pre-amplifier interface.

As well as critically examining the control surfaces of the two instruments, the types of controllers and their layout, the study presented in this paper will seek to examine the user/system interaction required to complete three tasks typical of these instruments, firstly, by using the '**cognitive walkthrough**' technique, and secondly, through a number of **user tests**.

The two synthesizers to be examined are the Roland XP50 and the Korg Trinity. The internal sound generation mechanisms of these instruments are broadly the same, using PCM samples as the basic waveform library, but using techniques derived from traditional analogue subtractive synthesis to process them [4].

2. Synthesizer user interfaces

The user interface of the electronic synthesizer has always been inextricably tied to, and expressed in terms of the means of sound production – the synthesis engine. Thus, the analog synthesizers of the 1960s and 1970s provided devices – oscillators, filters and amplifiers - for the control of three distinct perceptual sonic characteristics - pitch, timbre and loudness respectively. However, the increasing complexity of the synthesizer, coinciding as it did with the commercial availability of digital technology, has resulted in a move away from this simple but relatively intuitive interface, in which a relatively low number of parameters were directly accessible on the control surface, towards one which presents the user with a number of editable preset sounds (**patches**). Each of these patches is a complex assemblage of parameters, organised in a hierarchical structure; typically, in order to modify the sound, the user is required to negotiate his/her way around this structure using up/down, right/left arrow keys, the current position within the structure and the value of the parameter being modified being indicated by a LCD screen.

The typical contemporary synthesizer can be operated in a number of modes, two of which are Performance mode, in which the instrument is being played, and during which a number of real time controllers - pitch wheels, foot pedals etc - are available, and Edit mode, in which individual patches can be created or edited. It is the second of these modes which is the focus of this paper.

While individual manufacturers – Korg, Roland, Yamaha etc - have developed their own varying user interface designs, it is possible to discern commonalities between them, to the extent that one can conceive of a ‘generic’ synthesizer interface [14]. Controllers available to the user for editing sound fall into a number of categories:

- **Performance controls** e.g. pitch wheel, joystick. These are for real time timbre and pitch modification, and will not be considered further here.
- **Mode selection buttons.** These allow the working ‘mode’ (Perform, Patch, System etc) of the instrument to be selected.
- **Numeric key pad**, for entry of numbers.
- **Inc/Dec buttons.** These increment/decrement the numeric value of the current parameter being edited, and offer an alternative way of stepping up and down a numeric scale.
- **Rotary dial** for numeric selection.
- **Multifunctional buttons**, whose purpose varies according to the current selected mode.
- **Navigation buttons** - up, down, left and right arrows. These provide the means of navigating the patch structure.

Many contemporary synthesizers make a distinction between a **patch** and a **performance** (not to be confused with performance mode); this is true for both the instruments being examined here. A **performance** is typically made up of a number of **patches** assigned to different parts of the keyboard; in order to select parameters relating to a performance on the XP-50 (e.g. key span etc), the user presses the Performance button on the control surface. To achieve the same thing on the Korg Trinity, the user presses the button labelled Combi. The focus of this paper, however, is the study of the interaction required to modify a patch; thus, for both the cognitive walkthrough and the user tests, both synthesizers were set to **patch** mode. A **patch** on the XP50 equates to a **program** on the Korg Trinity; this disparity in terminology is something that bedevils synthesizer design. For the sake of clarity and consistency, the word **patch** will be used throughout.

3. The control surfaces

We will now look briefly at the control surfaces of the two synthesizers.

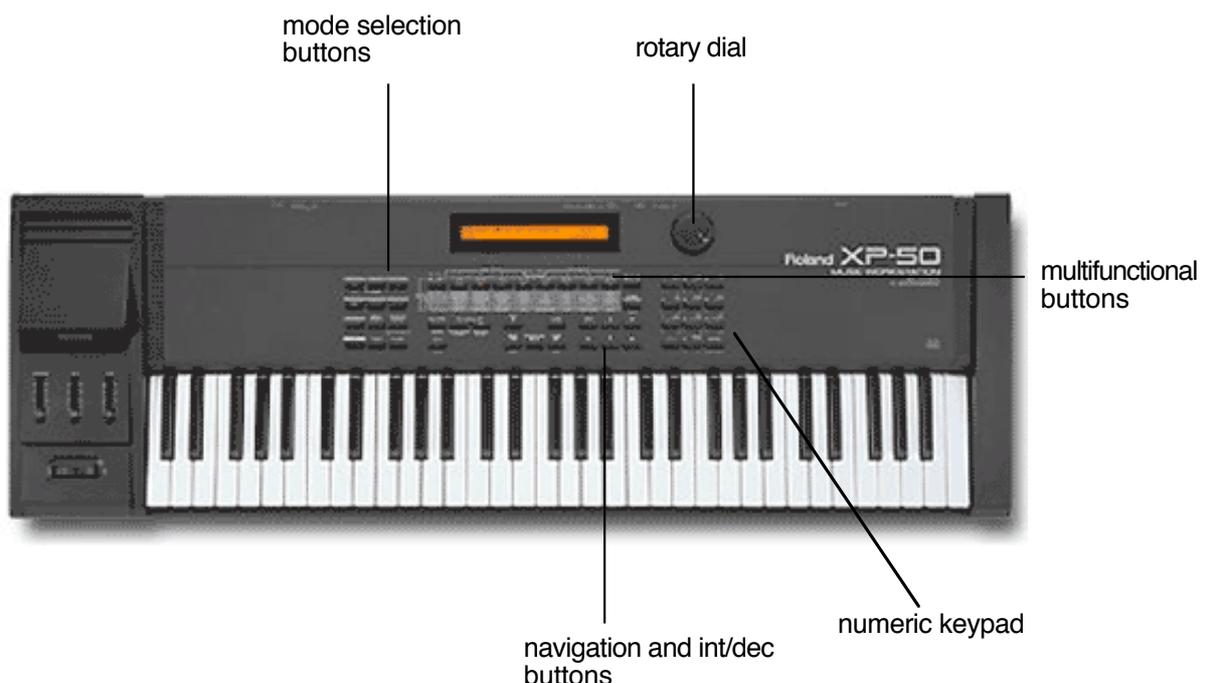


Fig 1: The control surface of the Roland XP-50

Fig 1 shows the control surface of the Roland XP-50. The numeric keypad is laid out on the right hand side of the control surface, similar to a computer keyboard. Mode selection buttons are laid out on the left hand side. In the centre is a row of multifunctional buttons, whose function is indicated by a printed matrix on the panel (see fig 2). So, for example, to access the filter parameters (TVF), the instrument should be put in Patch/Rhythm mode, and the button above the column in which TVF is located pressed.

PERFORM	COMMON	EFFECTS	MIDI	PART	-	-	-	INFO
PATCH/RHYTHM	COMMON	EFFECTS	CONTROL	WAVE	LFO	PITCH	TVF	TVA
SEQUENCER	TRK EDIT	QUANTIZE	NAME	TRACK	LOOP	RPS	SEQ UTILITY	(ERASE)
M.SCOPE	CREATE	ERASE	MOVE	COPY	PLACE	VIEW	(TIE)	(REST)
CHAIN PLAY	INSERT	DELETE	CLR ALL	-	-	-	-	-
SYSTEM	SETUP	CONTRAST	CONTROL	MIDI	SEQUENCER	TUNE	PGM CHNG	INFO

Fig 2: The printed matrix on the control surface of the XP-50. The terms TVF and TVA are circled for clarity.

The central display is a little cluttered - in particular the increment/decrement buttons and the navigation buttons are grouped together, which (as became apparent in the user tests) was a source of confusion.

System feedback, together with the current state of the machines parameters is given via an LCD display.

By contrast, the control surface the Korg Trinity synthesizer (see fig 3) is altogether 'cleaner' in appearance. As before, mode selection buttons are placed on the left hand side, the numeric entry keypad on the right hand side. Next to, and in addition to the increment/decrement buttons, there is also a numeric entry slider..

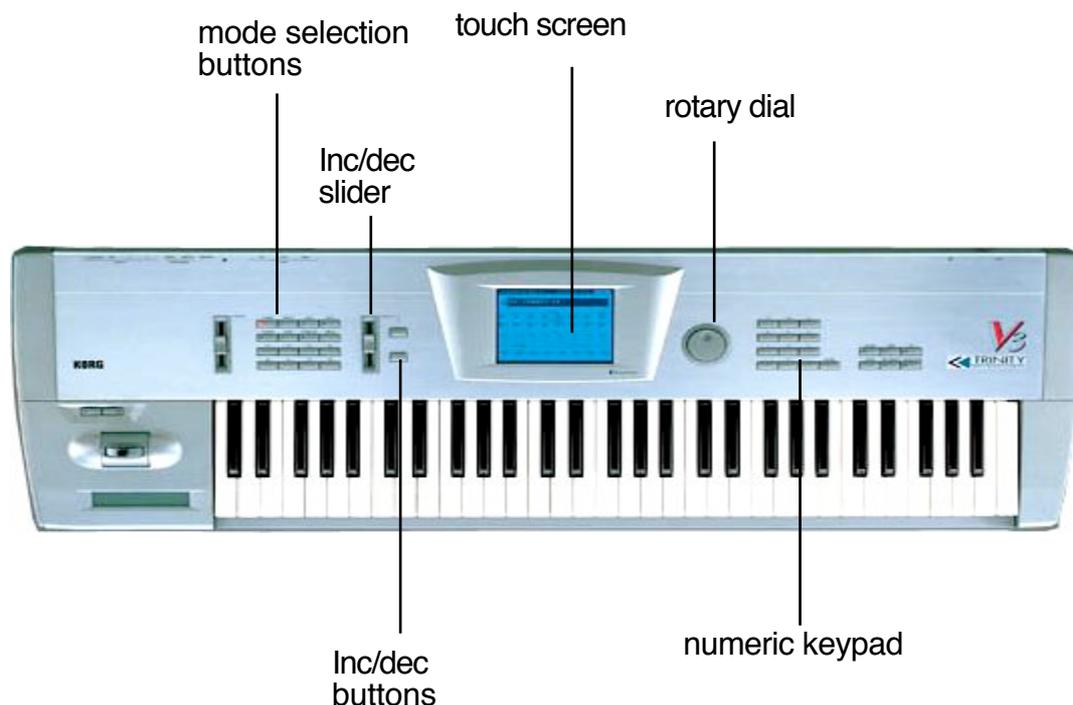


Fig 3: The control surface of the Korg Trinity

The distinctive feature of the interface is the touch screen, which provides a 'point and click' interaction with the system to a user familiar with Windows or MacOS. The screen display consists of a number of pages, each of which provides editable information relating to some aspect of the current patch. Page selection is made via set of hardware buttons located on the left hand side. Pop up menus can be accessed and software sliders/buttons manipulated by touching the screen. It also provides the principal means of feedback to the user.

4. The cognitive walkthrough - background

The cognitive walkthrough (CW) is a ‘formalised way of imagining people’s thoughts and actions when they use an interface for the first time’ [12]. The process is undertaken without users, but nevertheless with a clear idea of who the users are, and the degree of knowledge and experience they will bring to the task to be examined. In contrast to other types of usability evaluation techniques, CW focuses on an ‘imagined’ user’s cognitive activities as he/she operates the interface.

The methodology was employed and assessed in a study of four interface designs for a mail messaging system [11], and concluded that fifty per cent of the problems encountered by users of a given design could be captured using CW .

Both the number and nature of the problems identified by evaluators however can vary with the level of task detail given [9], and the method is not without its critics. Bradford *et al* [3] report on a study which deployed the method in the user interface evaluation of three contrasting software applications – a UNIX interface, a system designed to provide customer and product information to sales representatives, and a CAD tool for electrical engineers. It identified three problem areas – firstly, that it does not always map easily to current software development practice, secondly, that it focuses too much on lower level interface issues, and does not provide a more global view of the interface, and thirdly, that it offers no guidance on designing task scenarios. The last of these problem areas is also highlighted in a subsequent critique [10], which examines the use of the CW method for evaluating the user interface of a multimedia authoring tool. This study concludes that, on the whole, the technique is both learnable and usable, but that there exist a number of areas where further development and refinement of the methodology is needed - firstly, as stated, to provide some guidance on designing task scenarios, and secondly, to enable an assessment of the frequency and severity of usability problems.

On the other hand, some experimental work, reported in Franzke *et al* [6], supports theoretical assumptions underlying the cognitive walkthrough method. The paper notes that new users of display-based (GUI) applications employ an iterative scan-search strategy – first scanning the interface for an appropriate action, and then selecting that action. The degree of success of this strategy is dependent on the prominence of the next correct move in the interaction. Franzke *et al* report that well labelled actions, such as buttons and menu items, are generally favoured in this search, and that users will investigate these before they experiment with the use of unlabelled objects. Limiting the range of actions in the search set can help in narrowing the search. Finally, users tend to be reluctant to extend the search beyond the readily available menus and controls, and will attempt to make use of frequently used interface techniques rather than those less frequently used. Much of this is confirmed by our own findings.

5. Approach

The three tasks to be examined are

1. **The selection of a particular sound, or ‘patch’ from the available library of preset sounds. In this study, the sound was that of a piano.**
2. **The modification of the volume ‘envelope’ of that sound, such that, instead of beginning suddenly and percussively, the sound starts from inaudibility and increases in volume to a maximum over the period of about a second.**
3. **The modification of the ‘tone’ of the sound, making it sound ‘brighter’.**

The first of the three tasks is fairly self explanatory: as is typical of modern synthesizers, both instruments contain banks of preset sounds, indexed and accessed by a two digit number. The second task, however, requires further explanation.

In ‘classic’ subtractive synthesis, the envelope of a sound is characterised by four distinct phases – **Attack**, **Decay**, **Sustain**, and **Release (ASDR)**, as shown in fig 4. The **Attack** phase is defined as the time taken for the sound to reach its peak volume once a key has been pressed, the **Decay** phase is the time taken for the sound to reach the **Sustain** level from its peak, and the **Release** phase the time taken for the sound to move from the **Sustain** level to inaudibility. The **Sustain** parameter is not characterised by its duration (unlike the other three phases), but by its volume level.

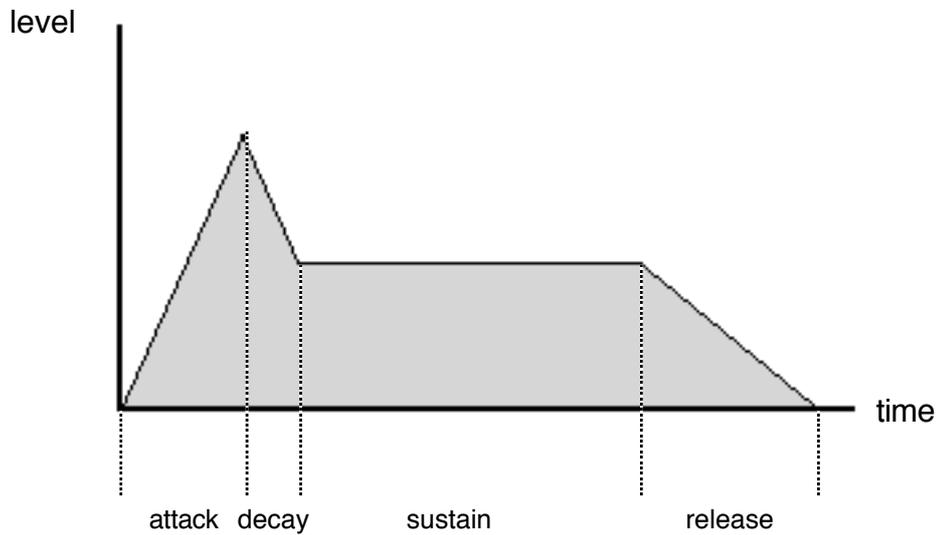


Fig 4: The ADSR envelope of a sound

The third task requires the test user to ‘brighten’ the sound. Sounds which have a ‘bright’ quality to them – e.g. trumpets - are generally characterised by plenty of high amplitude, high frequency components in their spectra. Conversely, sounds which are ‘dark’ or muffled will have less energy in the high frequency range. Again, In ‘classic’ synthesis, a sound can be filtered to boost or attenuate particular frequency components in the spectrum, thus altering the perceived ‘colour’ of the sound. A low pass filter (LPF), for example, is one which attenuates frequency components above a specified frequency, known as the ‘cut off’ frequency (see fig 5) Most of the ‘preset’ sounds in a contemporary synthesizer will have been low pass filtered to a greater or lesser extent; consequently, a typical way of ‘brightening’ such a preset is simply to raise the ‘cut-off’ frequency – this has the effect of increasing the amplitude of high frequency components, which would otherwise be of low amplitude.

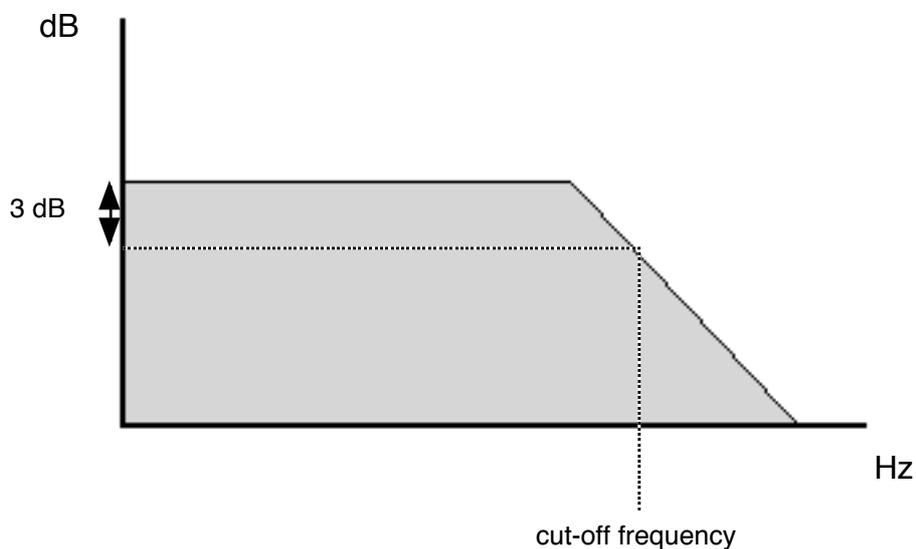


Fig 5: The frequency response of a low pass filter. Frequencies above the cut-off frequency are reduced in amplitude.

Before applying the cognitive walkthrough to the interfaces of the two synthesizers, an important caveat needs to be addressed. The Korg Trinity does offer an apparently simpler way of performing tasks two and three. The main touch screen display provides software controllers for the modification of attack (task two) and filter cut off frequency (task three). However, the degree of control offered here is limited (and, for some patches, does not even appear to work); it could also be argued that these are real time, ‘performance oriented’ controllers. The tasks chosen for this analysis are representative of a much wider range of sound editing tasks that could be performed on the instruments, but which cannot be completed from this touch screen. It was therefore decided to exclude this screen from the study.

In order to complete the walkthrough, a written list of the ‘correct’ actions to be undertaken in order to perform the task with the interface is required. These are presented here as flow diagrams. In line with the methodology of the cognitive walkthrough, four questions were considered for each action completed [12].

- a) **Will users be trying to produce whatever effect the action has? i.e. is the action something that it would occur to the user to do?**
- b) **Will users see the control for the action i.e. notice that it exists?**
- c) **Once users find the control, will they recognise that it produces the effect they want?**
- d) **After the action is taken, will users understand the feedback they get, so they can go on to the next action with confidence?**

The cognitive walkthrough approach requires assumptions about the user to be explicitly stated. Here, it is assumed from the outset that users are not novices in this area of music technology, and that they will be broadly familiar with the terminology and processes of ‘traditional’ subtractive synthesis. (In fact, this is an assumption made by manufacturers of software based synthesizers like *Reaktor* and *Reason*, whose interface is entirely derived from this form of synthesis.)

A detailed account of the CW analysis appears at http://www.city.londonmet.ac.uk/~seago/cognitive_walkthrough.htm

6. User testing

To test this analysis, a series of user tests were run, in which users were asked to perform the tasks described above.

The users were relatively experienced users of synthesizer technologies, who were able to bring to the tests some previous knowledge of similarly configured equipment. Here, eight students of music technology at London Metropolitan University were each asked to perform these three specific tasks with each of the two synthesizers.

At each stage of the interaction, subjects were asked to describe aloud what were thinking - what were they trying to do, what questions and problems presented themselves, what the interface was trying to tell them. The gathering of this data was more difficult than was anticipated; in many cases, the subjects became silently absorbed in the task, and frequent prompts were needed to elicit commentary. This verbal commentary, together with my prompts, were recorded on minidisk for all tests.

In many cases, prompts were also needed for the successful completion of the task. The places in the interaction where this was necessary were noted. In running the tests, it was important to emphasise to the students that the process was about the usability of the interface, and not an assessment of their own abilities and expertise. It is possible that, in some cases, the subjects felt inhibited by this; throughout, it was felt necessary to assure them that they were not doing anything ‘wrong’.

A detailed account of the user testing analysis appears at http://www.city.londonmet.ac.uk/~seago/user_testing.htm.

7. Discussion

The successful negotiation of the interfaces of both these instruments depends to a great extent on, firstly, the domain knowledge that the user brings to the task, and, secondly, the degree of practical experience that the user has of similarly configured instruments. Providing he/she is familiar both with the basic principles of subtractive modular synthesis, and with the processes of navigating the architecture of a ‘typical’ contemporary synthesizer, the user’s overall experience should not present any major surprises (although the XP-50 employs its own acronyms which do not conform with any that the user might be familiar with). By the same token, however, a user without this background will be deterred from tackling anything other than the most basic editing tasks. This was borne out during the user tests; many of the subjects were broadly familiar with concepts like ADSR and filtering, and expected to find these reflected in the interface.

However, the interfaces are not without problems, even for the more technically oriented user.

Firstly, the limited 'real estate' of the control surface makes necessary controls which are multi-functional according to operating mode, and which, being hardware based, are always present even when they have no function in a given mode. Compare this with a software based system, in which such controls can be removed from the display.

Secondly, in neither instrument is the user able to view or edit the sound globally (although a 'wider' view is afforded by the Trinity's touch screen). The view given at any point is that of the smallest branch of the structure, and most of the editing work consists of altering parameter settings at this relatively atomic level, which is time consuming.

Interestingly, most users encountered difficulties only in the early stages of the interaction – interface operation became easier and more fluent once they had navigated through the structure to the point that individual parameters could be edited.

A crucial flaw in the user test methodology was that the two synthesizers were presented to the subjects in the same order (i.e the Roland XP-50 first, followed by the Korg Trinity) – this meant that each subject was able to transfer and apply the experience of working with the first one to the second one. Firstly, this explains to a great extent the relative ease with which they completed the third task on the Trinity, and secondly, may account for (and certainly influenced) the unanimous preference of the subjects for this synthesizer.

However, this in itself was revealing, and suggests that the notion of the 'generic' synthesizer interface, whose appearance and layout is, to some extent, standardised, (regardless of the synthesis method being employed), is one that could be realised.

[1] **Baecker R M, Grudin J, Buxton W A S, Greenberg S** (1995) Readings in human-computer interaction: toward the year 2000 (Morgan Kaufmann: San Francisco)

[2] **Bly S, Frylinger S, Lunney D, Mansur D, Mezrich J & Morrison R** (1985) Communicating with Sound (Proc CH '85, ACM, 115-119)

[3] **Bradford J, Franzke M, Jeffries R, Wharton C**, (1992) Applying cognitive walkthroughs to more complex user interfaces: experiences, issues, and recommendations Proceedings of the SIGCHI conference on human factors in computing systems

[4] **Colbeck, J** (1995) Review of Roland XP50 in *Sound on Sound*

[5] **Fernandes, G & Holmes, C** (2002) Applying HCI to Music-Related Hardware CHI 2002

[6] **Franzke M, Rieman J, Redmiles D** (1995) Usability evaluation with the cognitive walkthrough Conference companion on human factors in computing systems, ACM Press New York, NY, USA

[7] **Grinstein, G & Smith S** (1990) The Perceptualisation of Scientific Data. In Farrell E (ed) Extracting Meaning from Complex Data: Processing, Display, Interaction (Proceedings of the SPIE, Vol 1259, 190-199)

[8] **Hazel, P** (1992) Synthesizers: Interface, Design and Development M.Sc dissertation
<http://www.paulhazel.com/docs/sid.htm>

[9] **Hess D J, Sears A.** (1998) The effect of task description detail on evaluator performance with cognitive walkthroughs CHI 98 conference summary on human factors in computing systems, ACM Press New York, NY, USA

[10] **John B E , Packer H** (1995) Learning and using the cognitive walkthrough method: a case study approach Proceedings of the SIGCHI conference on human factors in computing systems

[11] **Lewis C, Polson P, Wharton C , & John Rieman J** (1990) Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people ACM Press New York, NY, USA

[12] **Lewis, C & Rieman J** (1993) Task-centered user interface design - a practical introduction
ftp.cs.colorado.edu

[13] **Lunney D, Morrison R** (1990) Auditory Presentation of Experimental Data. In Farrell E (ed) *Extracting Meaning from Complex Data: Processing, Display, Interaction* (Proceedings of the SPIE, Vol 1259, 140-146)

[14] **Pressing J** (1992) Synthesizer Performance and Real-Time Techniques Oxford University Press

[15] **Ruffner J. W & Coker G W** (1990) A Comparative Evaluation of the Electronic Keyboard Synthesizer User Interface *Proceedings of the Human Factors Society 34th Annual Meeting* Human Factors Society

[16] **Williges, R C Williges B H & Elkerton J** (1987) Software Interface Design In G Salvendy (Ed) *Handbook of Human Factors* New York: John Wiley & Sons