(inter)facing the music

The history of the Fairlight Computer Musical Instrument

John Twyman

Thesis submitted in partial fulfilment of the requirements for Bachelor of Science (Honours)
Unit for the History and Philosophy of Science
University of Sydney, 2004

Acknowledgements

I must express my deepest thanks to Rachel Ankeny and Jason Grossman. Over the last two years, they have supplied vital academic guidance and shared and sustained my enthusiasm for this project. I hope this work does justice to the depth of their commitment.

Second, I am indebted to all the participants of this study. This thesis would not exist were it not for their willingness to give of their time and knowledge so freely. In particular, I must thank Peter Farleigh, who kindly, and without question, supplied me with the manuals, newsletters, magazine articles and other materials from which I was to learn so much about the Fairlight.

I have learnt too, from my father, who has done so much for this project. He has shared his encyclopaedic musical knowledge, edited countless pages of my work and generally done his best to help me see my work through the eyes of the musicians of whom I have written.

His selfless efforts are just one example, however, of the untiring love and support I have received from all my family, friends and colleagues. Not only have they endured my continued ramblings about my research, they have actively engaged in it, asking questions, offering opinions, and never once dismissing it. They have variously supplied food, drink, transport and innumerable other ‘little’ things when I’ve needed them, provided an escape from the work when it’s all seemed too much, and always believed in my ability to finish. It has been a most humbling experience. Thank you all.

ハイちゃん、
私の事をいつも信じてくれてありがと。

Contents

Figures

Abstract

Since the invention of the Telharmonium in 1906, electronic musical instruments and synthesizers in particular, have played an important, ever-growing role in music production. As this role has developed, so too has a body of literature that examines the nature of their impact. Two main directions can be found within this literature. There exists a large collection of practice-oriented discussions and a smaller number of sociologically grounded narratives that focus upon the relationship between synthesizers and the societies within which one finds their creation and utilisation.

This study is situated in the latter of these two groups, yet draws upon the first. It presents the first history of the Fairlight Computer Music Instrument. An Australian invention, it was a synthesizer that introduced sophisticated sampling and sequencing capabilities to musicians the world over. A detailed examination of its design, and particularly its sequencing tools, reveals how different aspects of the instrument's interface met the different conceptual, procedural and gestural concerns of its many users.

These considerations, and the Fairlight history overall, are contextualised within the heuristic model found in the Social Construction of Technology programme. In so doing, it is shown that there is a need to revise the model to better reflect the nature of the relationships between technology and the social groups that make use of it. Finally, the thesis concludes with a brief examination of some of the possible directions for future research.

A note on style

This thesis examines the interplay between musical and technological knowledge domains. Given the maturity of these domains, it has been impossible to avoid the use of relevant terminology. Every effort has been made to define, in-text, the most important terms, and a glossary is included at the end of the thesis for the benefit of the reader.

Readers will notice the use of both the American *analog* and British *analogue* throughout the thesis. The American spelling has been chosen to make the distinction between different technologies used in synthesizer design and the British used in the broader sense of the word. As the first synthesizer ever made was built in the United States, and Fairlight Instruments itself used the American spelling, this seemed an appropriate choice to make.

In an effort to improve readability, references to certain direct quotes have been removed from the text. Unless otherwise indicated, quotations in the text were taken from interviews conducted with the following individuals: Michael Carlos (3rd September 2004), Bruce Tulloch (3rd September 2004) and Peter Wielk (31st August 2004). Whilst not formally interviewed, Greg Holmes (8th October 2004) and Peter Vogel (27 September 2004) kindly answered my questions via email, and their statements have been afforded the same stylistic convention.

Introduction

For as long as there has been music, there have been musical instruments. They enjoy a rich history, spanning tens of thousands of years and every continent of the globe. Much has been written about them; authors have examined their construction, their aural characteristics, how they are best played and the relationship between instruments and the cultures in which they can be found.

Electronic musical instruments, however, have not enjoyed the same depth or breadth of analysis. With the first electronic instrument, Thaddeus Cahill's Telharmonium, invented only as recently as 1906, this neglect is perhaps unsurprising. Nonetheless, the literature surrounding these instruments has begun to grow and mature, just as they have.

Of all electronic musical instruments, synthesizers are perhaps of greatest interest. Instruments capable of not just electronically producing sound, but also shaping it, they have been employed across a wide range of musical disciplines, practices and cultures. Synthesizers can be found in first- and third-world countries. They can be designed as standalone instruments, yet are nowadays also found on every sound card inside every computer. With such a pervasive and all-encompassing presence, their stories are particularly worth exploring.

Most of the extant literature on synthesizers is written by and for musicians, composers and enthusiasts. Periodicals such as *Keyboard* and *Electronic Musician* regularly contain interviews with synthesizer musicians and inventors, and there is a strong focus on the notions of practice (frequently in the form of tutorials or guides) and consumption (reviews of the 'latest and greatest' gear). One of the more popular monographs on the history of synthesizers comes from the publishers of *Keyboard* magazine (Vail 1993). Far from being a rigorous academic study, this work includes performance techniques from musicians, pricing and production information and tips on finding and buying 'vintage' synthesizers. Nonetheless, Vail's work and music periodicals in general are invaluable. They represent a large part of the transcribed oral history of electronic music technology and are oft-cited primary sources for detailed studies of the synthesizer.

The analytic study of music technologies, known as ‘organology’, can be characterised as the “classification of musical instruments, histories of instrument building, and accounts of the development of playing techniques” (Théberge 1997, 6). Much of this literature deals with traditional acoustic instruments such as the violin or flute, and with ‘classical’ musicology issues such as orchestration. No literature within the canon of organology has been identified that explores electronic instruments such as synthesizers in any significant detail. This is not to suggest that there are no accounts of synthesizers that explore the history of their construction or playing techniques, simply that such accounts are generally not found within *traditional* organology literature.

Pressing (1992) does offer such an account of the synthesizer. His account of its development is comprised of a succession of instruments and their inventors, with little or no effort made to explore any of the contextual aspects surrounding them. Pressing’s central concern is the use of synthesizers for performance, and there is much of value to be found in his study. Much of his book is devoted to exploring the range of possibilities that different gestural interfaces, systems which track physical movement and convert it into (typically musical) commands, offer to the performer. Over the years synthesizers have used wheels, sliders, foot pedals, ribbons, keyboards and more recently embedded software to enable the performer to create and manipulate sound. Pressing essentially provides a detailed ‘how-to’ document for the practice of musical performance, noting the musical potential that these types of interfaces have to offer.

Pressing’s work raises questions about interface design, and how one might evaluate musical instrument interfaces. The field of Human-Computer Interaction (HCI) is diverse and historically has drawn upon a range of disciplines such as software engineering, psychology, cognitive science, ethnography and linguistics (Carroll 2002; Newman and Lamming 1995). Co-opting this existing corpus of knowledge and research in non-musical areas, Wanderley and Orio (2002, 62) “approach the evaluation of input devices for musical expression by drawing parallels to existing [HCI] research.” Where Pressing (1992) offers qualitative analysis of synthesizer interfaces, Wanderley and Orio offer methodologies for quantitative assessment of the capabilities and limitations of synthesizer interfaces.

Whilst organological methodologies and analyses are necessary for any complete understanding of the synthesizer as an instrument, they lack a comprehensive consideration of any social and cultural significance that synthesizers may embody. One might expect these relationships to be explored within an ethnomusicological tradition; however, there is relatively little ethnomusicological literature that focuses on music *technology*. Most ethnomusicological writings have concentrated on the relationship between ethnic or cultural identity and musical *performance*. Furthermore, their subjects have largely been non-Western culture. Recent work by Connell and Gibson (2003) has brought ethnomusicological analysis to bear upon Western musical tradition, but again no specific interest in musical technology can be found in this literature.

One notable instance of ethnomusicological writing related to music technology is an exploration of ‘cassette culture’, the growth of the cassette industry in developing countries such as India and Indonesia (Manuel 1993). Cassette technology “was both an agent of homogeneity and standardisation, but at the same time it was a catalyst for decentralisation, democratisation and the emergence of regional and local musical styles” (Connell and Gibson, 2003, 168). This is a critical theme for the study of music technology and is examined in the more sociologically grounded literature on synthesizers (Théberge 1997, Holmes 2002, Pinch and Trocco 2002).

Théberge (1997, 6) explores this theme extensively as part of his broader concern about “the manner in which popular musicians have become ‘consumers of technology’.” Théberge (1997, 6) is not arguing simply that musicians have become consumers of synthesizers and other musical technology, rather that they have “aligned their musical practices with a kind of behaviour akin to a type of consumer practice.” The synthesizer plays a fundamental role in mediating the relationship between the musician and consumerist practice.

Théberge’s methodology is modelled upon the approach to the ‘sociology of culture’ that Williams (1981) employed within the field of cultural studies. This approach is characterised by its commitment to studying several areas of concern, including amongst others, ‘institutions’, ‘formations’, the social relations of specific ‘means of production’ and ‘forms’ (Théberge 1997, 8). According to Williams and Théberge, it is the relationships between these areas of concern that constitute the primary research focus of the sociology of culture. Théberge investigates the relationships between the musical instrument industry (institution), musicians (formations), the recording studio (means of production) and musical works (forms). He is searching for “patterns of association, apprenticeship, and…changes in musical practice that both are and are not manifest in musical sounds" (Théberge 1997, 10). The synthesizer has become an entity that exemplifies these patterns and that represents these relationships. It is not a "'purely' technical artefact…[but rather] a highly social actor that deserves careful consideration" (Latour 1988, 298). It is exactly this definition of the synthesizer that underscores Théberge's study.

Pinch and Trocco (2002) share this appreciation of the social role of the synthesizer and raise similar issues. They provide a micro-level sociological analysis of the development of voltage-controlled synthesizer technology, particularly the Moog, and to a less extent, the Buchla synthesizer. They show that "technology and cultural practices are deeply intertwined" (Pinch and Trocco 2002, 9). Drawing upon oral histories and existing archives, the result is an accessible, informative micro-level study of synthesizer design, designed for a broad audience, with references to theoretical models from the sociology of technology kept to a minimum.

Pinch and Trocco's account of the Moog and Buchla synthesizers provides the point of departure for this dissertation for two reasons. First, their stories of these two synthesizers echo those of earlier synthesizers, most notably the Hammond organ and the Theremin, and illustrate a key theme that is explored in later chapters: successful musical instruments are characterised by the artful integration of new technology and existing musical standards and practices.

The Theremin, designed by Lev Termin in 1917, and Don Buchla's 1965 Buchla Box, in spite of considerable differences in their technological bases, shared a distinct lack of convention in their interface design. The Buchla Box featured "arrays of touch-sensitive metal pads housed in wooden boxes" for controllers and the Theremin was played without any direct interaction (Pinch and Trocco, 2002, 44). Neither interface was conventional by any musical standard, and the Theremin was particularly unusual. By contrast, both the Hammond organ (1935) and the Moog synthesizer (1965) made use of piano-style keyboards. These were interfaces that were immediately understood by musicians, in sharp contrast with the Buchla Box and the Theremin.

Both the Hammond and Moog instruments were enormous successes and redefined the music industry of their generation (see Vail, 2002 and Pinch and Trocco, 2002 respectively). The

Theremin and Buchla Box, on the other hand, were relative failures, economically and otherwise. Any consideration of these instruments in these terms alone would of course be an incomplete one, yet the inclusion or omission (for whatever reasons) of conventional interface components is a critical concern for anyone who wishes to document and evaluate such instruments.

The second reason for choosing Pinch and Trocco as the best point of departure for this study is the restriction of their efforts to the development of analog synthesizers; no comparable detailed account of the development of digital synthesizers exists. As valve and transistor technology did before it, the introduction of microprocessor technology in the 1970s has changed the face of the music industry. The social technological and musical impact of this technology merits investigation.

This thesis traces the story of the Fairlight synthesizer, the world's first digital synthesizer-sampler. It is a story worth telling for historical reasons; being the first synthesizer of its kind, the Fairlight marks, alongside the Synclavier, the beginning of the microprocessor age in music instrumentation. In addition, the Fairlight was an Australian invention, and documenting the nature of its global impact is something from which this author can gain no small amount of (admittedly parochial) satisfaction. Parochialism notwithstanding, the geographical home of the Fairlight synthesizer is of some interest, in part because of its insignificance in the eyes of the instrument's users, but also in its role in the Fairlight company's eventual demise.

Geographical concerns, however, are not the focus of this study. They are noted in passing only, as the main emphasis is on the instrument's interface design. One of the main objectives in focusing on the interface is to reinforce the proposition articulated above, namely that the integration of new technology and existing musical standards and practices is a crucial aspect of musical interface design. Following an historical account of the Fairlight instrument and company discussed in chapter two, chapter three looks specifically at the software interface of the instrument, identifying the different practical and conceptual understandings of music and computing that it embodied. This aspect of the instrument's interface was its most unique feature as compared to any earlier instruments, and exposed the Fairlight as both musical instrument and computer at the same time. This multiplicity of identity was captured, and made manifest, by the design of the Fairlight's interface and had important consequences for the nature of the relationships between the instrument and its users.

Variation of meaning and its implications are popular topics for current-day sociologists and social historians of technology, and this study provides new case study material for the field. The particular programme it draws upon is the Social Construction of Technology. The last chapter of this thesis expands upon the themes that emerge from the Fairlight study to argue that this theoretical programme requires reassessment and modification. In so doing, it opens up new prospects for sociological investigations of technology that hold great promise for future research.

History

Other than the occasional retrospective article in industry magazines (such as that found in the April 1999 issue of *Sound on Sound*[1]), there are few sources one can turn to for a history of Fairlight. It seems odd that relatively little attention has been paid to the company and its flagship instrument that was so influential.

This is not to suggest that Fairlight was overlooked at the time. Contemporaneous newspaper and magazine articles often represent the best sources of information, although in retrospect they fail to illuminate many of the details of the history, particularly those pertaining to the instrument's construction. In light of the relative paucity of traditional historical sources, Fairlight 'folk lore' too becomes a valuable resource. Contemporary sources of such canonical information within the community include mailing lists and web sites created and used by Fairlight owners and enthusiasts, and the recollections of musicians who used, or worked with other musicians who used, Fairlight instruments. Such community-generated information is a window on the music industry's reaction to Fairlight and is a valuable contribution to any history. Taking these sources into account, an appreciation emerges of the enormous impact that Fairlight made on the global music industry and the lasting nature of its influence.

Kim Ryrie and Peter Vogel became friends while attending the same high school. They shared an interest in electronics and spent time designing and building electronic devices together, some of which garnered considerable public attention. One such device was Mobile Environmental Response Vehicle (MERV), a small robot Vogel designed and built, which earned him an appearance on ABC television as well as published articles in Electronics Today International (ETI) magazine. ETI was part of Modern Publishing, a publishing house owned by Ryrie's father, and this personal connection provided Ryrie and Vogel ample opportunities to publish their early designs. During this period (roughly 1973-1975) they began designing electronic audio equipment, including a hybrid analog-digital synthesizer (a synthesizer that made use of digital controllers to accurately manipulate analog sound-generating components).

[1] Now available online at http://www.soundonsound.com/sos/apr99/articles/fairlight.htm

In 1975 Ryrie and Vogel first met Tony Furse, a fellow electronics enthusiast, through Ryrie's family ties with ETI magazine. Furse had by this time already spent some years designing and building synthesizers, working most notably with Australian composer Don Banks and the Canberra School of Music (of which Banks was the director). His collaborations with Banks had already led to the building of the QASAR II, a duo-phonic hybrid analog-digital synthesizer and by the time Ryrie and Vogel met him, Furse's plans for the QASAR M8 were well underway. Indeed, by the beginning of 1975, Furse had focused his efforts solely on building the prototype model of the M8. Unlike the QASAR II, the M8 was to be an entirely digital instrument, driven by two microcomputer processors. This design concept was to be an enormous departure from all existing synthesizers. The M8 was, in essence, a highly customised computer running specially written software to enable it to function as a musical instrument. Every aspect of the computer had been redesigned to achieve this goal; the internal hardware, interface and software, including its operating system, had all been tailored to make the computer 'heart' of the M8 function solely for the purpose of making music.

Clearly impressed by the possibilities the M8 offered and acknowledging that their own hybrid synthesizer was, in Vogel's words, "not the way to go", Ryrie and Vogel approached Furse in 1976 with a deal to complete the M8 software, re-design the musical keyboard and work on the graphics display, in return for the right to manufacture under license the M8 technology as both a standalone computer and as a synthesizer. Later that same year the Fairlight QASAR M8 became available on the market[2]. Ryrie and Vogel were operating as Fairlight Instruments, named after the hydrofoil operating on Sydney Harbour on which, one day in 1975, a bank manager finally agreed to provide financial backing for Ryrie and Vogel's plans (as reported by Ward, 2000).

[2] Fairlight QASAR M8 Promotional material, Tony Furse archive, 1976, Powerhouse Museum, 96/382/2

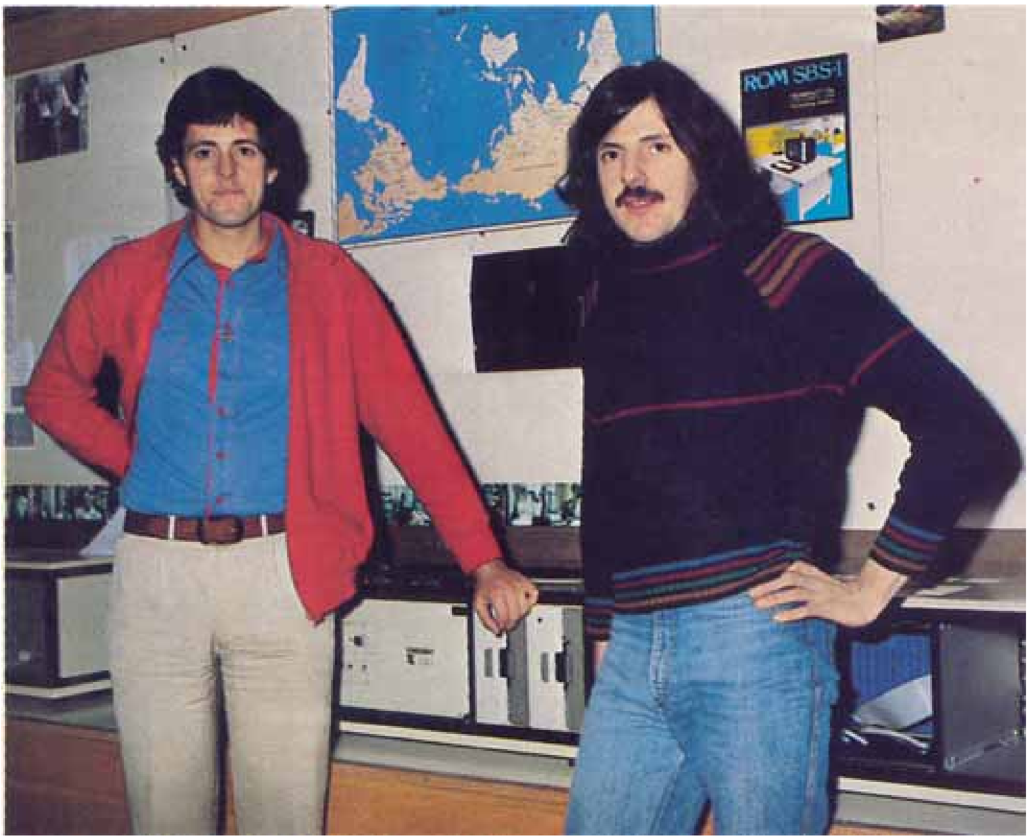

Figure 2.1. Kim Ryrie and Peter Vogel circa 1980. CMI series I assembly cages are in the background. Source: Faithfull, 1980, 82.

Whilst the seeds of the Fairlight synthesizer were to be found in the QASAR, most notably the light pen interface, eight bit eight voice polyphony, the ability to save and load files on floppy disks and the software 'page' architecture (covered in detail in the next chapter), a great deal of work was required to turn it into a commercially viable instrument. Several objectives needed to be achieved, notably simplification of the overall system design, improvements in sound quality and usability and finally a better positioning of the system within musical and commercial contexts. Most of the Fairlight company's efforts during the period of 1976-1978 focused on a comprehensive re-design of the QASAR in order to achieve these objectives. The re-designed system was to be labelled the Computer Music Instrument (CMI).

Vast improvements in both hardware and software design were introduced and it was this work that truly differentiated the M8 from the CMI. The M8 originally had twelve channel cards devoted to producing its eight voice polyphony. Engineers at Fairlight were able to reduce this to a total of eight cards, one for each voice. This improved design reduced the overall cost of the CMI and simplified quality assurance tasks prior to sale because there were fewer major components within the CMI. Most significantly the sampling capability which made the CMI so revolutionary was added in 1978. The CMI's sampling capability was borne out of Vogel's frustration with the overall quality of the sounds generated by the instrument. As part of their attempts to resolve this problem Vogel "decided to sample sounds so that we could Fourier analyse them [to decompose a waveform into similar parts], to help figure out what makes interesting sounds. On a whim I decided to see what would happen if I changed the software to allow the sampled sound to be replayed at a pitch determined by the [musical] keyboard...I captured a fragment of a piano note and when I played it back on the keyboard I was surprised how good it sounded, especially polyphonically." (Vogel quoted in Street, 2000) This was exceptionally significant, as no other instrument could offer this functionality of instant playback and re-pitching of a sampled sound. Only the Mellotron, developed in 1962-1963, came remotely close to the CMI's sampling functionality; when a key was played on the Mellotron, it triggered a reel of tape that was played through tape heads. In this sense samples were possible, but each sample was bound to a particular key of the instrument and could only be played at the pitch it was recorded. Furthermore each sample had to be recorded to tape by devices other than the Mellotron. Loading the reels of tape into a Mellotron was a long and delicate process. The CMI made sampling an easy task, integrated within a single instrument. Sampled sounds could be saved to and loaded from floppy disk, making it possible for users to build sound libraries for later re-use.

As mentioned previously the music keyboard was re-designed, the product of elaborate design and testing within Fairlight. The feel of the keyboard was dramatically improved and the range of octaves increased from four to six. Michael Carlos, early collaborator and eventual employee at Fairlight, recalls that "a lot of thought went into the weighting and the feel of it. A lot of the competitors had keys like electronic organs that had no weight, no inertia. [Unlike the CMI] you couldn't even rest your hand on the keys." Importantly, the CMI was capable of registering the velocity of key strikes, fundamental for capturing the dynamics of note performance. The early CMI music keyboards featured three faders and two switches. These could be used to achieve additional real time musical expression as determined by the user. The faders and buttons could be assigned to different controls, such as vibrato, volume, glissando and sustain.

Perhaps the largest amount of effort during this period went into developing the software. The M8 sorely lacked the mature software needed to present itself as a powerful, comprehensive music synthesizer with an easy to use interface. Whilst the page architecture was extant in the

QASAR, code printouts of early M8 software indicate a system that was designed not with musicians in mind, but with technicians who were modelling and synthesizing music. The pages referred to loudness ‘vectors’ and ‘vector slopes’, ‘pitch registers’, ‘waveform modulus’ and channel run, rate and vector ‘masks’[3]. Almost all of this language was to be rewritten or replaced entirely, reflecting the growing recognition within the company of the importance of presenting the CMI as a musical instrument.

One major aspect of the software redesign centred on the complete integration of the light pen into the majority of the CMI’s functions. The light pen was one of the aspects of the interface that brought it immediate attention within the musical community. It signalled in no uncertain terms that the CMI was something new and vastly different to any synthesizer that had come before it. The light pen could be used as a substitute for the alphanumeric keyboard for many tasks, including navigation between the different pages, and was used for editing sounds within the system. The CMI was capable of displaying a visual representation of a sound’s waveform, which could then be tweaked to the user’s satisfaction. For the first time through the use of the light pen one could now *draw* a sound from scratch.

With the completion of these major changes, Fairlight was ready to bring the CMI to market. With its improved hardware design it was easier to manufacture and maintain, satisfying those within Fairlight such as electrical engineers and programmers. The improved interface, alongside the newly added ability to sample sounds made the CMI attractive to musicians in a way that the M8 could never have been. Finally, thanks to the marketing decision to rename the instrument, the CMI was explicitly presented to the world as a musical instrument, and not a computer.

The release of the CMI, the world’s first digital synthesizer-sampler, occurred in 1979, with what became known as the series I being sold to Stevie Wonder, Kate Bush and Peter Gabriel. A 1982 article in the Financial Times (London) reported that Wonder was Fairlight’s first customer, apparently deciding to purchase a CMI “after hearing a tape of a dog singing a song which was generated by computer programme” (Williams, 1982). Other anecdotes from within the community suggest that Gabriel was the first to buy a CMI. Irrespective of who was first to own one, it was Gabriel’s *Start*, released in May 1980, that is recognised as the

[3] QASAR source code, Tony Furse archive, date unknown (circa 1975), Powerhouse Museum, 96/382/2

first popular song to employ the CMI, making use of a string sample, beating Bush's *Babooshka*, released just one month later, which made use of the CMI's sampling abilities to include the sound of breaking glass.

One of the most immediate standout aspects of the CMI was its reliability. Comprised entirely of digital components, it didn't suffer the same heating and consequent tuning problems that often plagued earlier analog synthesizers. Where instruments like the Moog or Prophet V were renowned equally for their ability to go out of tune once warmed up alongside their ability to produce a range of sounds, the CMI simply worked as required, as expected, all the time.

Figure 2.2. The CMI series I. Source: Faithfull, 1980.

With the CMI now commercially available the hardware for the CMI was not altered significantly for a number of years, at least until the introduction of the series II. The software however, was to be continuously modified. In spite of its commercial positioning and use as a musical instrument, the CMI was still at heart a computer, making it easy to update, fix and improve existing features and indeed to add newer capabilities. It was the introduction of two software pages in particular that was to greatly increase the popularity and use of the CMI within the music industry.

A software update in 1979 saw the introduction of Page C and the Music Composition Language (MCL). Now the CMI became not only the first digital synthesizer-sampler but also the first instrument that also provided sophisticated integrated music composition features. Users could write complex musical pieces for multiple parts which the CMI could play back at a later time. The introduction in 1982 of Page R gave the world the first real-time programmable sequencer. Now musical pieces could be constructed as the CMI was playing them, providing the user with instant feedback regarding compositional decision making. For many Page R was seen as the ‘killer app’ that made the Fairlight such an attractive choice for music composition and performance. Both Page C and Page R are the focus of detailed analysis in the next chapter.

After the introduction of Page C and Page R the next major revisions to the CMI were focused on hardware. In retrospect one might be tempted to think that the hardware specifications of the CMI were incredibly poor, but at the time it was an impressive machine. The series I featured 64kB of system memory (RAM) and with an additional 16kB memory dedicated to each voice or channel (on the eight individual channel cards), it could sample sounds at a maximum of 24kHz, allowing playback across a frequency range of 0kHz to 12kHz. This allowed for sound samples of a quality roughly equivalent to half that of CD quality. This sampling rate was seen as a significant problem for the series I; high fidelity samples, particularly of high frequency sounds such as piccolos or upper-register piano notes, were difficult if not impossible to obtain. The series II, released in 1982, directly addressed this concern. It featured improved channel cards that allowed for a maximum sampling rate of 32kHz, delivering a frequency range of 0kHz to 16kHz, much closer to CD quality.

At the time of the series II release, Fairlight’s annual turnover had reached $2 million Australian and the company employed some 30 workers. Thanks to the early and rapid adoption of the CMI within the music industry, Fairlight was able to expand its operations globally. Matsushita, one of Japan’s largest industrial concerns, approached Fairlight seeking exclusive distribution rights for the CMI in the Japanese domestic market (Williams, 1982). Syco Systems in the United Kingdom (operated by Peter Gabriel’s cousin Steve Payne) was already selling the CMI in the United Kingdom and Fairlight itself established a United States office. This growth was aided by Australian Federal Government export allowances that reduced export costs, something crucial for Fairlight as most of their sales were to overseas customers. Further Federal Government high-tech funding and development schemes saw up to 50 per cent of Fairlight’s development costs reimbursed (Williams, 1982).

Just one year prior, in 1981, major music industry players including Roland, Akai and Sony began developing the MIDI protocol[4]. The protocol specified a standardised mechanism for the transmission of timing and control information between suitably capable instruments, making it possible for computers, synthesizers and drum machines for example to control one another and exchange information. With so many important companies signing the protocol, it was important technologically and commercially, for Fairlight to embrace the protocol. During 1982 and early 1983 Fairlight focused its research and development efforts on introducing MIDI support into the CMI.

Released in 1983, the series IIx maintained backwards compatibility with the series I and II CMIs but introduced MIDI and SMPTE (used for sound work in film and television) support through a specially designed plug-in card. Now the CMI could either control, or be controlled by, other MIDI-capable devices, be they other musical instruments or computers. Other improvements included an upgrade of total system memory (from 64kB to 256kB) and the introduction of the newer 6809 Motorola microprocessors (replacing the original 6800s). The newer CPUs (featuring expanded instruction sets) and increased system memory made possible improvements to the overall performance of the CMI. Faster, more effective communication between components within the instrument was achieved, leading to better response times; users would not have to wait as long for certain types of tasks (such as displaying a sound waveform onscreen) to complete.

1986 was a year of significant change for both the CMI and Fairlight. The series III signified a major redesign of the entire CMI architecture, with the introduction of newer microprocessors (the 68000), a new operating system (OS9), improved 16 bit architecture and hard drive storage the most technologically significant changes. Sixteen voice polyphony was now possible, thanks to redesigned channels cards which were now capable of supporting two voices each, and the maximum sampling rate possible was increased significantly to 50/100kHz (dependent upon whether the sample was stereophonic or monophonic), due to an increase in both system and waveform memory. All of these changes meant greater response times during use, and far greater sound quality for sounds produced and sampled by the CMI. A single sound could now be sampled at multiple pitches. More sample pitches meant the CMI had to do less mathematical transposition when replaying the sample at different pitches,

[4] The first MIDI enabled instrument, the Sequential Prophet-600, was released in December of 1982.

resulting in less distortion. In short, sounds sounded more ‘natural’. With an increase in the instrument’s polyphony, these higher quality sounds could be employed in more sophisticated compositions.

Multi-pitched sample sounds necessarily had greater storage requirements and took longer to load and save but the inclusion of hard drive storage ensured there was no associated increase in wait times between operations. Having files now stored on a hard drive instead of a floppy drive resulted in shorter wait times for file operations. The 80MB hard drive provided significantly more storage space than the earlier floppy drive (totalling 800kB) ever could.

Two aspects of the physical interface were revamped for the series III. First, the light pen was replaced with a graphics tablet integrated with the alphanumeric keyboard. This change occurred for a simple, yet significant reason; users complained of their arms tiring from holding the light pen to the screen for long periods of time. Second, the musical keyboard underwent modification. A common feature on other synthesizer keyboards of the time, such as the extremely popular MiniMoogs, was the inclusion of pitch wheels, spring-loaded wheels that could be used to bend the pitch of a note above or below its original tone. Peter Wielk, at the time the in-house studio manager at Fairlight at that time recalls,

> I think I can honestly say that it was through my insistence that we put pitch wheels on [the keyboard]…And coming from an electronics, mainly, background I just thought that, “Ok, there’s a wheel were you can bend the pitch and another wheel were you can introduce vibrato,” but it was only after working in Los Angeles [prior to working at Fairlight]—I was working on Chick Corea’s MiniMoogs—and he invited me down to see the show in LA afterwards. What had seemed to me to be just boring wheels just turned a normal electronic instruments into something that you can actually get a huge amount of emotion from.

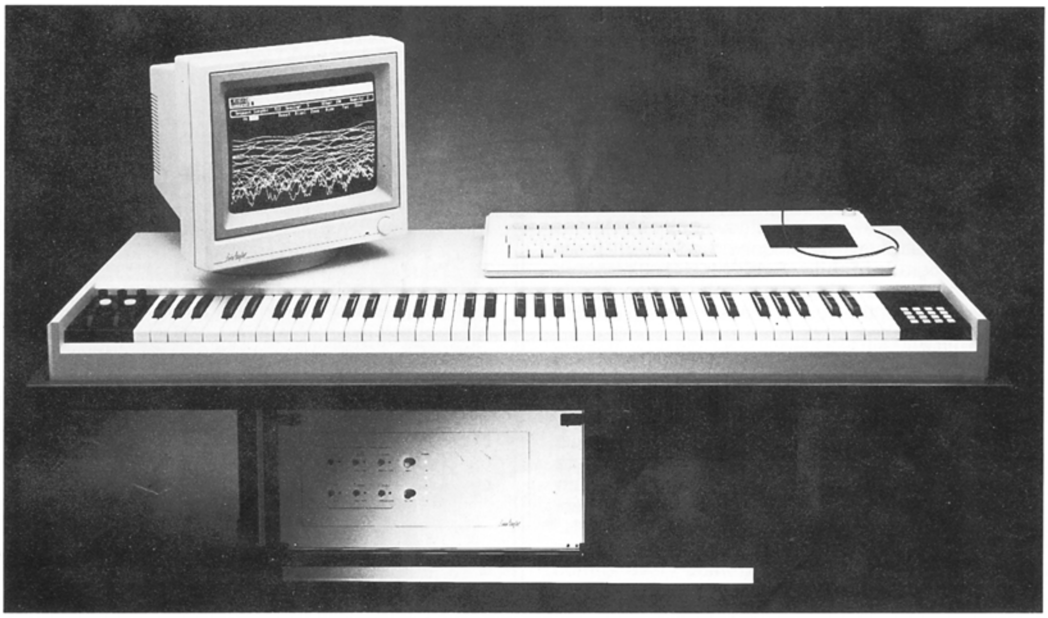

Figure 2.3. CMI series III. Source: Fairlight Instruments, 1985.

All these technical changes took effect during a period of significant growth and change for Fairlight Instruments. Under the Federal Government's Management Investment Company scheme, established in 1983, venture capital companies (MICs) were encouraged to provide venture capital and management expertise to Australian companies to establish a competitive footing in the global marketplace. To qualify for the scheme, a company had to possess at least three of the following features: "use [of] innovative technology; export orientation; international competitiveness; potential for rapid growth, and; potential for creating skilled employment in Australia" (Isaksson and Cornelius, 1998, 13-14). Upon this basis a company would be granted licensing by a statutory Board, with investors receiving a 100 per cent tax concession for the year of investment. By 1986 Fairlight easily qualified and received major investment funds from Advent Management Group, Samic and Delphin Finance. A new management team was appointed by the MICs and the number of employees within the company grew significantly.

With these newly expanded resources, Fairlight looked to diversify its product base. After the release of the series III the company's focus shifted from developing musical instruments to studio recording and audio post-production. The series III was to be the last musical instrument the company would make. Their next major product release was the MFX audio post-production console, MFX meaning "Music and Effects". It was designed for studio and particularly film and television work. Even though it signalled the end of Fairlight's work on musical instruments, it is worth noting that the first release of the MFX was nothing more than a series III CMI with a new interface and specialised software. Indeed it was not until the release of the MFX3plus in 1996 that the last vestiges of CMI architecture were finally retired from Fairlight products.

A large part of the justification for this shift in focus was the growing competition to the CMI in the marketplace. CMIs were not cheap; a fully optioned series III could cost over $100,000 Australian dollars. Wielk feels that another part of the problem was the rise of "Japanese companies using mass production techniques, that didn't sound as good, but they did a lot of what a Fairlight would do but for an absolutely tiny amount of money." By 1986 notable competitors such as Emu Systems and Casio were offering sampling capabilities at a fraction of the CMI's price. Emu Systems' Emulator retailed for $8,000 Australian, and the Casio SK-1, a less sophisticated sampling keyboard, could be purchased in New York for less than $100 US (Pareles, 1986).

The difficulties in maintaining market dominance eventually struck home. Fairlight ran into financial difficulties and finally declared bankruptcy in early 1989. MIC Samic withdrew its funding in 1988 and the Advent Management Group, unwilling to expose itself any further, withdrew a $2 million Australian commitment in late December 1988, leaving Fairlight unable to continue operations. Fairlight management at the time laid the blame with the MICs. Rob Young, appointed by the Advent Management Group to lead Fairlight in 1988, felt at the time that Fairlight's collapse exposed a serious weakness amongst the MICs, namely that they had "very little in the way of management skills…and I think they don't know how to apply it" (Young quoted in Roberts, 1989). Whilst Young's comments should be interpreted against the background of the personal disappointment he no doubt felt, they were echoed by other Fairlight employees. Derek Wilson, Fairlight's international marketing manager during the last four months of 1988, complained that their 1988 worldwide marketing budget of $300,000 Australian was grossly inadequate (Roberts, 1989). With only $15,000 of that money allocated to its US operations, Fairlight was simply unable to address "the cost of marketing and after-sales service in the US, where its customers were. Its technology, in the end, was not critical [to its collapse]" (Duncan, 1989). New England Digital, maker of the competitor Synclavier synthesizer, in contrast spent on the order of $1.5 US million in the same period (Roberts, 1989).

The collapse of Fairlight was as much a blow for the Australian Federal Government as it was for the company itself, given that Fairlight's failure "was one of the biggest by a company funded under the Federal Government's Management and Investment Company scheme" (Dunn, 1989).

Fairlight however, was not to stay insolvent for long. Later that year the company was revived by a management team that focused on completely repositioning the company within the film and television industry, a move which, notwithstanding another bankruptcy filing in 2003, has seen the company continue to prove itself as a dominant force in the digital audio arena. The company's success in this second phase of its operations was recognised in early 2004 with Fairlight employees Andrew Cannon and Michael Carlos receiving a Science and Engineering Award from the Academy of Motion Picture Arts and Science in the United States (AMPAS, 2004).

From the moment of its release the CMI made a significant impact upon the music industry. It revolutionised the way contemporary rock and pop bands of the 1980s approached music production. The list of owners and users of the CMI reads as a veritable 'who's who' of the 1980s popular music scene: Stevie Wonder, Herbie Hancock, Kate Bush, Mike Oldfield, Jean Michel Jarre, Kraftwerk, Grace Jones, Queen, Frankie Goes To Hollywood, Thompson Twins, Human League, Tears for Fears and Peter Gabriel, to name just a few. Significant numbers of albums produced at the time bear the hallmarks of the CMI, be it in the form of particular sounds employed on albums (the CMI shipped with a sound library which became famous in its own right) or the nature and feel of the songs themselves, thanks to the use of the instruments sequencing capabilities, particularly Page R. Former in-house studio manager at Fairlight and studio producer Peter Wielk recalls,

> I'd get invited to do a session in EMI [studios in Sydney] with some top band and they'd just want these orchestral stabs which were just renowned. And I'd sort of say, "Well it does those stabs but it also does thousands of other things", I could reverse those sounds, I could play this, I could do that, I can merge it with something else. And they'd just say, "No, no, no. We just want this sound."

Indeed, so prevalent became the use of the CMI that it prompted Phil Collins to make a Fairlight-free album (*No Jacket Required*, 1985); a distinction he trumpeted in the liner notes[5].

Without a doubt, the CMI stamped its authority on the global music industry. The technological advances that Fairlight introduced with its CMI product range were, and remain, exceptionally significant. Its introduction of audio sampling and sophisticated sequencing technology changed the face of the music industry and users benefited as a result, being able to explore new musical opportunities that this technology afforded them. Its impact is still being felt to this day, as Wielk notes that "some of the sounds that I created there are still in use today and occasionally I listen to a track on the radio and I can hear a sound that either I, Tom Stewart or Warwick [Beauman, all employees at Fairlight] put together." Kylie Minogue even made reference to the CMI on her 2004 album *Body Language*; the love song/'ode' to making music *Sweet Music* containing the lyric, "Slap the high hat in, put the Fairlight on the track" (Minogue, 2004).

The commercial and musical success of the CMI resulted from the new opportunities that it yielded. Yet to bring these opportunities to fruition, users had to negotiate the instrument's interface, which acted as the intermediary between the CMI's technology, such as its sampling function, and its users. Guaranteeing that the design of the interface suited users was a critical concern for those working at Fairlight. In examining this issue, the following chapter looks at the CMI's software interface, and in particular, its sequencing tools, Page C and Page R. It analyses how well the interface matched the expectations, practices and understandings, of its users, and how it represented the CMI as a variety of artefacts.

[5] Which can be viewed online; see (Collins, 2004)

Analysis

The CMI introduced to the world a number of new technologies, most notably sampling and sophisticated pattern-based sequencing, which were to have an enormous and lasting effect upon the music industry, as detailed in the previous chapter. In the hands of musicians, composers and studio producers the CMI created an extraordinary range of new musical possibilities. Yet these new possibilities were not made available solely due to the new technology within the CMI. As was the case with earlier synthesizers, the way in which new technologies are packaged and presented to the user is of utmost importance, affecting not only the rate of adoption and success or failure of an instrument, but also how it is used and in which social contexts. A detailed study of the CMI's interface design will reinforce this point.

The CMI was the first musical instrument to offer sophisticated and integrated composing and sequencing tools. A detailed study of these tools and their design and operation, provides insight into the various understandings of music that informed the overall design of the CMI, and how these understandings evolved during the CMI's product life. The sequencing tools were components of the CMI's software 'page' architecture, and through a careful examination of this architecture an appreciation also emerges of the dual nature of the CMI: it was simultaneously a computer and musical instrument, an issue of significance in the context of interface design and sociological analysis of technology.

This chapter describes the CMI's software architecture before exploring Page 9, Page C and Page R in detail. These pages, the sequencing and compositional tools of the CMI, are of particular interest because they embody ideas about music composition, and not performance alone, as other aspects of the CMI's interface do. Much of the source material has been drawn from the CMI user manuals, particularly those written specifically for the sequencing tools. This information is augmented by oral histories which provide insight into not only the design process for each of these pages but also how they were used by professionals. In drawing upon this material, this chapter also reinforces the proposition made in the first chapter that the artful integration of new technology and existing musical practice is essential for the success of an instrument, and that the site for this integration must always be the interface. This chapter looks beyond existing explorations of musical interface design and analysis (such as Wanderley and Orio, 2002) by paying particular attention to the sequencing tools of the CMI and including the conceptual dimension alongside the gestural dimension in its consideration of musical interface design.

The CMI page architecture

The CMI ran a customised operating system on top of which numerous display pages were layered. The pages grouped various functions and served as the primary interface through which users manipulated and managed the instrument. Display pages were referred to by number or letter. The first release of the CMI contained eleven display pages (Pages 1-9, D and L) but over time, software updates caused new pages, and hence new features, to be added to the system. Developing an understanding of the display pages and their operations was said by Fairlight manuals to be crucial for "gaining a solid grasp of the system's potential, and [the visual interface] can be thought of as a window which one moves in order to view and gain access to different areas of control within the instrument" (Fairlight Instruments, 1982, 10). Whilst the display pages were designed to facilitate the making of music, their design and operation were strong indicators that the CMI was, at heart, a computer.

Page 1 was the index or menu page within the system, listing all of the available pages and a basic description of their function to the user. It was the first screen presented to users when the CMI was switched on. From Page 1, users could use either the alphanumeric keyboard or light pen to access all of the other pages.

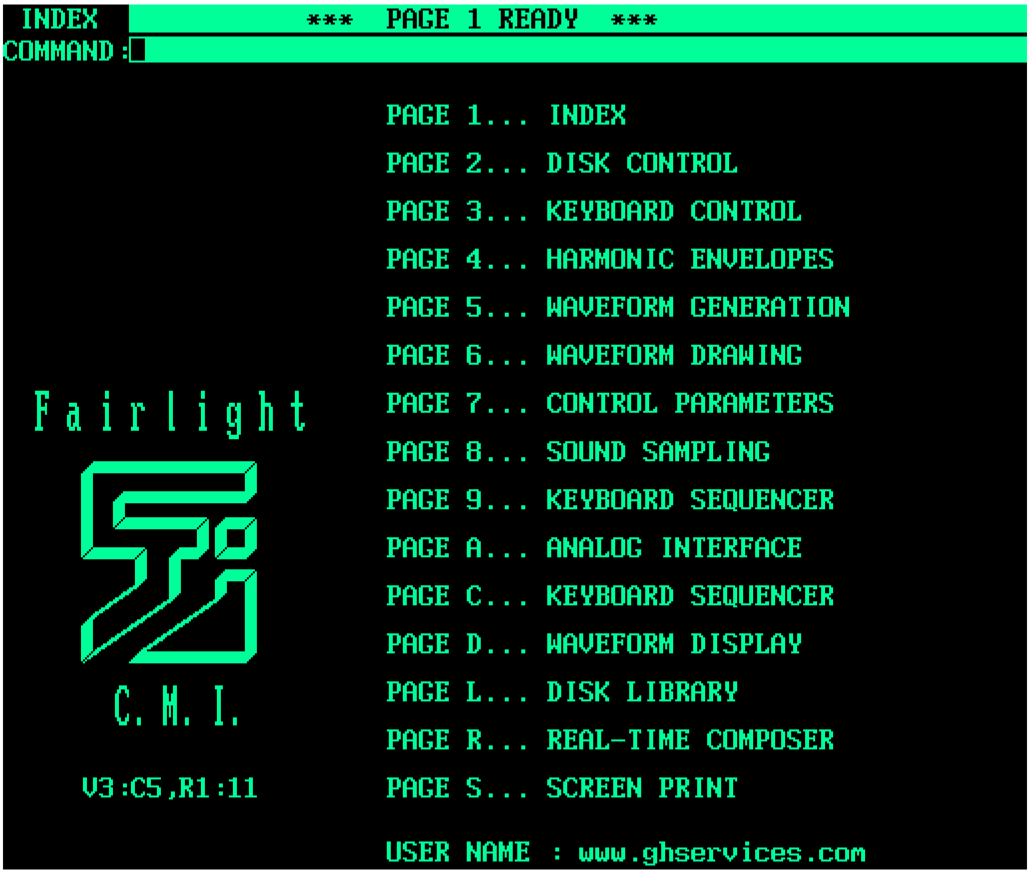

Figure 3.1. Page 1, the first screen presented to users when switching on the CMI. Source: Holmes, 1997a.

Page 2 provided a listing of all files on disk along with commands to load, save, copy, delete, and rename them. Page 2, alongside Page L and Page C, clearly revealed the CMI to be a computer. Its functions were entirely grounded within a computing context, since it focused on file management. Understanding this context was essential, as the functions were responsible for the management of the *musical* content of the CMI, such as sampled and created sounds, instrument configurations (as established via Pages 3 and 7, and discussed below) and music sequences created on Pages 9, C or R.

Page 3 was one of the most important pages for being able to operate the CMI, as it grouped all of the commands used for assigning different sounds (also known as 'voices') to different output channels of the instrument and to different octaves on the musical keyboard. The CMI used the concept of 'registers' to achieve this, with registers defined as "groups of 1 or more of the eight output channels. You [could have] from 1 to 8 registers…provided the total of their channels [did not exceed] 8 (later sixteen with the introduction of the series III)" (Fairlight Instruments, 1982, 31). Registers were also be mapped to different sections of the musical keyboard, with each considered a 'virtual' keyboard, which could be tuned as required; useful, for example, in situations where non-Western tunings were desired. Finally, different sounds could be loaded (via Page 2) and assigned to the active registers. These configurations could also be saved to disk for easy and quick recall.

All of these structures facilitated a very high degree of control over what might be considered to be the musical identity of the CMI. Specifically, what instrument or instruments would the CMI be? For example a single register encompassing all available output channels could be created with a harpsichord sound assigned to it, turning the CMI into an eight- (or sixteen-) note polyphonic harpsichord. A more sophisticated configuration could be used in conjunction with the sequencing tools of the CMI to (re)create an entire contemporary ensemble. For instance, with four single channel registers each assigned different drum sounds (such as bass, snare, hi-hat and ride cymbal), another single channel register assigned a bass guitar sound and the remaining channels allocated to a last register assigned an electric guitar sound, one could very easily create music that sounded like typical three piece rock bands. Figure 3.2 shows a relatively simple configuration where eight sounds have been assigned to eight single channel registers, with two of the registers (B and E) assigned to the main and (optional, secondary) ‘slave’ musical keyboards respectively. Register E has been tuned down two octaves and register F tuned up 25/100ths of a semitone.

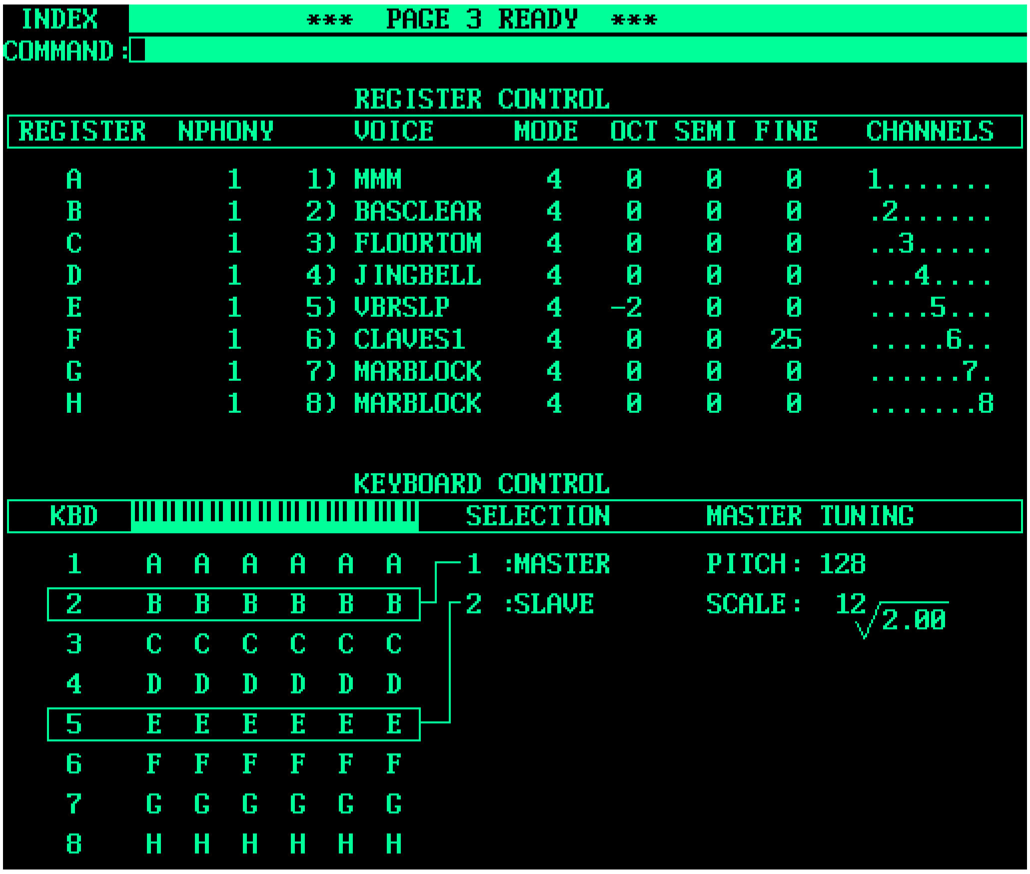

Figure 3.2. A simple Page 3 keyboard configuration, illustrating the assignment of voices to registers, registers to output channels and registers to keyboards. Source: Holmes, 1997b.

Pages 4 and 5 offered "facilities for additive synthesis by harmonic amplitude manipulation" (Fairlight Instruments, 1982, 10). Essentially these pages provided the tools required either to create sounds from scratch (using additive synthesis, the process of adding together pure tones to create complex and rich sounds, as seen in earlier instruments such as the Hammond Organ) or to edit existing sounds by manipulating their harmonic overtones. In many respects these pages were very similar, mostly because they shared the same fundamental representation of sounds as the results of additive synthesis techniques.

Page 6 offered similar functionality to Pages 4 and 5, in facilitating the creation and editing of sounds, but using a novel bit of technology; waveforms could be drawn on screen with the light pen. A sound's waveform would be displayed in a manner very similar to that of an oscilloscope. By using the light pen or the alphanumeric keyboard, points along the waveform could then be redrawn to change the acoustic nature of the sound.

Pages 4, 5 and 6 are striking because they provide the clearest evidence of the CMI's use of acoustical modelling. Whilst concepts and terminology such as additive synthesis, triangle, square and sawtooth waveforms and harmonic envelopes were definitely familiar to anyone who had experience with earlier analog synthesizers, or large-scale computer music systems such as MUSIC-N, these terms were not well-known within the broader musical community. Those working at Fairlight were aware of this, as is clear from the introduction of the CMI user manual which states that,

> coming to terms with all aspects of the system's use may possibly require the re-evaluation of some old habits and concepts. It will certainly require the acquisition of some new ones. Musicians…will need to develop a more absolute knowledge of what [audio modulation] processes mean in terms of actual waveshape and spectral content. (Fairlight Instruments, 1982, 1)

Once again the dual identity of the CMI is explicitly being presented to users. Were musicians or composers new to the CMI learning how to use a new musical instrument, or how to use a computer? The answer is that the CMI was both, as will be discussed in detail in the following chapter.

Page 7 provided functionality similar to Page 3, but focused instead on the setting of "expressive control parameters such as attack, level, vibrato, portamento and so on" (Fairlight Instruments, 1982, 10). These musically expressive devices could be assigned to the various switches and fader controls of the CMI. As with Page 3, these parameter settings could also be saved to disk for quick recall.

The sampling of sounds was achieved through Page 8, which allowed users to set parameters such as the sample rate and activate low- and high-pass filters to block unwanted frequencies from being recorded. Setting the correct sample rate was particularly important with pitched sounds; for example a sound pitched at the note A (110Hz) would only sound in tune on the CMI if the sample rate was set to 14080Hz. A table of notes and corresponding sample rates was supplied in the manual, and Page D, used for displaying waveforms, was designed to help "quickly determine the accuracy of the sample rate" (Fairlight Instruments, 1982, 74).

Many of the display pages were strongly related to each other, functioning in unison to govern over the real-time operations of the CMI, as they were active at all times, irrespective of which page was actually displayed to the user. For instance, Page 2 maintained continuous supervision of all disk activities, ensuring file integrity, preventing users from accidentally overwriting certain files, using duplicate file names and other such monitoring tasks. Page 3 was responsible for the allocation of output channels to the voice registers and the assignment of registers to the music keyboard. It was also a utility page, and performed allocation, routing and tuning functions that applied to the instrument. Pages 4, 5 and 6 formed "the heart of the Fairlight and [provided] three different approaches to waveform creation, modification and control" (Fairlight Instruments, 1982, 11). Page 7 supported additional functions that applied to the actual playing of a waveform. All remaining pages were functionally passive, playing no part in the CMI's real-time operations unless actually displayed.

With the exception of Page C, all of the design pages featured common interface elements. Figures 3.1 and 3.2 demonstrate that visual elements, such as the page title and the command line, were consistent across different pages. Certain commands issued on the alphanumeric keyboard, most notably the 'load page' and 'help' commands, worked irrespective of which page was currently being displayed. Furthermore, within individual pages most commands could be executed either by typing them in, or by making use of the light pen and activating corresponding visual triggers. In short, the CMI was designed to offer users multiple ways to carry out any particular task, and to provide a consistent framework within which those tasks could be understood.

This consistency of design and multiplicity of task completion paths is significant for a number of reasons. Much user interface research has focused on the different demands of novice and expert users and how those demands might be met in a single interface (such as Schneiderman, 1998; Padilla, 2003). What is particularly interesting about the CMI is the impact that its dual identity plays on these concepts: there is a conspicuous blurring of the distinction between the concepts of novice and expert users. Typically, users as seen as *either* expert or novice, but CMI users were frequently experts *and* novices; more than proficient musically, but unskilled when it came to the use of computers. Striking the right balance between 'CMI as computer' and 'CMI as musical instrument' was critical; multiple ways of performing the same task was one way of achieving that balance, particularly where the musical keyboard was involved. This was an interface component with which all musicians were familiar. Where musical interfaces could not be employed, as in the case of the software pages, consistency of design minimised the complexity and subsequent learning curve that

CMI users faced. These issues were perhaps most important for the CMI’s sequencing tools, which presented the CMI less as a computer musical instrument, and more as a computer music compositional tool.

The CMI and sequencing

The first sequencing tool developed for the CMI was Page 9. It provided the user with the ability to record the operations of the Fairlight, capturing both the sounds produced and the settings of the faders and switches on the machine. Multiple overdubs could be achieved through the simultaneous playback of one recording, and the recording of that playback along with what the user played on the musical keyboard. For musicians, Page 9 was an easy system to understand, as it was strongly analogous to the multi-track recorders that were already popular. Having such facilities embedded within the CMI offered greater flexibility than multi-track recorders alone: performances could be played back at any speed without a change in pitch and could be saved to disk and replayed at a later time, allowing users to assign different sounds or voices to the playback. Furthermore, the sequences could be shared between CMIs.

However Page 9 frustrated users for two reasons. First, they were unable to edit recorded sequences, and second, Page 9 forced them to be creatively bound by their ability to play the musical keyboard. Greg Holmes, who provided composition and performance services for bands such as Rush and various television productions, felt that “Page 9 was not that good, or even useful to me. It was nothing more than a ‘virtual tape recorder’, and once the notes were played (into the recorder), they were lost in the ‘black box’ and could not be individually adjusted.” Michael Carlos, who was a critical player in the CMI’s design, echoed this sentiment when describing the feedback from composers who used the first series I CMIs:

> We had an immediate demand from people that said, “I'm a composer and I don't want to play anything…I just want to write it down. Just give me some way to feed it in.” We had people who were coming from old SYSTEM V programs on mainframes and they'd say, “This is my life as a computer musician I just want to feed in the input.”

Fairlight was to respond to these demands in 1979 with the release of Page C and Music Composition Language (MCL).

Page C and MCL

By the time Carlos became involved with Fairlight, he had already established himself as an accomplished composer and musical performer. He had been a member of the Australian band Tully, composed the soundtrack to the film *Storm Boy* and had been appointed musical director of the groundbreaking 1972 Australian stage production of *Jesus Christ Superstar*. He was also one of the first users of the Moog synthesizer in Australia. Initially working as a consultant to Fairlight after being approached by Ryrie and Vogel in 1975, Carlos quickly taught himself programming. After first designing Page L, used to build a catalogue of files stored on up to 80 floppy disks, during the period 1978-1979 he tackled what he saw as the compositional shortcomings of Page 9:

> [Page 9] was wonderful in itself but what I needed was to be able to have some music that I couldn't actually play and then the editor to correct and quantise it and do something meaningful with it. It occurred to me that I could take a text file and parse it and build the same output file that Page 9 produced, then feed it back in as playback material and have Page 9 play it.

The results of Carlos' efforts were, in his words, the "prototype of the [MCL] concept [something that] Peter [Vogel] very much took over." Vogel recalls working on a means by which he could "program the CMI to play tunes as I could not play keyboard." He vaguely recollects that Carlos took this work he had done and developed Page C from it. Where their accounts do coincide is that once a prototype existed, they worked together on developing it into MCL, and that the language's syntax was something that evolved and matured as they went. In particular it appears quite clear that initially there was little higher order organisation to the language. This higher order structure was to be developed in time for MCL and Page C's release in 1979.

MCL was a tree-structured hierarchical language. Every composition consisted of three levels of information. The top level corresponded to the entire piece of music, the middle level to the parts that constituted the piece, and the lower level to the sequences that made up each part. Saving Page C compositions resulted in the 'piece', 'part' and 'sequence' files respectively.

The overall design of MCL was intended to resonate with a composer's overall general conceptualisation of music. A 'piece' "behaves like a 'conductor', instructing which parts are required to play” (Fairlight Instruments, 1983, 1). ‘Parts’ can be thought of as musicians. ‘Sequences’ are the individual pieces of music to be played. It is possible for multiple parts to play the same sequence, which is analogous to various instruments in an orchestra all playing the same part. A part can also play a series of sequences. This is akin to the musician who performs one song, then plays another, or the musician who plays the first movement of a composition then the second. A maximum of nine parts can be active in any single piece. A maximum of eight parts can be assigned to keyboards (or registers) as per the sound-keyboard assignment functionality of Page 3, and a single part can be used for controlling the use of the CMI’s faders and switches (as per Page 7).

Using Page C is very similar to entering programming code. Users enter lines of MCL code that contain various instructions. As is the case when programming code, users can add comments to aid anyone who might look at the code later. When creating or editing a composer file, line numbers are automatically inserted, which are ignored by the system when processing composer files, but are very useful for users. They provide a means of reference for review of code, and also are used by the system to notify users of probable locations of errors in the code. When users have finished entering data, the result is saved as the relevant file (sequence, part or piece).

Sequence, Part and Piece Files

The details of the syntax and interplay of sequence, part and piece files are especially interesting, as they point both to the computer nature of the CMI and the particular understanding of music that shaped Page C’s design. They embody the balance struck in meeting the concerns of both the computing and music composition knowledge domains, and the degree of natural affinity that these two domains share.

Any line in a sequence file can contain default values for the instructions that are to follow. Defaults are entirely optional but extremely useful, and can appear anywhere in a line. Typical parameters given default values include key signature information, octave selection and key velocity. Other than defaults, lines contain instructions to be played, which are predominantly notes and related parameters, but also can be commands for the various controls and switches usually accessed via Page 7.

A very simple example of what a sequence file might look like:

```
0001  * MARY HAD A LITTLE LAMB
0002  O=3 !#F#C#G C4 B A B C4 C4 C4,2 B B B,2 C4 E4 E4,2
0003  C4 B A B C4 C4 C4 C4 B B C4 B A,4
```

The first section is the automatically generated line number. The first line is simply a comment, ignored by the CMI when the file is played. The next line begins with the setting of the default octave for all of the notes to the third octave. All notes up until the next setting of an octave default will play in this octave unless explicitly stated. The key signature of A major (with the three sharps F#, C# and G#) is specified next. All Fs, Cs and Gs thus will be sharp unless explicitly stated. The rest of this line instructs the Fairlight which notes to play. The numbers immediately following the Cs and Es specify that they should be played in octave 4 (higher). The comma and number following it dictate that the note should be played that many times longer than the other notes.

In addition to specifying the octave and duration of notes, MCL has parameters to specify accidentals (sharps, flats and natural notes), key velocity, gap (the time between the end of one note and the start of the next), hold (the inverse of gap) and transposition (Fairlight Instruments, 1983, 7-9). Within MCL, a chord is a sequence of individual notes that are to be played simultaneously. Page C treats chords identically to individual notes in the manner in which it assigns parameter values such as velocity, gap or hold.

Repetition of instruction sequences is possible through the use of triangular brackets. Anything inside the brackets will be repeated as many times as specified (by a number immediately following the brackets). Nested repeats can also be used:

`<<A B C>2 D>2` expands to `A B C A B C D A B C A B C D`

Some settings do not always need to be assigned explicitly. The default values for octave, transposition, velocity and beat can be modified using relative specification. Relative specification allows for the addition, multiplication, division or subtraction of some number to the current value of a parameter, which is particularly useful to change the speed of several sequence files which have differing beat values. The use of a relative ratio (be it multiplication or division) ensures that the sequence files will remain in time.

Users are not forced to type in all of the notes comprising a sequence file. After the issuing of a particular command, it is possible to input notes from the music keyboard. Information such as key velocity and note duration is ignored in the input; the notes themselves are the only thing recorded from the keyboard. By default, each note is recorded with the specific octave attached to it. Alternatively a base octave can be specified when enabling music keyboard input so that the notes are recorded with relative octave specifications. Black keys (accidentals) can be set to be either recorded as sharps or flats as desired. The manual specification of accidentals is required because MCL does not assume anything about the nature of the composition. If there is a transposition from one key to another in a composition, it is always possible that notes that were previously sharps could become flats.

Part files contain lists of sequence files to be played sequentially by that part. Part files also can contain keyboard assignments (as per Page 3) and sequence commands themselves. The following example (adapted from Fairlight Instruments, 1983, 16) illustrates why these are useful:

```
0001 !K=1         Play the following on Keyboard 1.
0002 SEQ1         The sequence file SEQ1.SS is played on Keyboard 1.
0003 "R,4         Rest four beats before executing the next command.
0004 !K=3         Play the following on Keyboard 3.
0005 SEQ2         The sequence file SEQ2.SS is played on Keyboard 3.
```

Piece files contain lists of part files (and sequence files if desired) which are to be played simultaneously. If the specified part files do not contain keyboard assignment information, the piece file will automatically assign the parts to the keyboards in the order of their listing in the file. For example:

```
0001 PART1.PT     Play PART1 on Keyboard 1.
0002 PART2.PT     Play PART2 on Keyboard 2.
0003 PART3.PT     Play PART3 on Keyboard 3.
0004 SEQ5.SS      Play the sequence SEQ5 on Keyboard 4.
```

A more complex illustration of the interplay between sequence, part and piece files can be found in Appendix 1.

In addition to the commands particular to sequence, part and piece files, MCL contains a number of commands that are used to perform tasks such as the creating, saving, loading and copying of files within the MCL system. Other commands also provide some particularly powerful functionality to the user, the most notable of which is the COM (compile) command.

This command is used to convert MCL part or sequence files into Page 9 sequencer files so that they can be replayed or merged in Page 9. This device is particularly useful when one remembers that MCL is intended for non-interactive playback, whereas Page 9 is used for real-time interactive performance. A musician might use MCL to create intricate rhythm sections for a composition, export them to a Page 9 sequence file, and then use Page 9's overdubbing capabilities to record an improvised melodic line over the top.

Playing and debugging MCL files

After entering in all of the desired sequence, part and piece files, Page C can then be used to non-interactively playback the piece. It is at this *run-time* stage that Page C detects problems in the MCL files. It classifies problems as either *warnings* or *errors*, depending on their severity. Warnings will not stop the music from playing; the CMI will cope as best it can. Warning messages are typically issued when parameter settings hit their limits, such as an octave specification outside of the allowable range, a gap or hold value of either zero, or longer than the note duration, or a Page 7 control value exceeding the acceptable range. When an out-of-range condition is met, the value played will be forced to the limiting value.

Error messages result from "part or piece files requesting files that do not exist or are not loaded, syntax errors in the files, and any characters in a file which Page C does not recognise" (Fairlight Instruments, 1983, 34). When an error is detected, the part or sequence concerned will stop playing.

All messages (warning or error) are accompanied by the name of the file in which the problem was encountered. The offending line and its line number are presented to the user, and an upward arrow is displayed underneath the first character in the line which the MCL debugger has considered responsible for the error. These messages are not the only aids available to users for finding and resolving problems. The HALT and STEP commands are particularly useful. The HALT command can be used to freeze a performance and allow the user to inspect the MCL code at that particular point; the STEP command forces the CMI to process the MCL file a single step at a time.

To anyone acquainted with computer programming, there are several features of MCL that should seem very familiar. The process of playing and debugging MCL compositions is identical to the process of executing and debugging computer programs. The issuing of warning and error codes at run-time finds its analogue in the computing sphere when a program is being compiled, and the HALT and STEP commands are perhaps the two most fundamental commands used in debugging any computer program that contains errors.

The gestural vocabulary, the set of motor skills associated with a particular practice, required to use Page C and MCL is almost entirely based upon extant models of human-computer interaction. The alphanumeric keyboard is the primary means of interacting with Page C. Moreover when the music keyboard is employed to input sequences of notes, not all of the information that can be conveyed by the actions of the performer is in fact recorded. Specifically, key velocity is ignored by MCL, and users need to edit the entered notes later to set the key velocity value (either per note or through the use of default values). Motor skills associated with computing or typing thus are valued over motor skills that are associated with musical performance. When John Lewis, a classically trained musician and composer working in London, was asked shortly after his purchase of a CMI what his early impressions were of the instrument, he noted “I wasn’t very *computer literate* when I got the Fairlight” (Lewis quoted in Hammond, 1983, 157; emphasis added). His take on MCL clearly emphasises the importance of possessing the right gestural vocabulary:

> I think the language on the Fairlight is very straightforward although it seems intimidating at first. I had never done any computer work and there seemed to be so many codes, so many different control letters, but in fact I learned them in a very short time and I use them without even thinking now...*I'm grateful now I learned to type properly at one point in my life*. (Lewis quoted in Hammond, 1983, 158; emphasis added)

The hierarchy of the language, particularly as manifested by part and piece files, is strikingly similar to the function-based modularity present in any mature programming language. For example BASIC features subroutines, and Pascal and C both utilise functions. The CMI’s code, sequence and part files are just as reusable as a BASIC subroutine. This code reusability has equivalence in many musical forms. Musical compositions such as rounds, canons and fugues are all defined by their repetition, through multi-part restatement of a main theme. Similarly the MCL capacity for repeated sections of and within a note sequence, comparable to control structures such as the ‘for’ and ‘do…until’ loops one finds in programming languages, exists to represent a feature extant in music compositional practice.

These repeated sections, in conjunction with the setting of default parameter values and their relative modification, closely resemble procedural and syntactical techniques typical of mature programming languages. It is very common for programmers to use a control structure such as a ‘for’ loop to iterate through a range of values for a set of variables. One might perform something similar in MCL to affect aspects of a musical piece such as the volume or vibrato of a particular instrument. If, for example, the volume of a piano sound has been assigned to Control Switch 1 (via Page 7), the command

```
C1=0  < R C1=+1 >127
```

instructs MCL to gradually increase the piano’s volume from zero (i.e. silent) to 127 (the loudest volume possible) one beat at a time (the ‘R’ representing a single beat rest).

The important thing to note here is that the similarities are not just syntactical, but also procedural and conceptual. Given that MCL is itself a programming language (albeit a domain-specific one), it seems unsurprising that such similarities should exist. What emerges from such a comparison is a broad relationship between musical scoring and programming code to complete defined tasks. As Bruce Tulloch, a software engineer at Fairlight who was tasked with rewriting Page C and MCL for the release of the series IIx, said, “when you boil it down, a lot of music’s algorithmic.”

In its particulars however, MCL represents a fundamental departure from earlier computer music languages such as the MUSIC-N series (beginning with the original MUSIC-I created by Max Mathews in 1957), and descendants such as Csound[6], CMusic[7] and Max/MSP[8]. These earlier languages are far more analogous to the analog, voltage-controlled synthesizers that preceded the CMI. They are characterised by their emphasis on the definition and production of *sound* rather than *music* (see Mathews, 1969). Typically a user of the MUSIC-N language family would begin by defining sound objects (or instruments), through the specification of particular combinations of waveform generators (such as sine, square and sawtooth waveforms), pitch frequency, oscillators and envelope generators. Only after such definitions were completed would users begin to concentrate on the task of composition. By contrast,

[6] http://www.csounds.com/
[7] http://www.crca.ucsd.edu/cmusic/
[8] http://www.cycling74.com/

MCL never concerned itself with the creation of instruments which was a task for which Pages 4, 5, 6 and 8 were designed.

More importantly MCL allowed users to create compositions by specifying *musical notes*, something not possible in the earlier languages. MCL's syntax reinforced many compositional ideas, demanding that composers, as in traditional composition, explicitly state note pitch, duration and timbre. Earlier computer music languages instead employed a vocabulary grounded in the domains of computing, sound synthesis and processing, with terms like 'signal generators', 'oscillators' and 'opcodes' the norm rather than the exception (see Berg, 1979 and Vercoe, 2004, for examples of how different these languages could be).

Page R

Page C, and MCL's syntax in particular, clearly provided a sequencing tool that strongly resonated with traditional understandings of composition, centred as it was around the model of composer, performers and pieces. Its interface, however, really only suited very particular groups of users. Carlos identified these users as "people who knew what they wanted. They could write it down in notation." Tulloch echoed Carlos' view, recalling that, "MCL was never very widely used. It was used very deeply by some composers who liked the way the language worked…but it was a radical departure from the way many musicians operated." Page C was so great a departure for the CMI's largest user base that it was ultimately retired with the release of the series III. What was required was an alternative sequencing tool that would better harmonise with the gestural vocabularies and work practices of a broader cross-section of musicians.

The immediate motivation for Page R, however, was not addressing these shortcomings of Page C's interface. As Carlos recalls,

> The initial inspiration came from Kim Ryrie when Kim and I went to New York late in the seventies and saw the Linn drum [machine]. I remember Kim turning to me and saying, "Why can't we do that?" and I was thinking, "Well we've only got eight bit samples." What was blowing me away was not that he was programming it, because I'd done that on the [Roland] micro-composer, but the enormity and beauty of the sound. Kim was saying, "No I don't mean the sound, we'll solve that eventually, it's the being able to program it and build a pattern like that." I said, "Yeah we can do that." And it kind of went from there.

The standout features of the Linn drum machine were its built-in sampled sounds, programmability and, in particular, its use of quantisation (the automatic altering of note timing and duration to exactly fit the beat, seen equally as a good or bad thing depending on one's musical tastes). Ryrie and Carlos aimed to develop a similar tool for the CMI. Page C had already demonstrated that the CMI could be programmed with a very high degree of control over timing. This, in conjunction with the CMI's ability to sample any sound and its polyphony, could deliver considerable musical opportunities unachievable with the Linn.

To develop Page R, Carlos set out with "a discrete set of [musical and compositional ideas]…I mean R stood for rhythm…We want quantisation, it's not to be seen as a bad thing, it's what we want, and we want to have this kind of iterative, additive way of altering the music that it's playing." What he developed was a system that let users build rhythmical *patterns* and in this respect it was, at least initially, just like the Linn.

The Page R operating manual defined 'pattern' as the term "often used in modern music to describe what all (or most) of the instruments are playing at a given point in a song" (Carlos and Stewart, 1983, 1). In the context of Page R this concept was understood as a one-bar score for eight (later sixteen) monophonic keyboards, consisting of individual sequences of musical notes. When a pattern was executed, all of the sequences were simultaneously played, each performed by the associated voice (coordinated via Page 3).

By making the pattern-building process iterative and additive, Carlos guaranteed that users did not need to be virtuoso keyboard performers to construct complicated rhythm patterns. He ensured that Page R could be instructed to repeatedly play a pattern so, for example, users "don't actually have to be able to play that tricky hi hat pattern, we can just put in one tick every time it goes around and gradually build it up." This capability was to prove exceptionally popular with users.

With these basic foundations in place, over the course of a year or so, Carlos and his colleagues at Fairlight began to look beyond rhythm to see how they could extend Page R's functionality. At this stage, Page R only supported a single pitch per sequence; multiple sequences would have to be used if multiple pitches for a single instrument type were desired. The next iteration of the software addressed this by recording note information beyond simple note duration (i.e. rhythm); key velocity and most notably pitch were now registered. These changes were to be the major additions to Page R that would make it so vastly popular amongst users the world over. It could now be used to construct multi-pitched, complex patterns across multiple instruments, something musicians and producers the world over took great advantage of. Trevor Horn, for example, was to use Page R in producing hits for Grace Jones (*Slave to the Rhythm*), Frankie Goes To Hollywood (*Relax*) and the Pet Shop Boys (*Left To My Own Devices*).

These were not, however, to be the last significant additions to the software. What was required was a method for ordering these patterns so that they could be combined to construct complete musical compositions. Carlos extended the pattern idea and created a 'song list', "basically a list of pattern numbers in the order in which they are to be played" (Carlos and Stewart, 1983, 5). It consisted of sequential steps, with each step containing two pieces of information: the number of the pattern, and how many times it was to be played. He also created 'sections', similar to the song list but eight steps in length, useful for "breaking a large composition down (e.g. verse/chorus/middle 8) for convenience in editing and recording…and to generally simplify the structure of the song" (Carlos and Stewart, 1983, 7). Carlos saw the need for such a system because

> you needed to start with a sketch. What I learnt from composing, you can't just sit down and decide to write something for an 80 piece orchestra—bar one, beat one, two flats, first note, etc.—you just can't do that. You need some way to input information at a sketchier level.

Page R allowed users to define a song's overall structure alongside (or even ahead of) composing the particular melodies or chord changes. In this way, Page R was aligned with many musicians' approaches to composition; a considerable achievement.

To present these two aspects of Page R's operations, its interface was designed around 'pattern editor' and 'song editor' modes. Users would switch between the two modes depending on which compositional task they wished to undertake. Figure 3.3 illustrates what a typical pattern editor session might look like:

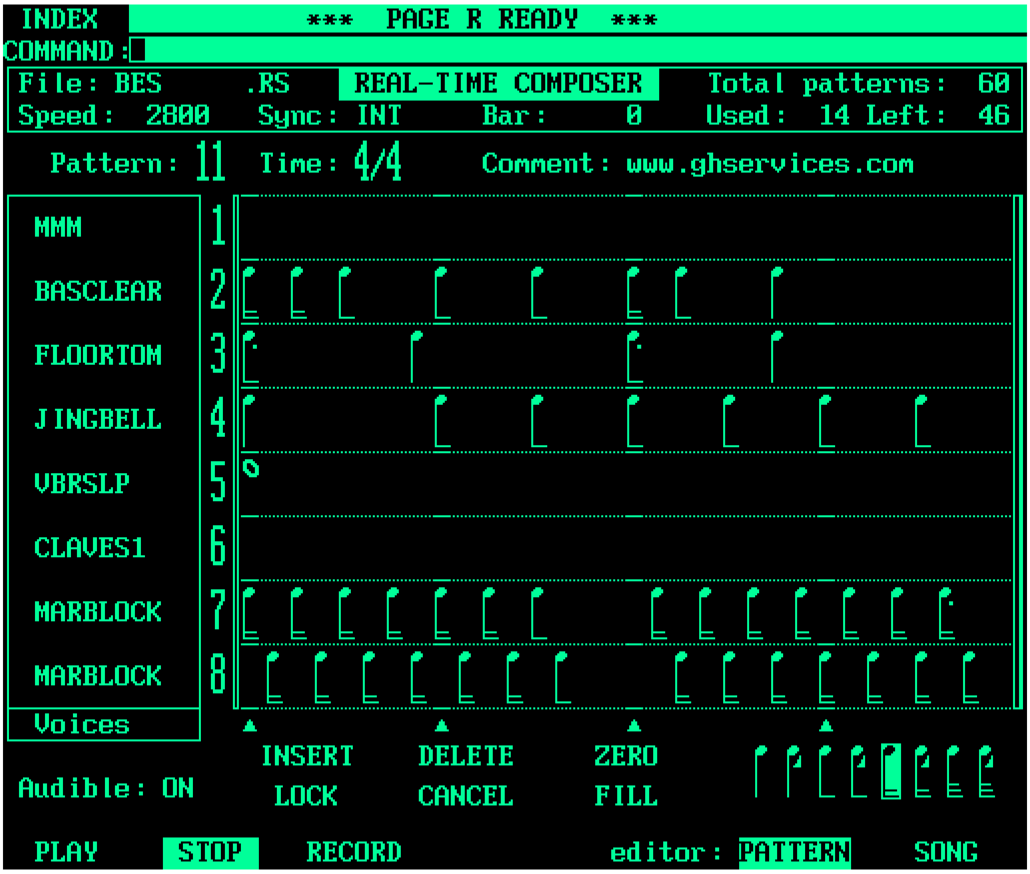

Figure 3.3. A typical session in Page R's pattern editor mode. Source: Holmes, 1996

Creating patterns of notes could be achieved in a variety of ways. Perhaps the easiest and quickest way was to make use of the musical keyboard. After selecting a particular sequence (with either the light pen or alphanumeric keyboard), a command could be issued to switch Page R into recording mode. All notes played on the musical keyboard would be recorded, overwriting any existing notes. When Page R was instructed to repeatedly loop the pattern whilst recording, the construction of complex patterns was possible with the musical keyboard alone.

The alternate means for creating patterns was to step through a sequence and enter notes one by one. Once a sequence was selected, the user controlled a cursor that could be placed at the particular point within the sequence where he wished to insert a note. Inserting notes could be achieved by issuing the 'insert' command from the alphanumeric keyboard, selecting the insert button onscreen with the light pen or by using the musical keyboard after entering recording mode.

This particular mode of pattern creation highlighted the quantisation that Page R performed. The timing resolution of a pattern dictated where notes could be put into each sequence. The resolution itself was determined by two factors: most immediately, timing resolution was dependent upon the note duration selected prior to input. The list of notes in the bottom right-hand corner of the screen would affect the resolution of a sequence, for the simple reason that twice as many quavers could be entered in the same time duration as crotchets, twice as many semi quavers as quavers, and so on. Second, time resolution was "also related to time signature in a more general sense" (Carlos and Stewart, 1983, 23). The range of notes available to the user was dependent upon the time signature. If the number of beats per bar was less than, or equal to half the beat value (e.g. 2/4), the notes ranged from quavers to hemidemisemiquavers; more than half or equal to the beat value (e.g. 3/4 or 4/4), crotchets to demisemiquavers and finally, greater than the beat value (e.g. 5/4), minims to semiquavers. Thus certain timing tricks were sometimes necessary to achieve particular musical outcomes. For example, to have a note play over two patterns, a single 8/4 bar would be required in place of two 4/4 bars (Carlos and Stewart, 1983, 23).

If entering notes one by one via the alphanumeric keyboard or the light pen, note-attribute values were taken from the note-editing table, which 'hid' behind the voice information in the left-most panel (as shown in Figure 3.4). Accessing the note-editing table required pointing "the light pen at any voice name in the left-most window or [pressing] the left arrow key" on the alphanumeric keyboard (Carlos and Stewart, 1983, 25).

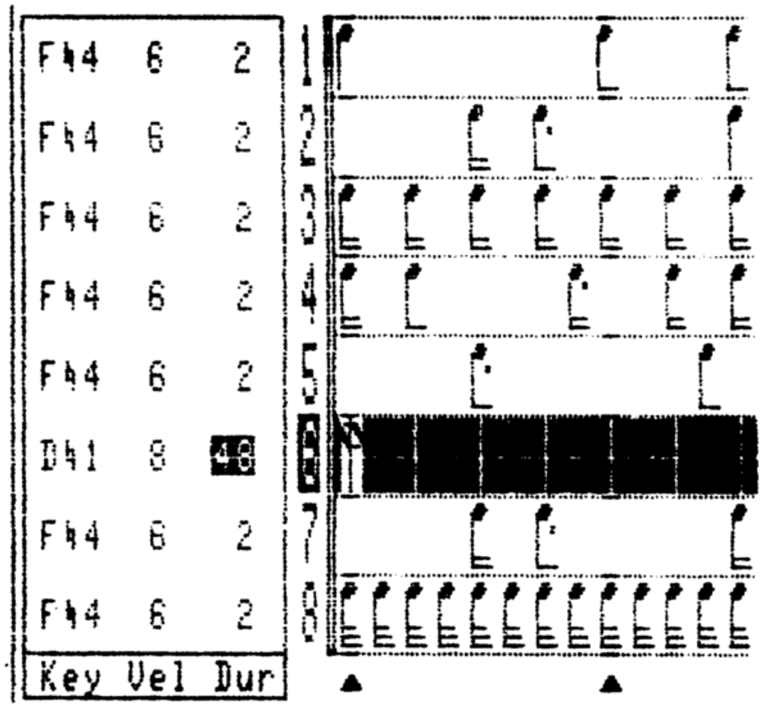

Figure 3.4. Page R with the note editing table active. Source: Carlos and Stewart, 1983, 25

With a single note in a sequence highlighted, such as the first note in sequence six in the figure above, its pitch, key velocity and note duration could be modified.

The combination of all input methods often was the most efficient way of creating patterns. The music keyboard could be used to create a ‘first take’ of a pattern and then the step-by-step approach could be employed to edit individual notes to achieve the pattern desired. This approach accommodated musicians and non-musicians alike, and was suggested in the Page R operation manual (Carlos and Stewart, 1983, 59). Wielk, a self-described ‘non-musician’, found that

> for example I would try and play a bass line and make a few mistakes and whereas a musician might just go in there and play it again I just go in there and correct it with a light pen. So for a non-musician it was incredibly quick to get basic bits of music happening.

The pattern editing mode of Page R also contained a number of very powerful commands for creating and editing large blocks of patterns. Sequences could be copied and pasted between patterns. For example, a snare, hi-hat and kick drum pattern could be copied from one bar to a hundred bars with a single command. Melodies spanning multiple patterns could be transposed in a similar fashion.

The most immediate feature of the pattern editor interface was its employment of musical notation. Further, given the vertical arrangement of the sequences (in this case eight, though this increased to sixteen in the series III) and their associated voices, the interface strongly resembled a traditional musical score. This resemblance was of considerable import. Page R's pattern editor interface made sense visually and conceptually to anyone who approached the CMI with even the slightest musical background. This is in sharp contrast with Page C's interface, which offered little reminiscent of music visually, and only aligned itself conceptually with those who knew what they wanted prior to approaching the instrument. Of Page R, Wielk noted that

> it's very logical. You have horizontal lines with the instrument on the left hand side then the melodic line there [running horizontally across] and then going down the page you'd have all these different instruments and it's exactly like an orchestral score except that there's no note information actually there it's just rhythm.

In addition, the pattern editor recognised the gestural vocabulary of the user. Whilst designed to accommodate a wide variety of users, as Wielk's self-identification as a 'non-musician' indicates, its interface definitely rewarded use of the musical keyboard. Whether it was the creation of an entire musical pattern or the note-by-note editing afterwards, Page R recorded every attribute of every note. The way a user played the keyboard mattered, unlike in Page C. Thus, in the eyes of its users, Page R was a compositional tool embedded within a musical instrument, whereas Page C was a compositional tool embedded within a computer.

Once enough patterns had been created, users could switch to Page R's song editor mode to actually build a composition. Song editor mode was comprised of two windows of information, the song list and the sequence list (as shown in Figure 3.5).

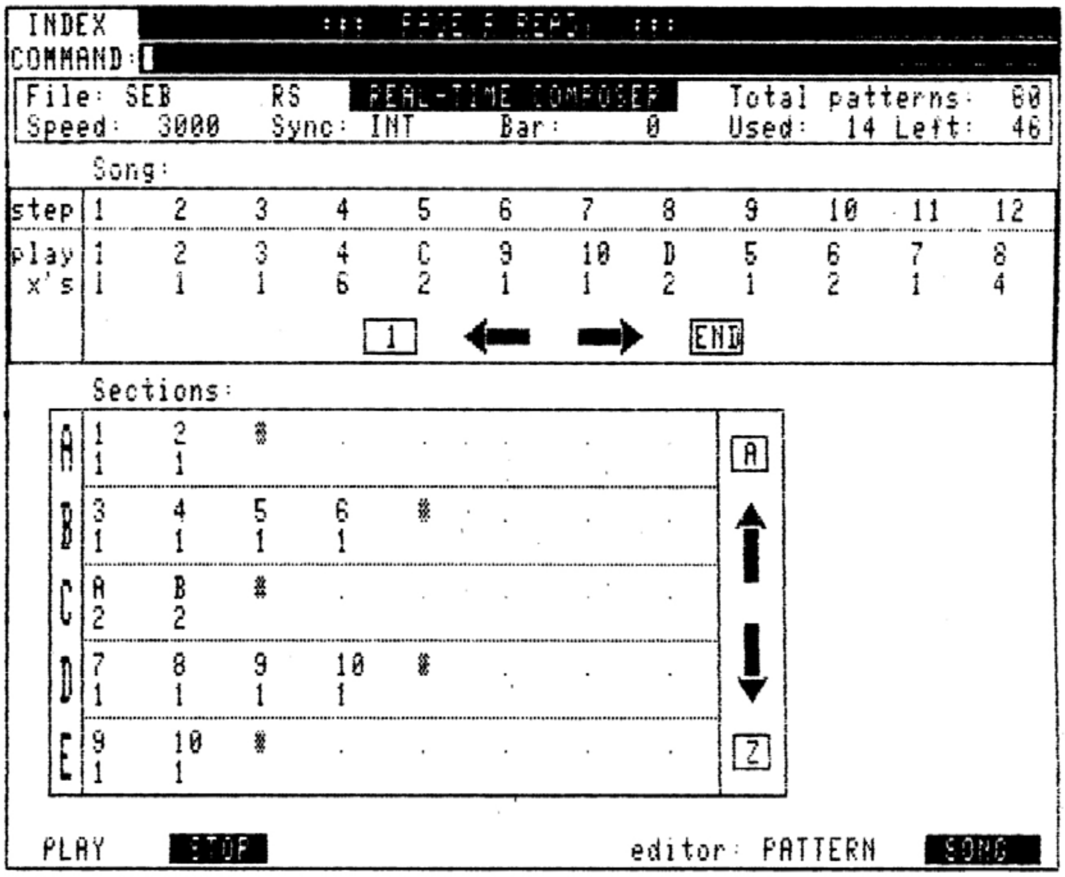

Figure 3.5. A typical session in Page R's song editor mode. Source: Carlos and Stewart, 1983, 43

The song list window showed the sequence of steps and the number of times each step was to be played. Although it only displayed twelve steps at once, a maximum of 255 steps were possible per song. Beneath the song list window was the section window, which displayed five of a maximum of 26 allowable sections (lettered A-Z).

Entering step information was identical for both the song list and sections. At each step a pattern number or section letter was entered, followed by the number of times it was to be played. The only constraints on this process were the inability for a section to include itself in one of its steps, and the restriction of values between one and 127 or infinity for the number of times a step was to be played (Carlos and Stewart, 1983, 46).

The song editor contained block commands similar to those in the pattern editor. Blocks of steps could be inserted, overwritten, copied and deleted with ease. There was also a command which would create a section from a block of up to eight steps starting from the cursor position, be it in the song list or another section. Once created, the original block of steps would be deleted and replaced with a single step referring to the newly created section.

It is worth considering how well this modular, or block, approach to composition suited musicians. More specifically, it is important to consider which styles of composition it most benefited. The key to exploring this lies in appreciating the importance of the creation, and repetition, of musical ideas.

Repetition exists across almost all musical genres. It is a powerful musical device, used to fix the musical material in the minds of listeners. The ordering and repetition of contrasting sections of music can be used to develop tension and release, a concept fundamental to the creation of most artworks. Repetition plays a significant role in classical music forms, such as the sonata (with its exposition, development and recapitulation), and in particular the minuet and trio. It is also the most significant feature of popular music. Old sea shanties, Negro spirituals and contemporary pop and rock music all make heavy use of repetition through chord and verse structures, riffs, hooks and overall song structures such as binary and ternary form. Given this, and Page R's decomposition of music into patterns and their ordering and repetition, it really is no surprise that it was embraced by the popular music industry during the 1980s.

In this respect Page R and C differ greatly. Page R simply could not satisfy classical and orchestral music composers, although it did work for popular musicians. When the suggestion was put to Carlos that Page R encompassed an approach to composition that suited contemporary or popular music much better than it did classical, he responded, "Yes…your point about contemporary versus classical [music]; in the generic sense that's quite right." Similarly, when asked about Page R's suitability for typical classical compositions, Wielk replied, "I wouldn't be able to successfully try and reproduce a wonderful swirling orchestral score with…Page R because it wasn't up to it." Taken on its own, given Wielk's lack of any formal musical training, his statement would not seem definitive on the matter. However, not a single reference could be found indicating the use of Page R for classical composition. Further, when asked if he was aware of any classical composers who made use of Page R, Carlos' response was emphatically in the negative. Page C and MCL appears to have been the favoured option for classical composers, such as the Australian Martin Wesley-Smith.

By focusing on Page C and Page R, it has been shown that looking beyond typical framings of musical interface design is not only possible, but in the case of the CMI, necessary. Through consideration of these sequencing tools, interface design can be seen as not just a matter of satisfying particular performance concerns (such as a performer’s gestural vocabulary), but also as the realisation of different conceptualisations of music, and in particular, of the process of composition.

At its simplest, this distinction could perhaps be summed up, as Carlos puts it, as: “if this is how you want to work you need one of these; if that’s how you want to work you need one of those.” But such a treatment overlooks the character of the distinction between the two tools and the related differences between the users who favoured one over the other. Through its detailed examination of the two sequencing tools, this chapter has shown how differences in the two tools’ design reveal the variation of the CMI’s identity alongside the diversity in the conceptual and gestural priorities of its designers and users.

That Page C was eventually removed from the CMI is of enormous significance. The focus on the details and differences between Page C and Page R reveals why it happened; it failed to adequately satisfy the extant conceptual, procedural and gestural concerns of the CMI’s largest user base, musicians. There remains, however, a need for an overarching framework within which events like Page C’s removal can be adequately described and understood; a model, within which, the relationships between the various social groups, gestural vocabularies, conceptual and procedural differences, and interpretations of the CMI, can be made explicit. It is to this purpose that the next and final chapter commits itself.

Discussion

Up to this point in this study, many ideas about the identity and meaning of the CMI, the relationships between the developers and musicians and the important role that the musical interface plays in establishing these meanings and relations have been explored. On the surface it appears somewhat difficult to imagine that a single, consolidated framework could be used to unify and structure all of these different aspects of this study. This final chapter outlines such a heuristic framework, taken from the Social Construction of Technology (SCOT) programme. After an initial application of this framework to the materials outlined in the case study on the Fairlight, its suitability is evaluated, leading to modifications in the SCOT model. The benefits of expanding the model are made visible via reference to the CMI study. Finally, the chapter concludes with a discussion of some of the opportunities that exist for further study of the CMI.

The Social Construction of Technology

The SCOT programme originated in the Strong Programme of Sociology of Scientific Knowledge (SSK). In proposing his Strong Programme, David Bloor (1976) posited a programme of inquiry that held to four basic tenets—causality, symmetry, reflexivity and impartiality—and greatly influenced subsequent research programmes within the broader SSK tradition.

Harry Collins employed the ideas of Bloor's programme to great acclaim in his (ongoing) research into the field of physics concerned with the detection of gravity waves (see Harry Collins 1975, 1981, 1985). A controversial science, it lent itself remarkably well to Collins' research methodology. Where the meaning of results and the suitability and performance of instrumentation were fiercely contested, Collins constructed a narrative that demonstrated the inherently social nature of the negotiation, and where appropriate, the closure of these problems. Collins' approach has been termed the Empirical Programme of Relativism (EPOR) and is generally considered to be distinct from other similar approaches in SSK due to its "focus upon the empirical study of contemporary scientific developments and the study, in particular, of scientific controversies" (Pinch and Bijker, 1984, 409).

Two of the observations that EPOR has made about science provide the bases on which SCOT is founded: the interpretive flexibility of findings, and the existence of social 'closure' mechanisms of controversy. Proponents of SCOT consider technology with reference to these features through a conceptual model that considers technological development as an alternation of variation and stabilisation.

Variation is manifested in both technological artefacts and the meanings that are attached to such artefacts by the various social groups to which they are most pertinent. SCOT proponents argue, in spite of popular stories of development, that there is no 'best way' to produce a particular artefact. Variation of an artefact itself is unsurprising. Pinch and Bijker's study of the early development of the bicycle neatly illustrates this point; numerous variations on bicycle design existed, "quite different from each other and equally…serious rivals" (1984, 411). Such variety was necessary because of the variety of meanings that were applied to the bicycle by different social groups, which is why the notion of one 'best way' is ultimately a fruitless one.

Variation exists because SCOT defines the relationship between technology and relevant social groups in terms of problem solving. A social group will have a number of key problems that it wishes to resolve, and its relationship with technology frequently offers methods of resolution. Consequently there exists not just a variation of artefact design but also a variety of meanings that are applied to an artefact. It is in this fashion that we find the translation of interpretive flexibility from EPOR to SCOT studies. Technology can provide multiple solutions to multiple problems. How each relevant social group utilises technology to solve its problems requires stabilisation.

Stabilisation as articulated in EPOR involves the types of social 'closure' mechanisms. In SCOT, these mechanisms are understood to operate in two ways: through rhetorical closure and problem redefinition. Rhetorical closure provides stability of technology and its meaning through ensuring that the relevant social groups see their problems as being solved, whether or not the technology in question actually does solve said problems. This mechanism is distinct from closure by redefinition which translates the meaning of technology in order to resituate it as the solution to a problem other than the one with which it is already associated.

SCOT and the CMI

The SCOT programme is well suited as a theoretical framework for the CMI study. Several social groups had well-defined relationships with the CMI and each other (through the instrument), and the developmental history of the instrument lends itself to a model that aims to explore how technology is constructed.

Through the case study, three main social groups have been identified. First, there is the group of individuals who comprised the electrical engineers and software programmers who designed and built the CMI. Second, there are the composers who were particularly interested in Page C and MCL. Third, there are the performing musicians and studio producers who used the CMI for writing and performing popular music (Figure 4.1). It is possible to break down these groups further, for instance by separating hardware designers from software designers, but such distinctions are not of particular benefit for this study.

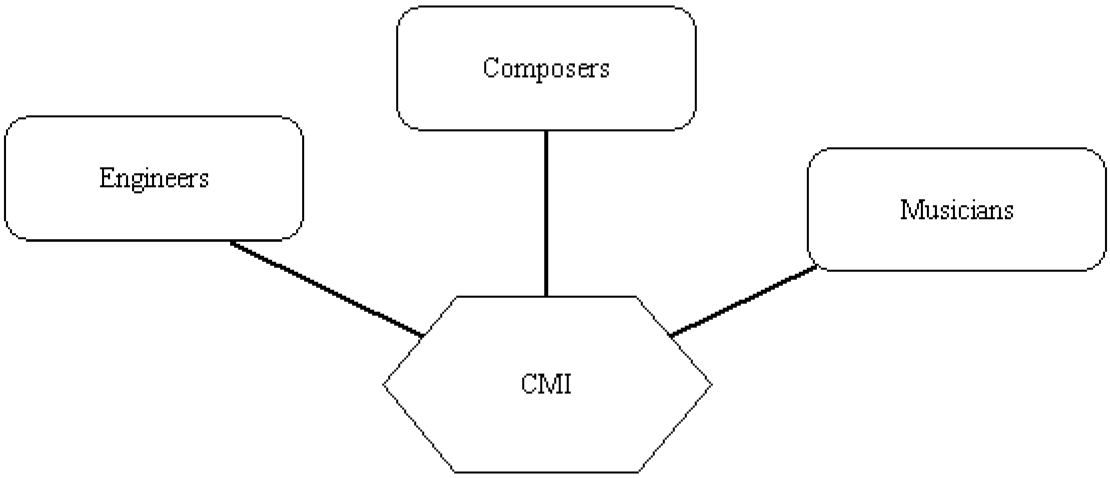

Figure 4.1. The relationship between the CMI and relevant social groups

For each of these groups, the CMI held different meanings (Figure 4.2). To the engineers at Fairlight, the CMI was a sophisticated, dual processor computer, running a real-time multitasking operating system. Composers, attracted to the CMI because of MCL, saw a tool that could perform their already-complete compositions with any orchestration they desired. Musicians and studio producers saw the CMI as a tool that facilitated the rapid composition of musical pieces and as a tool that could add new dimensions to their live performance.

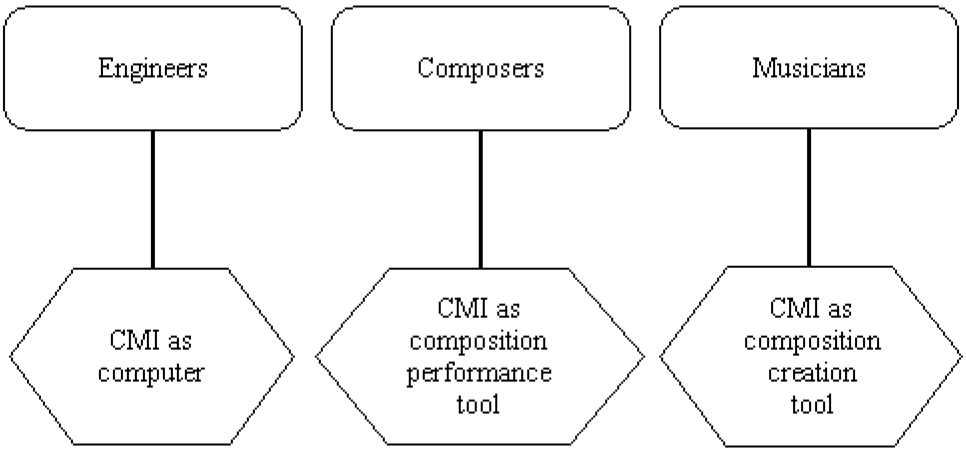

Figure 4.2. Variation of meaning across social groups

What is particularly noteworthy about the CMI is how it simultaneously managed to be many things to a single social group. Not only was there variation of meaning *across* social groups, there was also variation of meaning *within* a single group. For musicians, the CMI was simultaneously a sampler, live performance instrument, compositional tool and computer (Figure 4.3). The meaning of the instrument could shift in step with the procedural or conceptual gaze of the individual relating to it.

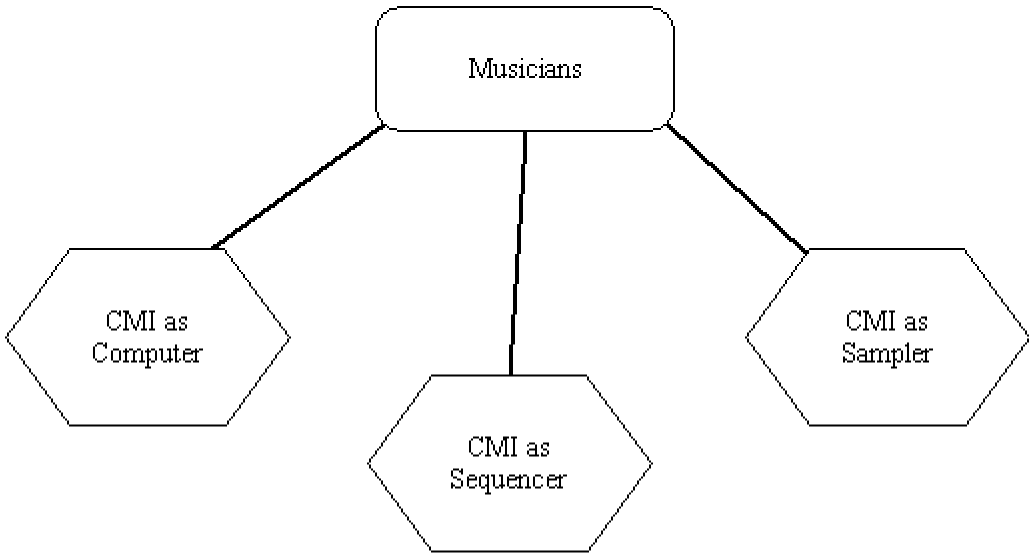

Figure 4.3. Variation of meaning for a single social group

This multiplicity of meanings was made possible by the modular nature of the CMI's interface. With separate pages for separate functions, it was possible to present the instrument as being dedicated to whichever task was foremost in the user's mind. Different conceptual models of music could be presented to a user, particularly in the cases of Page C and Page R. Further, with multiple procedural paths available for the completion of a single task (e.g. the use of the light pen *or* the alphanumeric keyboard *or* the music keyboard for editing single notes in Page R), the CMI could also facilitate different gestural models of interaction.

One of the more significant problems for users of the series I was its sound quality. They were restricted in the range of sources they could use to generate samples, with high frequency sound sources, such as piccolos, returning unsatisfactory results. The increase of system and waveform memory, and the introduction of improved processors in the series II, were presented as solutions to this problem (Figure 4.4).

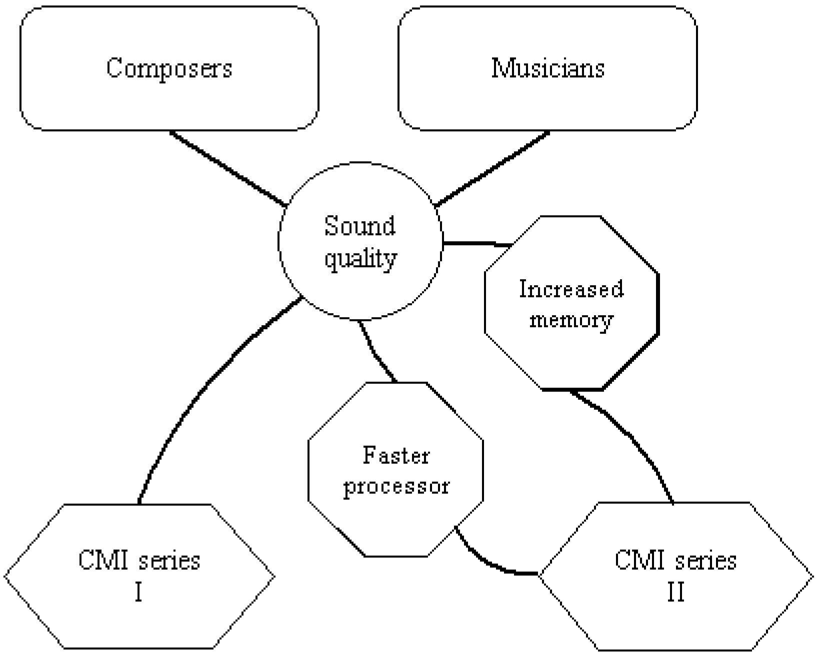

Figure 4.4. The problem of sound quality and its solutions as offered by the series II

A problem that all the social groups that have been identified faced was the lack of MIDI and SMPTE support in the series I and series II. As employees, Fairlight engineers had a commercial interest in seeing MIDI and SMPTE capability introduced, and composers and musicians alike needed their instrument to be able to interface with other equipment that they, or their contemporaries, were using. The series IIx, with its dedicated MIDI/SMPTE card, provided a solution that satisfied all (Figure 4.5).

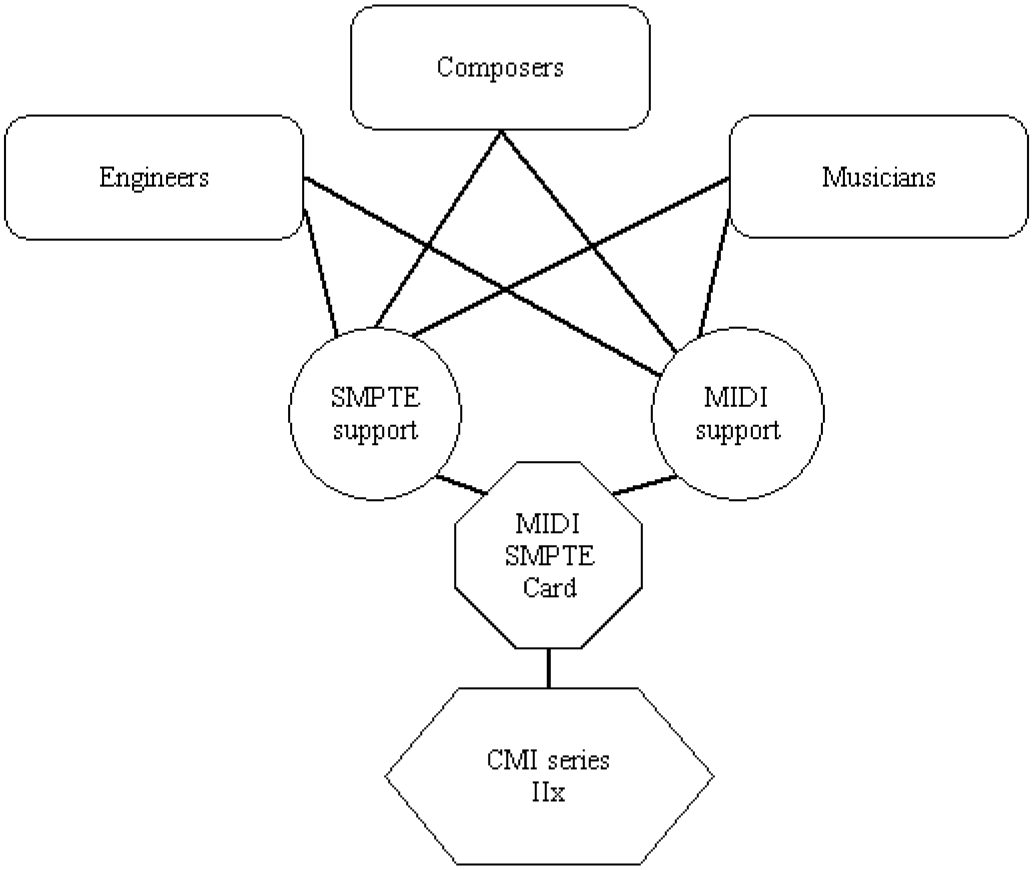

Figure 4.5. The resolution of MIDI and SMPTE problems

Thus far, the relationships between the CMI and relevant social groups, their problems and the provision of solutions, all fit the SCOT heuristic model well. But does this model do justice to the CMI story or does it fail to capture all of its salient aspects? Upon consideration of many musicians' experiences with the CMI, it becomes clear that the model, as defined by Pinch and Bijker, is incomplete.

Opportunity

The SCOT programme, in Pinch and Bijker's words, "brings out the interpretative flexibility of technological artefacts and the role which different closure mechanisms may play in the stabilization of artefacts" (1984, 419). Unfortunately the manner in which it does this seems too narrowly defined. The relationships between social groups and an artefact are only described in terms of problems, and the closure mechanisms accordingly only relate to their resolution. The major omission from such a heuristic model is the role that opportunity can play in defining these relationships, and how stabilisation might be achieved as a result of its inclusion.

Technology offers new possibilities just as often as it solves problems. If presented in a manner that is meaningful to a particular social group, it will be explored and exploited. Its meaningfulness will be derived from its ability, perceived or real, to solve the problems that are central to a particular social group, as outlined above, but also in how it might be exploited to do what are perceived as completely new things. What distinguishes opportunities and problems? Problems hamper existing practices and understandings within a social group. Through the closure mechanisms of rhetorical repositioning and problem redefinition, these problems are resolved. Opportunities, on the other hand, arise out of the relationship between a social group and an artefact, and present and allow the development of new practices and understandings.

As opportunities arise, they will necessarily stimulate new variations of meaning. Different social groups can be expected to identify opportunities specific to them. New opportunities also will lead to the creation of new problems. For an artefact to achieve stability requires that the closure mechanisms already described help to solve the problems that exist between social groups and artefacts, but also that all possible opportunities come to be realised.

Knowing when the extent of possible opportunities has actually been accurately assessed, however, may never be possible, as it would necessitate an external, ‘end of inquiry’ perspective. Thus what is important in achieving closure is that members of a social group *perceive* that all new possibilities have been exhausted. In this sense, closure is attained in a manner similar to the rhetorical closure mechanism that Pinch and Bijker described; members of a social group need only *believe* that all of the opportunities offered by an artefact have been exhausted, whether or not this is objectively the case and leaving aside the question about the possibility of such an objective assessment.

Throughout this dissertation, the new musical opportunities that the introduction of the CMI afforded different musical communities have been emphasised. To frame the CMI merely in terms of problem solving neglects this significant concern, and perhaps most importantly contradicts the experiences of musicians who used the CMI as a result. The inclusion of the notion of opportunity into the SCOT model, however, validates users’ experiences and also provides a better theoretical structure for analysing various aspects of the case study presented in earlier chapters.

Justification for the inclusion of opportunities alongside problems in the SCOT model can be found in Vogel's account of how he stumbled across the suitability of the CMI as a sampler. Recall (from chapter two) that Vogel sampled sounds with the CMI in an attempt to determine which characteristics of sound needed to be better modelled by the synthesis software within the instrument. In his attempt to solve a problem that he faced, he discovered a significant commercial opportunity for Fairlight and made possible numerous musical opportunities for musicians the world over.

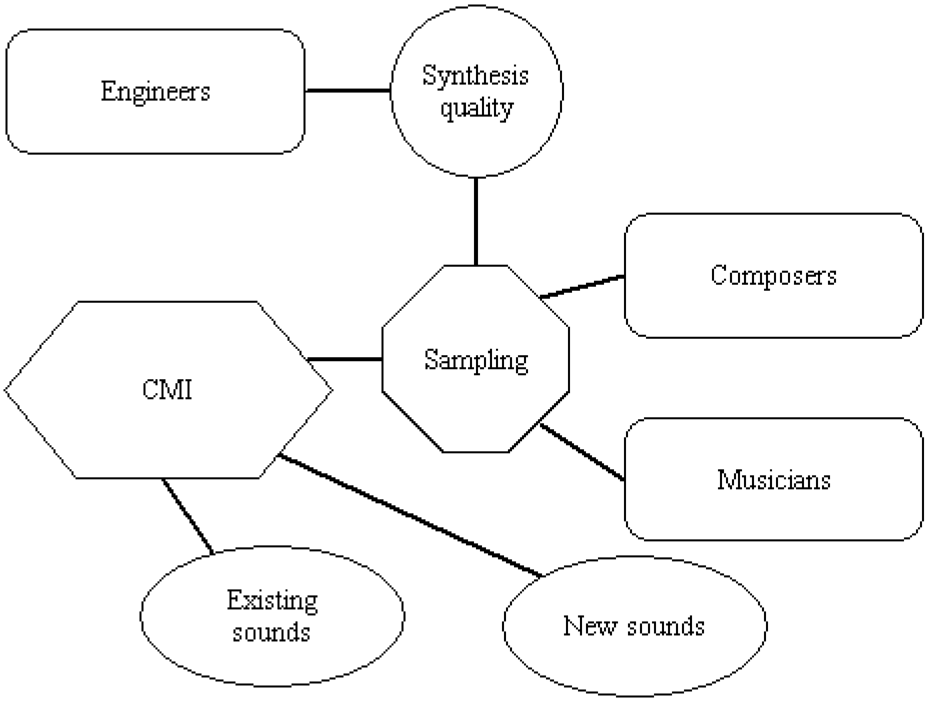

Figure 4.6. The realisation of new opportunities for musicians and composers, made possible by the CMI's sampling capability.

As Figure 4.6 demonstrates, this process was not a matter of problem solving merely through redefinition of the problem. The CMI's sampling function provided users with new musical opportunities. New sounds could be sampled and used in any way that the consumer could imagine. With samples of existing sounds, such as those of a flute or piano, it also was possible to compose a piece of music and be absolutely certain of how it would sound when played on the actual instruments. For example, a classical composer could have a CMI orchestra perform a piece before it was completed, providing a preview of the intended musical result. Existing sounds and instrumentations could also be employed in ways previously unconsidered. A melodic line performed by a bass drum, or a rhythm section comprised of oboes and pipe organs, was not out of reach.

Page R can be seen as one aspect of the CMI where stabilisation was achieved. Alongside the CMI's sampling ability, it became the main feature that drew musicians to the instrument. In the context of the SCOT heuristic model, it can be argued that Page R was perceived to be better at solving problems and as delivering greater musical potential than Page C and MCL. In many ways, it was a better fit with their expectations. In particular, Page R solved many problems that performing musicians had with MCL surrounding the use of the musical keyboard whilst providing many of the same opportunities (Figure 4.8).

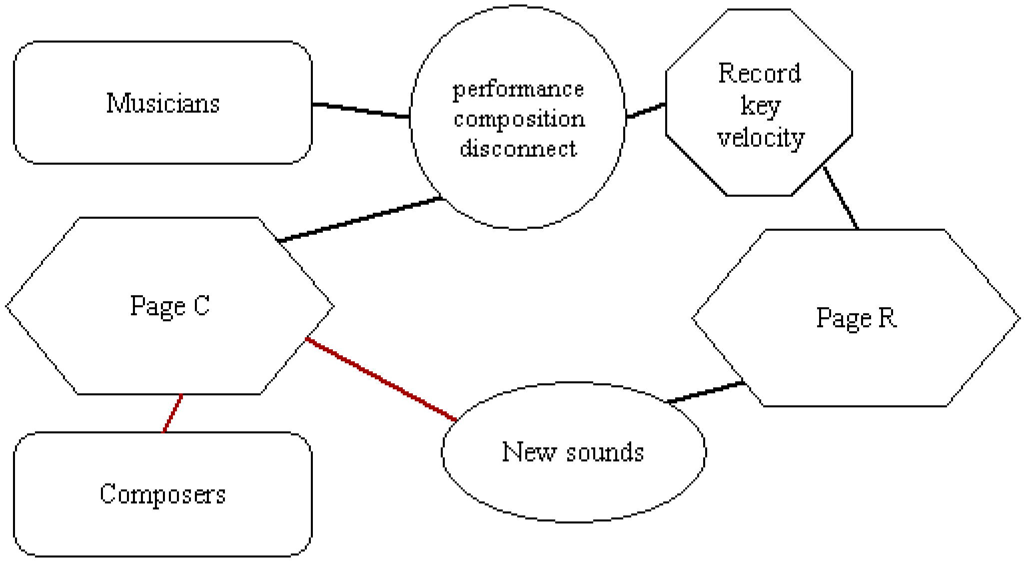

Figure 4.8. Page R solved a problem particular to musicians regarding Page C that didn't exist for composers and delivered the same new opportunity

Further evidence of Page R having achieved stability is uncovered from recognising its important influences on the design of more recent sequencing tools. The real time, pattern-based, multi-voice sequencing approach to composition that Page R introduced dominates sequencing tools released twenty years later. Popular tools such as Cubase[9], Kyma[10] and Logic[11] are all descendants of Page R, sharing many of the same underlying notions about music composition.

With the relationships between the CMI, relevant social groups, problems, solutions and opportunities now incorporated in a single framework, it is worthwhile returning once more to the issue of interface design. It has been argued in earlier chapters that the realisation of new opportunities and the solution of identified problems were dependent upon the relationship

[9] http://www.steinberg.net
[10] http://www.symbolicsound.com
[11] http://www.apple.com/logic

that the interface mediated between users and the CMI's functionality. How well the interface presented the CMI's new technology in a context, familiar both conceptually and gesturally, was critical in defining its success.

In the context of the SCOT model, how is this role of the interface to be represented, and how can the difference between a good fit and a poor one be identified? First the CMI's new technology, and the familiar context within which the interface situates it, must be defined in the model's terms. The CMI's new technology, and its musical implications, can be represented as new opportunities. The musical contexts within which the CMI was assessed and accessed by users can be seen as an existing network of problems and realised opportunities. Concerns regarding musical performance and composition, such as reliability, keyboard sensitivity and the compositional limits imposed, and opportunities afforded, by existing instruments, would constitute this network (Figure 4.9).

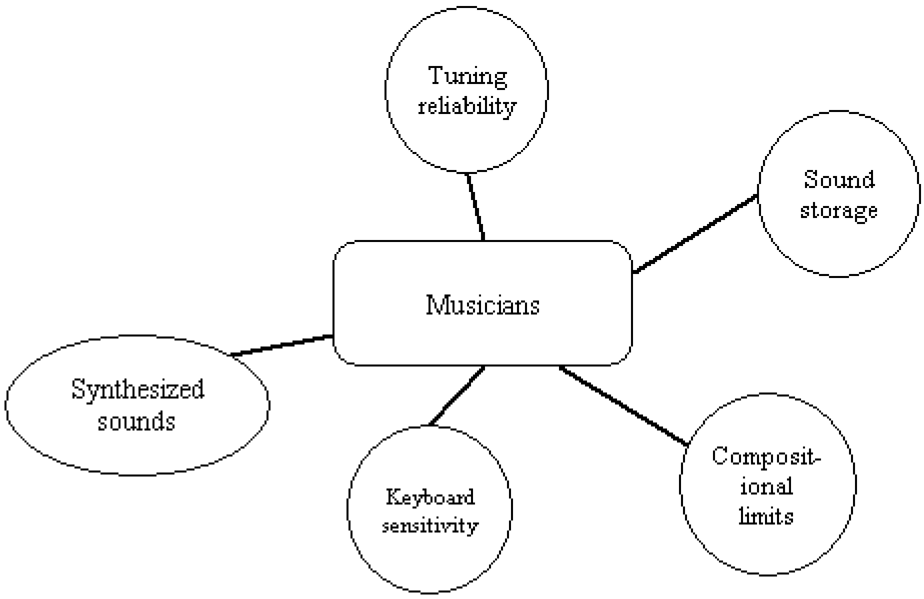

Figure 4.9. The extant network of problems and opportunities for musicians

Thanks to earlier synthesizers such as the MiniMoog and the Prophet V, musicians were already excited about the opportunities that synthesized sounds could offer. The enormous commercial and critical success of Wendy Carlos' groundbreaking *Switched On Bach* in 1968 (Carlos, 2001; see also Pinch and Trocco, 2002, 131-132) is clear testimony enough to that[12]. Similarly, by 1979, instruments such as the Linn drum machine had started to show what could be achieved with a high degree of control over timing in building note sequences. At the

[12] *Switched On Bach* garnered Carlos three Grammy awards, was the first Platinum-selling classical album and remains one of the highest selling classical albums ever.

same time, musicians were acutely aware that analog synthesizers also presented their fair share of problems. Foremost among them was the difficulty of keeping the instruments in tune (Pinch and Trocco, 2002, 193-194). This problem, alongside the often-considerable range and complexity of gestural tasks required to generate new sounds and change their parameters (e.g. try changing patch leads on the fly for a Moog), frequently meant that the instrument was not viable for live performance. Saving sounds and instrument settings was also problematic; the Prophet V allowed settings to be saved to cassette tape but the process took a considerable amount of time.

These problems and opportunities made up, at least in part, the musical framework to which the designers of the CMI interface had to cater. Overall, the instrument addressed many of these problems and facilitated the realisation of existing opportunities, as well as the creation of new ones. How the two sequencing tools, Page C and Page R, which in essence were two different interfaces to CMI technology, functioned in this context is of particular interest, along with the representation of their roles in the SCOT model.

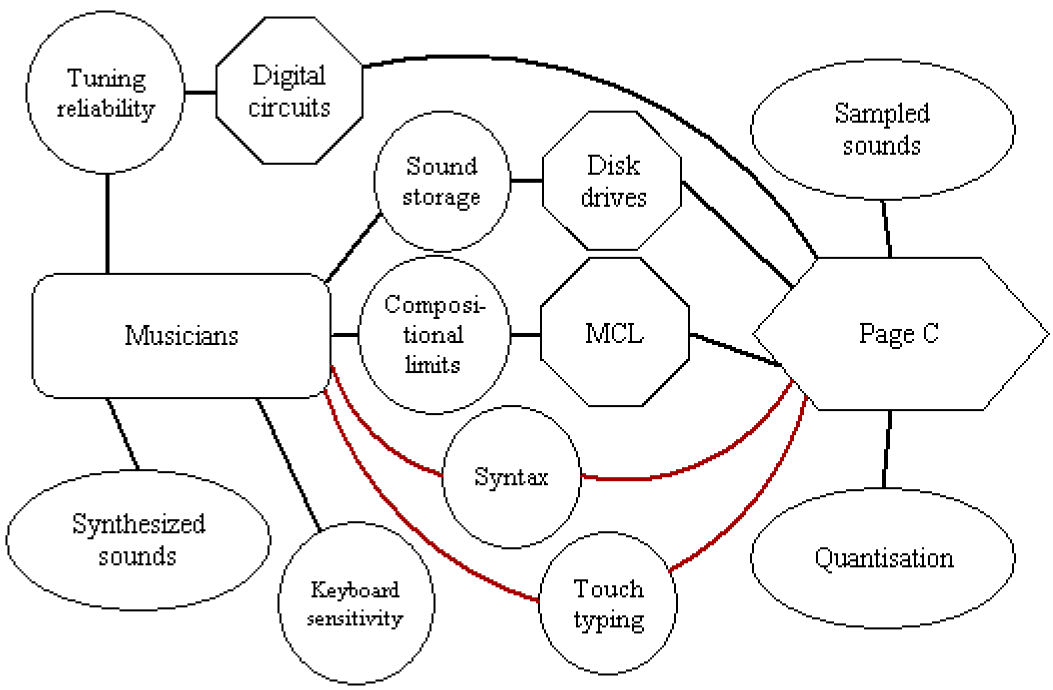

Figure 4.10. The relationship between musicians and Page C

Figure 4.10 provides a limited representation of the type of fit that existed between Page C and musicians. At first glance, it appears to be a largely beneficial match with all but one of the main problems addressed and with considerable access to new opportunities. However, a number of new problems arose from the relationship between musicians and Page C, such as gestural considerations including the need for touch typing skills, and the syntactic differences between MCL and traditional music notation (denoted by red links in figure).

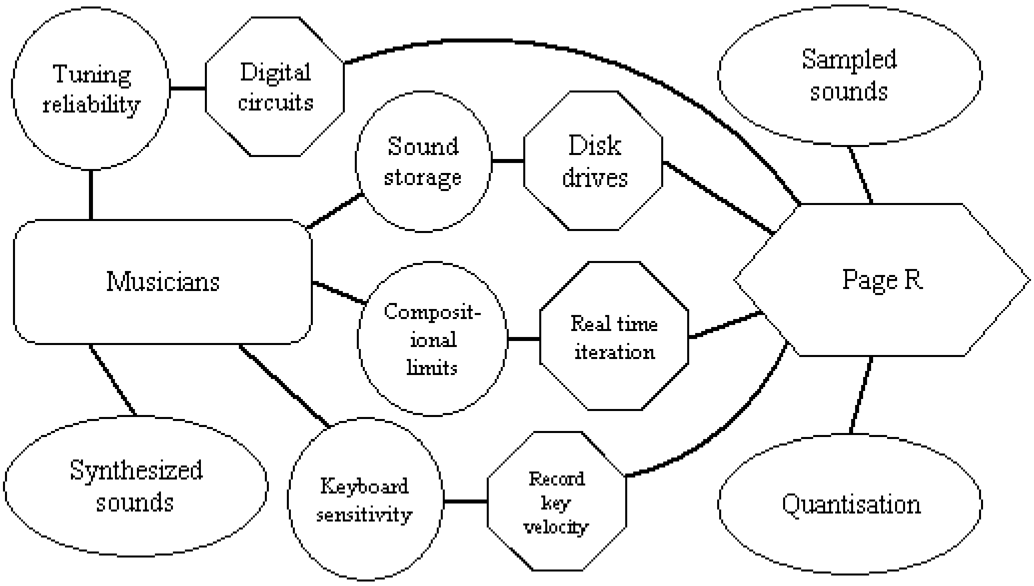

Figure 4.11. The relationship between musicians and Page R

By contrast, in Figure 4.11, Page R can be seen as providing solutions to all of the extant problems and facilitating the realisation of the same new opportunities as Page C. In this sense, it is was better 'fit' for musicians than Page C was. In generic terms, one interface solved more relevant problems and provided the same (or perhaps greater) opportunities than the other. In diagrammatic terms, the greater in number and more complete that the links are which join social groups and the artefact, the greater the fit is between that social group and a technology.

Thus it appears that social-technology relationships can be reasonably well described through a SCOT model. Certainly the inclusion of opportunities alongside problems makes for a better descriptive model than the more limited schema originally conceived by Pinch and Bijker.

Nonetheless, further development is required. Despite all that it does provide, the SCOT model fails to capture some of the crucial details. For example, there is no method for assessing and gauging the relative importance of the problems and opportunities that a social group encountered. In the two preceding figures, the distinction between Page C and Page R for musicians appears to boil down to simply three problems. However, the issue of keyboard sensitivity and gestural concerns (such as the need for touch typing skills) are critical for musicians. It is difficult to make this type of weighting explicit in the model. Considering how to measure and weight the particular paths between social groups and artefacts, and how such weightings might affect the overall model, seem logical first steps to modifying the model.

Music performance and composition are complex, dense tasks, and breaking them down in terms of discrete problems and opportunities runs the risk of oversimplification. As earlier chapters have illustrated, this complexity also is reflected in the interface design, through the interplay of conceptual, procedural and gestural design choices. This interplay leaves open the question of how the SCOT model might be reworked to better tease out the full nature of technology's role in these musical tasks.

Coda

Music is an ever-changing, complex art form with a long, rich history that speaks to people the world over. Nonetheless, as this dissertation has demonstrated, evaluating music technology in terms of its ability to solve problems and present opportunities can be rewarding. It provides a way for exploring the roles that technology plays in a set of unique social contexts.

It is important to recognise that the interface is not just the conduit between new technology and the user, but also between the different social groups that have shaped that technology. It is the crystallisation of a balance of the multiple meanings, problems and opportunities that exist across different social groups, and as such presents them at their most visible and powerful, yet most exposed and volatile. The interface is literally the face of the technology, the first thing that people approaching it see. It has the power to change the way that people think about their craft through its ability to present new opportunities. At the same time, it too is subject to change; if it is deemed incapable of addressing the problems that face relevant social groups, it will undergo transformation.

By virtue of it being such a densely layered focal point for meaning, the interface is of immense value for social historians of technology. In the particular case of the CMI, there is a vast wealth of historical information yet to be explored that holds the potential to shed further light on the role of technology in music. For example, because of the focus of this dissertation on the relationships between the CMI technology and its users, composers and musicians, very little has been said about the social groups *within* the Fairlight company (e.g. software engineers, hardware engineers, system builders, musicians, and so on), the nature of the interactions between these groups and the implications of these associations for the CMI technology and its users.

More generally, further work is required to fully explore the potential and the shortcomings that the SCOT heuristic model possesses. This analysis has demonstrated the need for the inclusion of opportunity in the model, but there are many additional areas to explore. For instance, the distinction between expert and novice users has largely been elided in this dissertation. Little has been said except to recognise that users of the CMI were frequently experts and novices at the same time. The distinction between the two has particular importance for exploring user interface theory in this domain, but the prospect of a user being both expert and novice at once, by virtue of operating in multiple knowledge and social domains, warrants a detailed sociological study.

This topic is but one avenue of exploration that could lead social historians of technology to the development of a heuristic model to better capture and help unpack the multiplicity of meaning that technology holds. The refinement of the SCOT model in this chapter, and the CMI story in general, implies that the sociological investigation of technology, much like its musical counterpart, contains ample opportunities that have yet to be explored.

Glossary

Additive synthesis A technique of audio synthesis that builds complex waveforms by combining sine waves whose frequencies and amplitudes are independently variable.

Analog synthesizer A synthesizer using electronic or mechanical components. Typically, operations such as sound synthesis are performed by manipulating properties of electrical resistance, voltages and so on.

Analog-digital synthesizer A synthesizer with digital controllers used to provide a higher degree of accuracy in manipulating analog sound-generating components. Also called a *hybrid synthesizer*.

Attack The first part of the sound of a note.

Digital synthesizer A synthesizer that uses and processes digital representations of signals.

Gestural interface A system which tracks physical movement and converts it into commands. Typical examples include piano keyboards and guitar fret boards. Atypical examples include the Theremin.

Gestural vocabulary The collection of motor skills employed by an individual in the service of specific task completion.

Glissando A smooth glide through a series of adjacent pitches. Each pitch is played discretely.

Harmonic envelope The shape of a sound, as defined by its harmonics, and their rate of decay.

Hybrid synthesizer See *Analog-digital synthesizer*.

Key velocity The rate of key depression on an electronic musical keyboard. Frequently tied to musically expressive controls such as *attack* or level.

Level The amplitude of a note or sound. Whilst not strictly so, it is frequently understood as the volume of a note or sound.

MIDI Musical Instrument Digital Interface. A protocol for the transmission of timing and control information between suitably capable instruments, making it possible for computers, synthesizers and drum machines for example to control one another and exchange information.

Oscillator An electronic device which generates a periodic signal of a particular frequency, usually a sine wave, but sometimes a square wave or other waveform. In an analog synthesizer, oscillators typically produce regularly repeating fluctuations in voltage–that is, they oscillate.

Portamento A smooth glide through a series of adjacent pitches. Unlike glissando, the glide is continuous.

Quantisation The altering of the times and durations of notes so they fit the beat or sub-beat perfectly.

RAM Random Access Memory. Volatile storage used in computers for actively-used and actively-changing information.

Register One or more output channels of the CMI, grouped together. *Voices* would be assigned to registers, which in turn could be assigned to various octaves of the musical keyboard.

Sampler An electronic musical instrument that uses stored audio signal samples, generally recordings of existing sounds, and plays them back at a range of pitches.

SMPTE Society for Motion Picture and TV Engineers time code. An eight-digit code for numbering each frame of on videotape in the form HH:MM:SS:FF (hours, minutes, seconds, frame number). SMPTE is used to synchronise video and audio signals.

Transposition The changing of key or tonal centre. Used as a musical device by composers. It involves moving a note or collection of notes up or down in pitch by a constant interval.

Vibrato A musical effect where the pitch or frequency of a note or sound is quickly and repeatedly raised and lowered over a small distance for the duration of that note or sound.

Voice A sound, either sampled or synthesised, used by the CMI. Voices had to be assigned to *registers* for playback.

Bibliography

AMPAS. 2004. *76th Annual Academy Awards* (cited 3 October 2004). Available from http://www.oscars.com/nominees/scitech02.html.

Berg, Paul. 1979. i never knew you cared. *Computer Music Journal* 3 (1):38-41.

Bloor, David. 1976. *Knowledge and Social Imagery*. London: Routledge & Kegan Paul.

Carlos, Michael, and Tom Stewart. 1983. *Page R - Real Time Composer Operation Manual*. Sydney: Fairlight Instruments.

Carlos, Wendy. 2001. *Switched On Bach [Enhanced]*: East Side Digital.

Carroll, John M. 2002. Introduction: Human-Computer Interaction, the Past and the Present. In *Human-Computer Interaction in the New Millennium*, edited by J. M. Carroll. New York: ACM Press and Addison Wesley.

Collins, Harry M. 1975. The Seven Sexes: A Study in the Sociology of a Phenomenon, or the Replication of Experiments in Physics. *Sociology* 9 (2):205-224.

———. 1981. Son of Seven Sexes: The Social Destruction of a Physical Phenomenon. *Social Studies of Science* 11:33-62.

———. 1985. *Changing Order: Replication and Induction in Scientific Practice*. Beverly Hills & London: Sage.

Collins, Phil. 1985. *No Jacket Required*: Atlantic Records.

———. 2004. *Phil Collins - No Jacket Required tracks and credits* (cited 16 October 2004). Available from http://www.philcollins.co.uk/jacket1a.htm.

Connell, John, and Chris Gibson. 2003. *Soundtracks: popular music, identity and place*. London; New York: Routledge.

Duncan, Tim. 1989. Australia's High-Tech Hoax. *Business Review Weekly*, 3 March, 42.

Dunn, Ross. 1989. Back Hi-Tech, CSIRO chief urges investors. *Australian Financial Review*, 2 February, 6.

Fairlight Instruments. 1982. *Fairlight Users Manual*. June 1982 ed. Sydney.

———. 1983. *Fairlight Music Composition Language Reference Manual*. Sydney.

———. 1985. Computer Musical Instrument series III configurations. *Fairlight Newsletter*.

Faithfull, George. 1980. The Fairlight C.M.I. *Sonics*, October, 80-85.

Hammond, Ray. 1983. *The Musician and the Micro*. UK: Blandford Press.

Holmes, Greg. 1996. *The Fairlight CMI* (cited 15 April 2004). Available from http://www.ghservices.com/gregh/fairligh/.

———. 1997a. *Fairlight CMI Series II Page 1* (cited 15 April 2004). Available from http://www.ghservices.com/gregh/fairligh/page_1.htm.

———. 1997b. *Fairlight Series II Page 3* (cited 13 October 2004). Available from http://www.ghservices.com/gregh/fairligh/page_3.htm.

Holmes, Thomas B. 2002. *Electronic and experimental music : pioneers in technology and composition*. 2nd ed. New York: Routledge.

Isaksson, Anders, and Barbara Cornelius. 1998. Venture Capital Incentives: A Two Country Comparison. Paper read at 10th Nordic Conference on Small Business Research, at Vaxjo University, Sweden.

Latour, Bruno, and Jim Johnson. 1988. Mixing humans and non-humans together: The sociology of a door-closer. *Social Problems* 35 (3):298-310.

Manuel, Peter. 1993. *Cassette culture: popular music and technology in India*. Chicago: University of Chicago Press.

Mathews, Max. 1969. *The technology of computer music*. Cambridge, MA: MIT Press.

Minogue, Kylie. 2004. *Body Language*: Capitol Records.

Newman, William M., and Michael G. Lamming. 1998. *Interactive System Design*. 3rd ed. Harlow, Essex: Addison Wesley.

Padilla, Michael. 2003. *Strike a balance: Users' expertise on interface design*, 2 September 2003 (cited 18 September 2004). Available from http://www-106.ibm.com/developerworks/web/library/wa-ui/.

Pareles, Jon. 1986. Digital sampling makes waves in music. *The New York Times*, 23 October.

Pinch, Trevor, and Wiebe Bijker. 1984. The Social Construction of Facts and Artefacts: or How the Sociology of Science and the Sociology of Technology might Benefit Each Other. *Social Studies of Science* 14:399-441.

Pinch, Trevor, and Frank Trocco. 2002. *Analog Days : the invention and impact of the Moog synthesizer*. Cambridge, MA: Harvard University Press.

Pressing, Jeff. 1992. *Synthesizer Performance and Real-Time Techniques*: Oxford University Press.

Roberts, Peter. 1989. Fairlight's failure a major blow to MICs. *Australian Financial Review*, 6 February, 8.

Schneiderman, Ben. 1998. *Designing the user interface: Strategies for effective human-computer interaction*. 3rd ed. Reading, MA: Addison-Wesley Publishing.

Street, Rita. 2000. *Fairlight - A 25-Year Long Fairytale*. IMAS Publishing Group (cited 25 October 2004). Available from http://www.audiomedia.com/archive/features/uk-1000/uk-1000-fairlight/uk-1000-fairlight.htm.

Théberge, Paul. 1997. *Any sound you can imagine : making music/consuming technology*. Hanover, NH: Wesleyan University Press : University Press of New England.

Vail, Mark. 1993. *Keyboard presents vintage synthesizers : groundbreaking instruments and pioneering designers of electronic music synthesizers*. Emeryville, CA: GPI Books ; Distributed to the book trade in the U.S. and Canada by Publishers Group West.

———. 2002. *The Hammond Organ: Beauty in the B*. San Francisco, CA: Backbeat Books.

Vercoe, Barry. 2004. *The Public CSound Reference Manual* (cited 10 October 2004). Available from http://www.lakewoodsound.com/csound/hypertext/manual.htm.

Wanderley, Marcelo Mortensen, and Nicola Orio. 2002. Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal* 26 (3):62-76.

Ward, Phil. 2000. Harbouring an ambition. *Pro Sound News Europe*, May 1, 18.

Williams, Elaine. 1982. The Rolls-Royce for musicians. *The Financial Times*, 4 August, 9.

Williams, Raymond. 1981. *Culture*. London: Fontana.

Appendix: Three Blind Mice

The following example is taken from the MCL manual (Fairlight Instruments, 1983). It is a fine illustration of the interplay between piece, part and sequence files.

```
FILENAME = MICE    .PC
   0001  * THIS IS THE "THREE BLIND MICE" PIECE.
   0002  * MOUSEA.PT, MOUSEB.PT, MOUSEC.PT
   0003  * ARE PLAYED SIMULTANEOUSLY
   0004  * SPEED SET TO 8000, PROMPT IS ON
   0005  * !S=8000 P=ON
   0006  MOUSEA
   0007  MOUSEB
   0008  MOUSEC

FILENAME = MOUSEA  .PT
   0001  !K=1
   0002  MOUSE
   0003  * MOUSE.SS IS PLAYED ON KEYBOARD 1

FILENAME = MOUSE   .SS
   0001  * THIS IS THE CENTRAL SEQUENCE TO BE
   0002  * PLAYED BY ALL THREE PARTS IN CANON.
   0003  <
   0004  B=24 O=3
   0005  <E D C R>2
   0006  G+ F+,1/2 F+,1/2 E R G+ F+,1/2 F+,1/2 E R,2/3
   0025  * NOW FOR THE MIDDLE BIT UP ONE OCTAVE
   0035  O=+ B=8
   0045  <G C,2 C B A B C,2 G G R>3 F
   0055  * THE LAST BAR DOWN ONE OCTAVE
   0065  O=-1 B=12
   0075  E D C R
   0085  * NOW REPEAT THE WHOLE LOT THREE TIMES
   0095  >3

FILENAME = MOUSEB  .PT
   0001  * SECOND PART OF THE ROUND.
   0002  * PLAYED ON KEYBOARD 2
   0003  !K=2
   0004  * REST FOR 8 BEATS
   0005  "B=24 R,8
   0006  MOUSE

FILENAME = MOUSEC  .PT
   0010  * THIRD PART OF THE ROUND.
   0020  * PLAYED ON KEYBOARD 3
   0040  !K=3
   0050  * NOW REST FOR 16 BEATS
   0060  "B=24 R,16
   0070  MOUSE
```

This example demonstrates how careful planning can lead to very efficient code. A round is necessarily built upon repetition of a main theme so it is a useful example for showing how the repeat command (the triangular brackets) can be used very effectively. It is also a good illustration of the use of sequence commands within part files (the commands setting the beat and rest period in the MOUSEA, MOUSEB and MOUSEC part files).

Stepping through the code we see that the MICE.PC file sets the speed of the piece and lists the parts that are played simultaneously. The MOUSEA part file assigns keyboard register 1 to the MOUSE.SS sequence file. MOUSEB part file assigns keyboard register 2 to the MOUSE.SS sequence file with the instruction to wait eight beats before stepping through the MOUSE sequence file. Notice here that the beat is explicitly stated in the MOUSEB part file, whereas it isn't in the MOUSEA part file. Because the beat is specified in the sequence file it is strictly unnecessary to explicitly state the beat in the part files. The MOUSEC part file is set up in similar fashion, assigned to keyboard register 3 and waiting 16 beats before stepping through the MOUSE sequence file.

The MOUSE sequence file illustrates how repeat commands may be nested for efficient coding. It also demonstrates how relative specification can be used, in this case for specifying which octave notes are to be played in. Notice also that the line numbers do not necessarily have to increment at any set interval.