Creating Ad Hoc Instruments with Pin&Play&Perform

John Bowers
School of Music
University of East Anglia
Norwich, UK
+44-1603-592452
jmb78@btinternet.com

Nicolas Villar
Computing Department
Lancaster University
Lancaster, UK
+44-1524-510372
nvillar@gmail.com

ABSTRACT
Drawing from conceptual discussions and a reflection on extended techniques in musical performance, we offer the notion of an ‘ad hoc instrument’ as an instrument constructed during the course of interacting with it. To concretise the idea, we explore the utility of ‘Pin&Play’ technology where interface elements like dials, sliders and buttons are pinned to a conductive substrate. We describe an ad hoc sound mixer and ad hoc synthesizer, and report on our performance experience with them. We conclude by discussing ad hoc instruments in relationship to physical interfaces, live coding, ‘living scores’, and issues in contemporary music performance.

Keywords
Ad hoc instruments, Pin&Play, physical interfaces, music performance, new interfaces for musical expression.

1. INTRODUCTION

1.1 The Concept of an Ad Hoc Instrument
In this paper, we explore and demonstrate a realisation of the concept of an ‘ad hoc instrument’. By this we wish to pick out instruments which are, in some significant way, constructed during the course of interacting with them. An ad hoc instrument is made and played at the same time. By interleaving performance with the fabrication of the instrument itself, one can explore extended possibilities for music performance. In our general concept of ad hoc instruments, we do not discriminate between hardware instruments, software instruments, acoustic, electronic, or computational ones, or hybrids. What is of interest to us is how activities normally thought of as separate and sequenced in time (building an instrument then playing it) can be interleaved in ways which can be interesting for an audience. Just as, on some definitions, improvisation interleaves composition and performance, we seek ways of blurring the boundaries between otherwise separate musical activities.

Another way of seeing our concept of an ad hoc instrument is by means of contrast with the instrument destruction performances of artists such as those associated with the Fluxus group. Nam June Paik’s *One For Violin Solo* involves the destruction of a violin as the performance. The composer remarked that he was intrigued by the possibility that a piece would feature an instrument’s terminal sounds, rather than its typical ones. In a contrasting yet complementary fashion, we could see the building of an ad hoc instrument equally as performance but (in the concept’s purest instances) of pieces where the instrument’s *very first* sounds are featured [cf. 1].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NIME'06, June 4-8, 2006, Paris, France.
Copyright remains with the author(s).

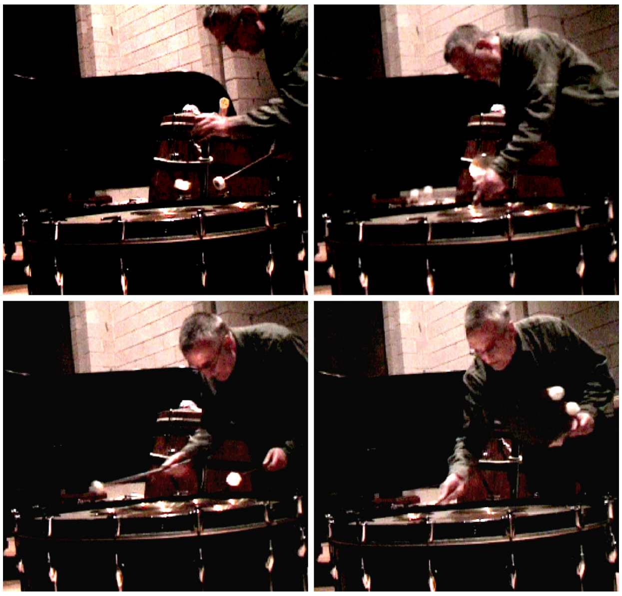

Figure 1. Eddie Prévost of The AMM constructing an ‘ad hoc instrument’ in improvised performance (see text).

1.2 Extended Instrumental Techniques
Of course, a degree of ad hoc construction is familiar from existing music practice. A mute may be placed over the bell of a brass instrument or a loop of guitar effects pedals switched in or out. Our notion of an ad hoc instrument arises at the extreme of these tendencies where somewhat less is fixed in advance. Modifications of an instrument and associated extended techniques are commonly explored in improvisation. Figure 1 shows four frames image-enhanced from a rather dark video of British improvisors The AMM in live performance. Top left, percussionist Eddie Prévost has just placed two small cymbals on top of a horizontally mounted bass drum and is exploring strategies for beating and damping the cymbals and affecting how they pass their energy to the resonances of the bass drum. Top right, about 30 seconds later, Prévost adds another cymbal to the surface. Bottom left, some two minutes further on, Prévost now has six cymbals of various sizes on the bass drum. With this packing of cymbals on the surface, they start colliding with each other as well as offering Prévost the opportunity of striking each one individually or in various combinations. After an intense minute or so exploring these complexities, the music begins to relax and Prévost starts removing the cymbals. Bottom right is about seven minutes after the first frame: two cymbals are being removed to leave just one. In

this episode, Prévost can be said to assemble and disassemble a complex percussion instrument which offers varying musical possibilities along the way. As he makes and unmakes his ad hoc instrument, so an episode is constructed in the music which, similarly, starts small, grows, and returns at the end to its components.

1.3 Configurable Interaction Devices

Based on these general reflections, our work explores the utility of a particular approach to building dynamically configurable interfaces for musical purposes, thereby making a particular kind of hybrid (software/hardware) ad hoc instrument. Research on physical interfaces that involve ad hoc composition and customisation by their users has an established history in fields such as Human Computer Interaction (HCI) and ubiquitous computing (ubicomp). For example, the Phidgets project [e.g. 4] works with a notion of customisable physical interfaces and provides widget taps to let the user bind physical controls to graphical controls, effectively as physical shortcuts for GUI applications. Behavior Construction Kits [12], Electronic Blocks [18] and Triangles [3] support a variety of functionalities (behaviour animation, information access, simple programming) through exploration of variable interface configurations sometimes in a playful fashion. Commercial products enabling the ad hoc construction of interfaces and control surfaces are emerging. For example, the ergodex input system [2] provides a way to easily arrange a number of buttons on a tablet to suit the user's ergonomic preferences. Though the design is proprietary, it is likely that RFID tags in the buttons are detected by a reader in the tablet.

From time to time, authors in the New Interfaces for Musical Expression (NIME) and allied research communities have explored instruments and other artefacts which manifest a degree of ad hocery. For example, the Flexipad element in Vertegaal and Ungvary's Sensorg [15] allows controls to be arranged on a metallic plate. However, Sensorg supports a somewhat limited number of controls which are all hardwired, constraining the ease with which they can be added to the ensemble and freely moved. As with ergodex, the motivation seems mainly ergonomic (e.g. allowing varied arrangements for ease of manipulation) rather than to explore a more extended utility of ad hoc interfaces for musical interaction.

BlockJam enables users to arrange blocks to construct musical structures [9] while Audiopad [11] also affords a level of ad hoc interaction, allowing the performer to compose by manipulating the arrangement of tokens on a surface. ReacTable [6] and Round Table [5] involve the manipulation of physical artifacts on a table surface and have both found application in supporting collaborative performances from multiple musicians.

Bowers and Archer [1] entertain the possibility that the juxtaposition of incomplete or half-made 'infra-instruments' could be a viable form of performance. Like that paper, we are concerned with ways of reformulating the instrument design 'life-cycle', in our case making means for dynamically configuring interaction surfaces available in/as performance. We do this by extending the musical applicability of another technology reported at NIME '05: Pin&Play&Perform [16].

2. PPP: PIN&PLAY&PERFORM

Pin&Play&Perform (PPP) builds on the more generic Pin&Play platform [16]. Pin&Play deconstructs the interface into atomic interaction units or 'widgets'—such as sliders, buttons, dials and joysticks—and provides a mechanism that allows these elements to be freely arranged on a substrate material (Figure 2). The substrate is a flexible laminate, produced in sheets that can be cut to size and used to augment existing surfaces.

Widgets are equipped with small pin-like connectors that allow them to attach anywhere on the surface of the substrate, and in any orientation. This attachment is both physical and digital, in the sense that the substrate acts as a communication medium between the widgets and a connected computer. As soon as a widget is attached it becomes connected, detected, identified by the system and is ready for interaction. All physical interaction on the substrate (attachment, detachment and manipulation of widgets) is reflected as software events on the computer. The net effect is the ability to construct and modify a functional physical interface on the fly without interrupting its operation. A number of potential applications for Pin&Play are discussed in [14].

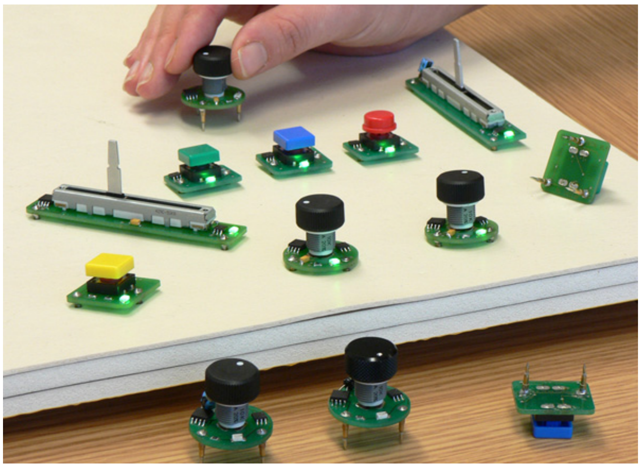

Figure 2. Pin&Play widgets can be easily inserted and removed from the substrate.

Pin&Play&Perform (PPP) [16] operates as a bridge between the underlying Pin&Play system and MIDI (and more lately OSC) enabled applications so as to support musical uses. [16] shows how the PPP approach supports the construction of interfaces which are very varied in their lay-out, thereby making for a high degree of customisation. In that earlier work, the imagined use scenario was one of setting up an interface to taste, or in cognizance of critical ergonomic concerns, and then using it in performance. However, this under-exploits the flexibility with which PPP can be configured to raise control messages in response to interaction events. PPP (and the underlying Pin&Play platform) can support a more radical interleaving of configuration and use. Accordingly, we sought to explore PPP as a platform for realising our emerging concept of ad hoc instruments.

2.1 Implementation

To demonstrate how ad hoc instruments could be made using PPP, we have built and performed with two demonstrators: a PPP mixer and a PPP synthesizer. Both of these allow one to incrementally build a Pin&Play performance interface without interrupting the music. In this way, we intend that the construction of an interface becomes part of the gestural repertoire of a performer. Furthermore, as we shall see, many of the background enabling actions that performers commonly have to do in advance of interaction (e.g. load patches or choose presets) can be elegantly folded in to their interface building and performance activity.

For clarity of demonstration, our PPP mixer and synthesizer are built from just three kinds of hardware widget—sliders, dials and buttons—though PPP can support several more. When a widget is inserted into the substrate, a characteristic MIDI message is raised from which can be extracted the type of widget it is, a unique numerical identifier, and a notification that the widget is attached and 'enabled' (i.e. ready to transmit data). The numerical identifier allocated is the smallest available. When a widget is interacted with, MIDI messages are raised from which can be extracted the widget type, its identifier, and the control data it is generating. When a widget is removed from the substrate, a MIDI message represents the widget type, its identifier and notifies that the widget is no longer enabled. This frees up the previously allocated widget identifier.

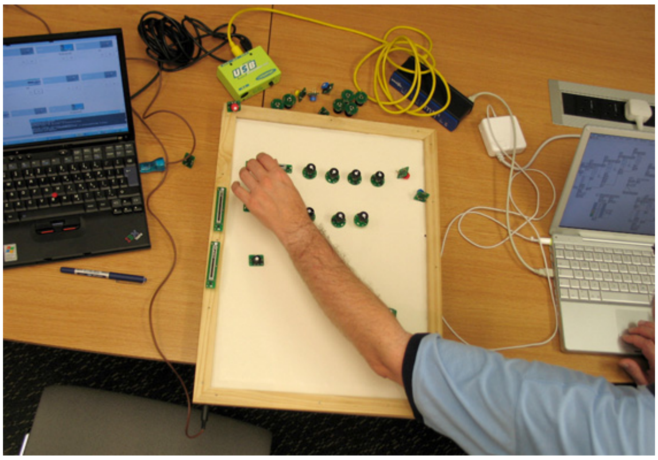

Figure 3. A Thinkpad (left) receives events from the substrate (centre), encodes them into MIDI messages and transmits them via a MIDI link (centre, top) to a PowerBook running Max/MSP (right).

In our demonstrations so far, an IBM Thinkpad X40 has been used to accept data from the substrate and generate MIDI messages which encode widget interaction events. These are passed via a conventional MIDI cable to an Apple G4 PowerBook running Cycling 74's Max/MSP. A patch (parseMIDI) receives the MIDI messages and extracts the information encoded in them—widget type (button, slider or dial), ID, enable/disable or control data values—and makes this available to other patches in the Max/MSP environment. Another patch (showStatus&Values) gives a graphical display of all messages received (both raw MIDI and interpreted).

2.2 PPP Mixer

PPP Mixer allows the performer to build a mixer interface interleaved with the performance of a mix. Up to four stereo soundfiles can be played back with amplitude control by means of sliders. In addition, each mixer 'channel' has associated with it a resonant filter, the centre frequency of which can be set with a dial. A typical interaction with PPP Mixer might proceed as follows.

Placing a slider on the substrate enables mixer Channel 1 and identifies a soundfile to be played back. The initial default amplitude is zero but this is supplanted when the slider is manipulated (Figure 4, top). The performer might, for example, raise the amplitude of the soundfile to moderate levels before inserting a second slider in the substrate. This would enable mixer Channel 2 and identify a second soundfile for playback (Figure 4, middle-top). Having adjusted the amplitude level to taste, the performer may then wish to filter Channel 1 (Figure 4, middle-bottom). A third mixer channel could then be created and so forth. Imagine now that the performer has created four mix channels, each with its own resonant filtering, and the music has reached its most dense moments. The performer may then wish to thin out the mix. Removing one of the sliders will stop the playback of the associated file (Figure 4, bottom). Removing a dial associated with a channel that is still playing will remove the effects of the resonant filter. A performance might be completed by taking the interface down to just one slider with its associated soundfile playing out. The performer could at any moment cut to silence by removing the last slider.

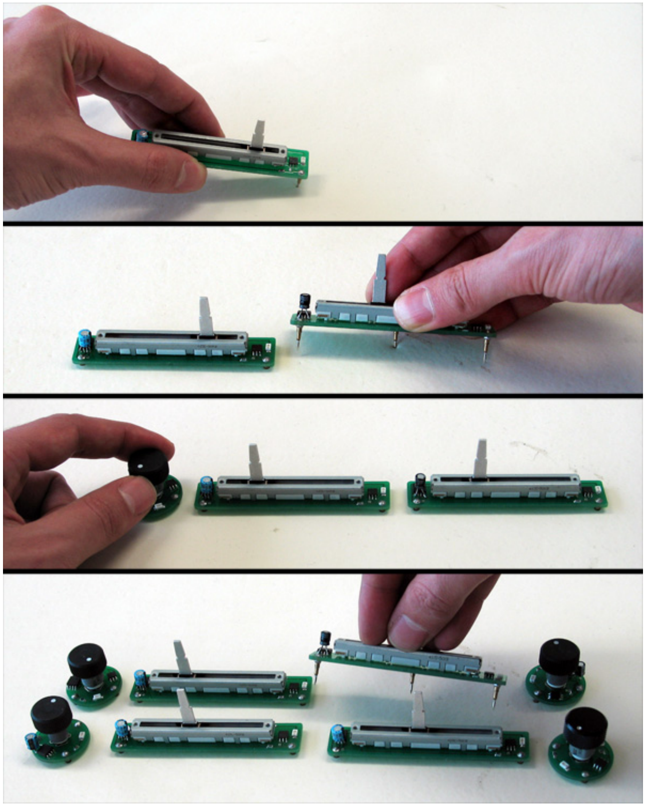

Figure 4. An example PPP Mixer performance sequence. Sliders start, stop and control the volume of associated soundfiles. Dials enable, disable and modify the centre frequency of a resonance filter for each file.

Naturally, we could have built our PPP Mixer application in many different ways, with different effects and default behaviours. It should also be clear from this exposition that the order in which soundfiles are enabled for playback and the resonant filtering effect becomes available is fixed in 'channel order'. This directly reflects the operation of our method for allocating identifiers to widgets. Our PPP Mixer application could be reprogrammed to allow different orderings and/or we could explore different methods for allocating identifiers if we sought different behaviour. Our point is not, however, to build a fully featured mixer with a Pin&Play interface at this stage. Rather, we seek to demonstrate the concept of an ad hoc interface through Pin&Play technology.

2.3 PPP Synthesizer

While we became aware that our PPP Mixer exhibited certain ordering constraints in the behaviours it was capable of, we wondered whether these could also be exploited in interesting ways. Our PPP Synthesizer demonstrates how a synthesis patch could be interacted with by means of incrementally building the interface to it. In contrast to the PPP Mixer, though, the order in which interface widgets are added is 'parsed' so as to further inform how the synthesizer should be configured. Placing a slider on an empty substrate makes available a sine wave oscillator with default amplitude and frequency. Manipulating the slider gives amplitude control. The next dial to be placed on the substrate will give frequency control for that oscillator (sweepable through the MIDI note number range). The next three dials will control phase modulation of the oscillator with the first controlling modulation depth, the next the frequency of a first modulator, and the last controlling the frequency of a second modulator which phase modulates the first (Figure 5, top). In this way, a synthesizer can be configured which has a single audible sine wave oscillator with two modulators cascading phase modulation. Placing a second slider on the substrate makes available a second 'voice' which can be incrementally added to in the same fashion as the first to create another cascade of phase modulation (Figure 5, middle).

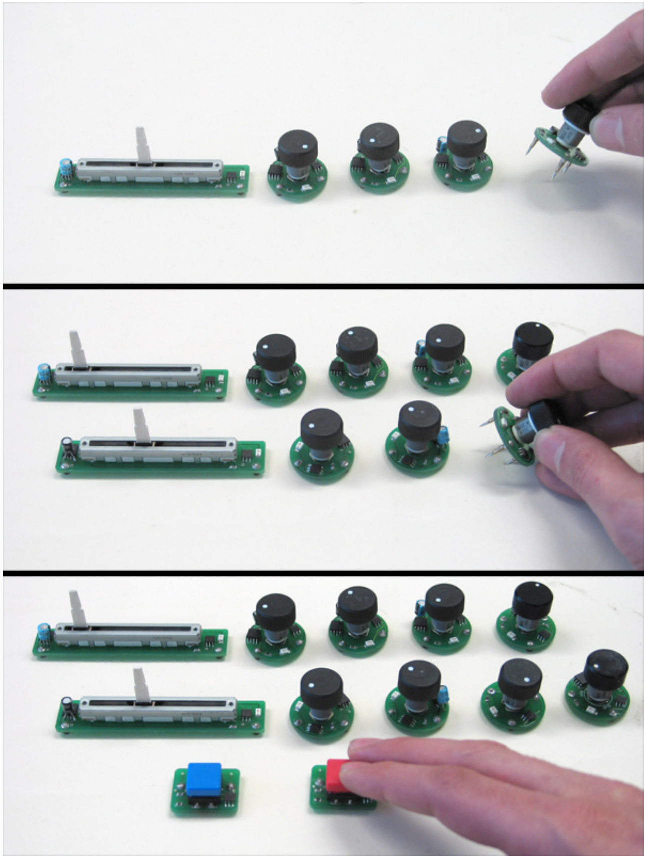

Figure 5. Constructing a PPP Synthesizer.

In our PPP Synthesizer, the exact significance of a widget depends (at least for the dials) on where in order they appear on the substrate. The first dial is a frequency control. Subsequent ones control various aspects of phase modulation. This contrasts with our PPP Mixer where a dial was always associated with the centre frequency of a resonant filter.

Our PPP Synthesizer can be completed by adding up to two buttons (Figure 5, bottom). These have fixed frequency oscillators associated with them. Pressing the button alternately turns the oscillator on and off. The time intervals between presses of the button are measured and used to automatically pulse the oscillator. This rhythm can be interrupted at any time and reset though three (or more) successive manual button presses or the pulsing can be stopped altogether by removing the button from the substrate.

Altogether then, the PPP Synthesizer has (up to) two frequency-variable oscillators which can be complexly phase modulated and two fixed frequency oscillators whose pulsing behaviour can be manually shaped. While this is a simple synthesizer, it is nevertheless capable of a variety of pulsing, rhythmic effects in a retro sort of fashion. The important point, however, is that it demonstrates how we can use Pin&Play technology to interface to synthesis, building interfaces as we configure the topology of synthesis units and do all that without interrupting the music. Furthermore, our PPP synthesizer shows how we can, in rudimentary ways, 'parse' the interface building activity of the performer to make more varied assignments between interface elements and their underlying function.

2.4 Spatial Arrangement of Widgets

The Pin&Play system can track the location of widgets by using additional sensors attached to the substrate. In our PPP Synthesizer example we have shown how meaning can be parsed from the sequence in which the interface is constructed. Additional mechanisms for implicitly extracting significance from the interface can be applied to the way that the components are arranged. Here are some possibilities we have demonstrated.

- The substrate can be regionalized so that different placements of the same widget can have a different significance (e.g. place a slider here and it's a mixer fader, here it creates an oscillator).
- The relative spatial arrangement of widgets (their location with respect to each other) can be used to specify associations between widgets. For example, a dial can become associated with the closest slider to it so that both widgets act together on the same underlying sound source or set of synthesis units. Further subtlety to this can be given depending on the above/below or left/right organization of placements. For example, a dial placed below a slider can control the playback rate of the soundfile whose amplitude is given by the slider, while placing the dial to the right of the slider might allow the dial to control a resonant filter.
- The surface location of a widget can be used to encode (at least) two dimensions of data which would otherwise have to be pre-set. For example, one could make specific file selections in a PPP Mixer in terms of where the fader is put or one can set the frequency of a PPP Synthesizer's pulsing fixed oscillator unit (e.g. higher up the substrate gives a higher pitch). Voodoo Cows, an installation by the second author, specifically exploits such features for pitch determination and sound spatialisation.

Generally, location sensing means that an application is less governed by constraints arising from the exact ordering in which widgets are placed on the substrate. Our simple PPP Mixer and Synthesizer demonstrations were constrained by such affairs (e.g. placing a slider on the substrate in PPP Mixer would always initiate the playing of the soundfile associated with the lowest available slider ID). While this is quite elegant in some applications, it is constraining in others. Being able to pick up location gives added flexibility.

3. PERFORMANCE EXPERIENCE

We have demonstrated our PPP Mixer and Synthesizer on a number of occasions. In particular, one of us (NV) performed as part of *Circuits of Malpractice*, a concert of new performance and installation work at the School of Music, University of East Anglia, Norwich, UK, 3rd October 2005.

In discussions with people that attended the performance we gathered that they enjoyed watching the process of 'construction'. All of the actions involving placement of widgets on the substrate were functional even if their significance was not immediately revealed (e.g. a slider would only be heard to control amplitude when it was moved). This gave the performance a kind of subtle legibility. Even if the exact mappings between action and effect were not immediately transparent, it was clear that the instrument was gradually, as one audience member put it, "written down on the blank sheet" as the piece grew and developed. It was interesting to find that, even though the underlying implementation was imagined to be complex or "very clever", the actual operation of the interface hardly needed explaining. Our audience members recognised the basic controls in our widget set and what can be done with them: knobs are for twisting, sliders are for sliding, buttons are for pressing. Sharp pins on the bottom of the controls and the soft, membrane-like rubberised substrate provide a strong sign of how the two can be used together. Amongst some musicians in our audience, the fact that the actual programming of the instrument is done in Max/MSP gave a sense that this was a technology that they could use and appropriate. It also encouraged speculation about how our applications could be tweaked and modified in their behaviour. The aesthetics of the components was appreciated—the mixture of electronics and mechanics involved. When people had a chance to play with the interface, someone commented on the pleasant tangibility of inserting a pin into the substrate describing it as "walking on snow".

A common suggestion from technically aware audiences and attendees at demonstrations was that we should be using OSC rather than the MIDI link we were using in our setup. This has been taken into account and the alternative protocol is now included in the PPP specification.

4. CONCLUSIONS

Drawing from conceptual considerations and a particular reflection on extended techniques in improvised music, we have offered our notion of an ad hoc instrument as one which supports the interleaving of instrument-construction and performance. To concretise our work, we have explored the utility of a hardware platform for building physical interfaces (Pin&Play) and examined a musical specialization of it (Pin&Play&Perform) which enables us to create simple kinds of ad hoc instrument (e.g. PPP Mixer and PPP Synthesizer). Let us review our experience.

4.1 Pin&Play Constraints

The Pin&Play platform currently has several constraints which impact upon its musical uses. There is a delay between interaction and resultant action due to the slow data-rates supported by the underlying network protocol. However, this is constant (<1s. for enabling/disabling, <200ms. for control events) and does not increase as more widgets become connected. Nevertheless, this means that there are certain musical gestures which cannot be adequately supported by Pin&Play at the moment. One cannot *rapidly* remove and reinsert a slider to punch a sound in and out of the mix, for example. Of course, it is arguable whether one should perform such gestures this way, rather than using a more appropriate widget already fixed in the substrate (a button in the example given). While delays on enabling/disabling are often tolerable, and indeed can encourage certain kinds of usage, the delays on control events are more annoying. We are currently investigating means for ameliorating them.

The idea of pins as a connection mechanism was inspired by Pushpin Computing [7] where pins are used to provide power to distributed sensor nodes. Pin&Play [3] builds on this by using pins to also provide a pathway for data communication. However, pins require a quite deliberative application to the surface. To get them to insert straight you need to choose your location and push them in with some care. This has benefits and problems. It makes for a durable construction which resists being unintentionally knocked but it prevents doing a sweep of the widget from one place to another without breaking connection. It also further inhibits the insertion or removal of widgets as a fast moving performance gesture.

An alternative could be to redesign the substrate to work like the network surface described in [13]. This subdivides the surface into an ingenious arrangement of tiles which enable an object to make the required connections no matter how it is placed on the surface. The Sensetable [10] platform provides a way to wirelessly track the position of objects on its surface, and has been used to implement the Audiopad [11]. Yet another alternative has been proposed in Magic [8]. A surface is embedded with a matrix of electromagnetic coils. Devices are equipped with small modules that allow them to communicate with the surface by means of magnetic induction. All of these would be alternative platforms for realising ad hoc instruments, each with their own technical idiosyncrasies and performance 'feels'. They wouldn't be like walking on snow.

4.2 PPP for Live Coding

Our concept of building an instrument while playing it has much in common with the philosophy of live coding. Indeed, Wang et al [17] discuss how, through writing code live, one can dynamically reconfigure controller mappings, reporting on successful experiments with commercially available control surfaces. However, one can go further and reverse this picture. The creation of an ad hoc control surface could become the means by which live coding takes place. That is, the substrate and widget-set could be simultaneously a means for editing code and performing it. In some respects our PPP Synthesizer simulates this for a very small synthesis universe. It is a simulation because we are activating code which has already been authored rather than properly coding live. With a richer set of widgets, and a more comprehensive approach to parsing user activity with them, one could readily create a live coding programming environment which was also a control surface construction kit. Indeed, this would unify our approach to ad hoc instruments not merely with live coding but also with the existing interest of researchers in physical programming interfaces [12, 3].

4.3 PPP and Living Scores

Another line of future interest for us is in the possibility of coordinating widget deployment with visual representations printed or projected on the substrate. Although much of our work with the PPP applications has been improvisatory, there are interesting possibilities involving pinning widgets into score-like representations to support the realisation of music with a notational/compositional element. Being able to locate widgets on the substrate, and hence their relation to whatever graphical entities are used in the score (and we are certainly not confining the discussion to common notation), is important here. In this way, we might be able to explore a notion of 'living scores', which are not so much instructions in how to realise the music (as scores are commonly conceived) but also the environment in which the music's instruments will be built, as well as providing a space for the gestural illustration of its form as an implicit part of performance. The use of moving image and projections onto the substrate could further enhance the liveliness of graphically notated material in such scenarios.

4.4 Making Performance Activity Legible

Much of the point of building new interfaces (or instruments) for musical expression is to enable a musician's performance activity to be made legible for an audience in new ways. This is sometimes seen as especially important for forms of music which might otherwise be difficult to follow. Our work with Pin&Play technologies adds to this concern in an interesting way. As a PPP Mixer or Synthesizer is being built in performance, in front of the performer and those audience members within perceptual range (and we do ensure that people can see what's going on) are just those interface elements which are needed for interacting with the music. In contrast to using a subset of the sliders or dials on a conventional controller and leaving some idle, the emerging and changing complexity of the interface, over the course of a performance, parallels and helps illustrate the complexity of the ongoing music. There are no surplus widgets to distract the performer or to enigmatically be left untouched. A performer's 'latitude for action' (variation, choice) is clearly displayed in terms of the available widgets: not just what you are doing but what you can immediately turn to. Bringing out a new widget presages an imminent change in the music, helping the audience anticipate and listen for transitions. The coming and going of dials and sliders can give a literal impression of the overall arc of the music and the return of the substrate to emptiness (and silence) ending a performance has been found aesthetically pleasing by many of our audience members.

It is a common complaint of contemporary digital music that watching a performer hunched over a laptop and a MIDI control box is a boring affair. In many ways, our task has been the re-enchantment of dials, sliders and buttons, to return a degree of fascination and intrigue to their use. By regarding these as elements to be worked with constructing an ad hoc instrument in the time of performance itself, we feel we have gone some way towards achieving this. The act of building the interface draws attention to the organization of the music and its relation to performer-activity, matters which are hidden in much conventional technology-rich performance. While we have explored just one way of making ad hoc instruments, we hope we have shown how this concept might help engender some innovative approaches to music making.

5. ACKNOWLEDGEMENTS

JB thanks the UK's Engineering and Physical Sciences Research Council (EPSRC) for a Research Fellowship which supported his contribution to the work reported here. NV thanks the Equator Interdisciplinary Research Collaboration also supported by the EPSRC.

6. REFERENCES

[1] Bowers, J. and Archer, P. Not hyper, not meta, not cyber but infra-instruments. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.

[2] Ergodex Input System. http://www.ergodex.com/

[3] Gorbet, M. G., Orth, M., and Ishii, H. Triangles: tangible interface for manipulation and exploration of digital information topography. *Proc. CHI '98*, ACM Press, 1998.

[4] Greenberg, S. and Boyle, M. Customizable physical interfaces for interacting with conventional applications. *Proc. UIST '02: User Interface Software and Technology,* ACM Press, 2002.

[5] Hoch, M., Bowers, J., Jää-Aro, K.-M., and Hellström, S.-O. http://www.nada.kth.se/erena/table.html

[6] Kaltenbrunner, M., Geiger, G., and Jorda, S. Dynamic patches for live musical performance. In *Proc. of the New Interfaces for Musical Expression (NIME 04)*, 2004.

[7] Lifton, J., Seetharam, D., Broxton, M., and Paradiso, J. A. Pushpin computing system overview: A platform for distributed, embedded, ubiquitous sensor networks. *Proc. Pervasive '02*, Springer-Verlag, 2002.

[8] Minami, M., Nishizawa, Y. Hirasawa, K., Morikawa, H., and Aoyama, T. Magic-surfaces: Magnetically interfaced surfaces for smart space applications. *In Adjunct Proc. Pervasive 2005*.

[9] Newton-Dunn, H., Nakano, H., and Gibson, J. Blockjam: A tangible interface for interactive music. In *Proc of New Interfaces for Musical Expression (NIME 03),* 2003.

[10] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: a wireless object tracking platform for tangible user interfaces. *Proc. CHI '01*, ACM Press (2001), 253–260.

[11] Patten, J. Recht, B. and Ishii, H. Audiopad: A tag-based interface for musical performance. In *Proc. of New Interfaces for Musical Expression (NIME 02)*, 2002.

[12] Resnick, M., Behavior construction kits. In *Communications of the ACM* (1993), 36(7):64–71.

[13] Scott, J., Hoffmann, F., Addlesee, M., Mapp, G., and Hopper, A. Networked surfaces: a new concept in mobile networking. *In Proc. of Mobile Networks and Applications 2002*.

[14] Van Laerhoven, K., Schmidt, A. and Gellersen, H.-W. Pin and play: Networking objects through pins. *Proc. Ubicomp*, ACM Press (2002), 219–229.

[15] Vertegaal, R. and Ungvary, T. The sensorg: A musical cyberinstrument with cognitive ergonomical touch. In Proceedings of IEEE SMC'98 Interntational Conference on Systems, Man and Cybernetics. IEEE, 1998.

[16] Villar, N., Lindsay, A., and Gellersen. H. A rearrangeable tangible interface for musical composition and performance. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.

[17] Wang, G., Misra, A., Kapur, A. and Cook, P. Yeah, ChucK It! => dynamic, controllable interface mapping. In *Proc. of New Interfaces for Musical Expression (NIME 05)*, 2005.

[18] Wyeth, P., and Wyeth, G. Electronic blocks: Tangible programming elements for preschoolers. *Proc. Interact 2001*, IFIP (2001).