Modular Structure: Observations on Managing Compositional Algorithms

Liam Bray, Benjamin Carey, Oliver Bown
Art and Design The University of New South Wales and Architecture, Design and Planning The University of Sydney
Sydney Conservatorium of Music The University of Sydney
Art and Design The University of New South Wales
liam.bray@unsw.edu.au benjamin.carey@sydney.edu.au o.bown@unsw.edu.au

Abstract

In this paper the authors present their initial findings from a study into the contemporary use of modular synthesisers. The study works towards the development of an evaluative process-model examining the strategies modular synthesiser composers use to manage algorithmic complexity. Including observations of users' compositional workflows, which provide insight into the process of building, exploring and rebuilding algorithmic music systems using modular synthesisers. We use these observations to consider what factors cause users to fix certain constraints and explore others. We argue that compositional practice with modular synthesisers provides the ideal testing ground for understanding how machine learning and other algorithmic composition techniques can be used in a co-creative context. Finally, we outline design principles for the design of creative music production tools that are inclusive of the unique requirements of algorithmic music composition systems.

Introduction

Modular synthesis, a musical practice facilitated by the modular electronic instruments developed by Don Buchla and Bob Moog in the 1960s, is experiencing somewhat of a renaissance in the field of electronic music. Since Dieter Doepfer developed the compact and comparably affordable ‘Eurorack’ format the 1990s, there has been an explosion of boutique manufacturers of modules, recent high profile documentaries and publications focused on the music and practice of modular synthesis (Bjørn and Myer (2018), I Dream of Wires, 2014, Patch CV, 2018) as well as the development of music software emulations of modular workflows.[1]

The modular synthesiser is an extremely flexible and customisable musical instrument. Because of its openness and extensibility, modular synthesisers provide an ideal testing ground for studying how artists explore generative spaces. Their recent resurgence has prompted questions surrounding how users navigate complexity in this space. In this paper we investigate the creative practice of patching a modular synthesiser system to uncover the strategies used by synthesists in managing algorithmic complexity in their work. We introduce the concepts of ‘conceptual black-boxing’ and ‘encapsulation’ to describe the way in which synthesists deal with the growing complexity of their patch designs on these systems. Throughout this study we uncover core strategies employed by synthesists, and introduce a novel process-model that seeks to better understand the way in which users view their growing system designs whilst patching.

Background

A modular synthesiser is made up of discrete modules. Each module has its own affordances and constraints, often providing unique musical functions and operations. Most commonly the choice of modules is curated by the synthesist, with a particular sonic goal or compositional strategy in mind, though it is commonplace for a synthesist to change modules often to better suit their musical intentions at a given point in time. The unique grouping of modules may be described as the synthesiser’s architecture, as the discrete modules together define the building blocks of the system’s range of possibilities.

1 Some examples include Native Instruments’ Reaktor Blocks, Softtube’s Modular, the free and open source VCV Rack, Max for Cats’ OSCiLLOT, Propellerhead’s Complex-1 and Audulus

A synthesist's choice of modules is a often an iterative process, as many modules take on multiple roles or interact with other modules in some highly specialised way. This allows for a large range of possibilities. This can mean the synthesist spends significantly more time developing the synthesiser's architecture as compared with using a closed system or semi-modular synthesiser. Popular websites such as ModularGrid (ModularGrid 2019) provide synthesists a virtual platform for the development of their synthesisers architecture, with a community driven database of modules being carefully maintained.

The synthesiser's architecture describes the range of potential actions that can be made within the system, and the potential sonic outcomes that could occur based on interconnecting the system in particular ways. The synthesiser's architecture also represents the intentional choices made in selecting modules to include in the system, and in this way the architecture itself is a set of creative constraints for the musician.

The primary mode of interaction with a modular system is through the configuration of system routings and module calibration in the form of a "patch". A patch represents the interconnectedness of each of the discrete modules. As the system has a fixed architecture at any given moment, the patch represents the precise manipulation of the structure of this architecture. The synthesist, through patching, navigates the range of potential actions that can be made within the system governed by the constraints of the architecture.

The paradigm that emerges from this description of a modular synthesiser is that even a small modular system can have the potential to be highly constrained but yet have near limitless configurations. It is the observation of the authors that this is one of the primary attractions to the use of modular synthesisers as a compositional and performance tool.

Transparency and complexity

At the level of the individual module, different modules express various levels of functional abstraction to the user. An example of this can be found in the differences between voltage-controlled oscillators (VCOs) available on the market. Consider, for example, the difference between a standard analogue oscillator and digital 'macro' oscillator, containing multiple synthesis algorithms. Module parameters and states are for the most part visible on the faceplate of each module, and the connections between modules that a user makes are represented by physical patch cables. As a physical instrument therefore, both the architecture of a modular system (the collection of modules), and structure of a specific patch (including the fixed parameter settings of individual modules), suggest a working environment that may be considered a kind of 'white box' system. That is, objectively speaking, all of the details surrounding the both the system architecture and designed algorithm are mostly visible, at all times, to the user.

With regards to control and automation in a modular patch, despite the inherent transparency of the system's architecture, a patch can quickly become an exceedingly complex network of compositional structures, full of nested hierarchies of control and automation. These networks may involve the use of automated components such as clocked CV and gate/trigger sequencers, triggered or looping envelopes and low-frequency oscillators, as well as more immediate, performative controls such as touch plates, joysticks, potentiometers and buttons.

One common type of control function utilised in a modular synthesiser are modules that can generate random and pseudo-random voltages for use throughout a system. Whether existing as self-contained modules or as a collection of modules (e.g. a noise source fed through a triggered sample-and-hold circuit), this class of modular functions are commonly used by synthesists in order to create variations of various degrees of unpredictability in a modular patch. For the sake of consistency, here we will describe these functions collectively as 'sources of uncertainty', a term that is derived from the original Buchla '266 Source of Uncertainty' module.[2]

Encapsulation and Conceptual 'Black-boxing'

As the precise combination between performative and automated control in a modular patch is up to the synthesist, a large part of the creative practice involves defining the function of each individual module within the structure of a patch as it is being formed. Given an architecture and patch structure's 'white-boxed' nature, the level and degree of complexity that arises when making music on a modular synthesiser is something the user must manage when working on the instrument. To manage complexity, typically a synthesist moves from defining low-level sonic and control structures into the development of higher-level control structures. Both low and high-level structures may be fully automated or leave open the potential for

2 For further information of various types of random sources, the reader is referred to an overview of this class of modulation source on p.252 of Bjørn and Myer (2018).

performative control. Regardless of the type of control desired, it is our contention that the decisions made in developing their compositional structure (patch) involve various degrees of algorithmic decision-making. As a patch becomes more complex, the synthesist often finds themselves devising strategies for managing various elements of their patch.

One such approach that is of interest is that of 'encapsulation', a common process in object-oriented programming practices whereby the details of an algorithmic process (function) devised by the programmer are 'hidden' from view, and accessed from outside of the encapsulated object (Chen & Crilly, 2016). Given the openness of a modular system, encapsulation may be considered a conceptual compositional strategy, that turns a series of often-automated control structures into a self-contained compositional algorithm, or 'black-box'. Whilst the details of this algorithm are often still visibly apparent to the user (displayed through both the state of connected modules and their connections via patch cables), by conceptually 'black-boxing' a compositional algorithm the synthesist may be aided in moving from low-level to high-level design of their patch. Encapsulation, as a heuristic is a creative search strategy which like problem-solving heuristics such as "working backwards" and "trial-and-error", provides a meaningful method in aid of creative problem solving (Klahr, 2000).

A musical algorithm or process may be said to be black-boxed once its internal process no longer needs to be accessed. The algorithm/process is 'doing something', and it is now the result of this process that is available to be made use of, similar to the use of a defined function in object-oriented programming languages. Depending on the particular algorithm, it may be used either as a musical/sonic material, or as a generative device for driving other processes. When this happens the synthesist takes on a new type of interaction with the system, delegating or relinquishing detailed creative control of the encapsulated component, and dealing only with the outcome of the process itself. This could also be described as establishing 'meta-creative' control over the system's configuration through the process of encapsulation. In addition, modules and functions that act as 'sources of uncertainty' may also be considered black boxes in their own right, as the user engages only with the results of their internal mechanisms.

In addition to its relation to problem-solving heuristics, our use of the concept of conceptual black-boxing draws on Akrich and Latour's description of technical objects in networks (Akrich and Latour, 1992). This description posits a distinction between technical objects that are either stabilised or non-stabilised as artefacts. As the second author has noted elsewhere:

"Non-stabilised artefacts are those considered to be still within a development and innovation phase, they are artefacts for which meaning is still emerging [...] By contrast, stabilised artefacts are those artefacts that have exited this innovation network and entered the real world to be made use of." (Carey, 2016)

This description is given in the context of actor-network theory. According to Akrich and Latour, stabilised technical objects can be considered 'instruments of knowledge' (Akrich and Latour 1992, p. 221). In order for us to access knowledge through these objects, one must be able to engage in a process of black boxing:

"the conversion of sociotechnical facts into facts pure and simple depends on the ability to turn technical objects into black boxes. In other words, as they become indispensable, objects also have to efface themselves." (Akrich and Latour 1992, p. 221)

This descriptive model of non-stabilised to stabilised technical objects applies to the process of patching a modular synthesiser as the synthesist moves from low-level components towards higher level structures in a fluid iterative cycle. Here we argue that conceptual black-boxing takes place when the synthesist (due to either the cognitive limitation, creative interest or both) encapsulates a component of either the architecture or configuration into a compositional algorithm. The algorithm may therefore be considered, for the purposes of the particular patch in question, stabilised.

In the process of navigating a synthesis architecture in order to create a unique patch, it may be said that the synthesist is constantly accessing knowledge about the broader context of their system by observing the results of algorithms they have black boxed throughout their process. In doing so, the user is able to gain knowledge about the current state of the algorithm with with they are engaging in practice. For Akrich and Latour, as black boxing describes the ability of a technology to act as an instrument of knowledge, this process enables the transcendence of what is knowable from inside the object's innovation network itself. In order for the user to understand the implications of the technology in its current usage context, the inner workings of the artefact and the complex and heterogeneous elements of the innovation network must fade into the background.

Methodology

Our initial study investigates the use of modular synthesisers through the observation of a small group of users (including one of the authors) creating patches with modular synthesisers. Intentionally following a creative course and documenting through, video, discussion and patch notes, the types of strategies that were employed.

The goal of the study was to discover the methods by which users managed complexity in their algorithmic patches. In each case, users worked within the confines of a specific synthesis architecture, this was either their own modular system or one provided by the researchers. In the design of this study it was identified that there was a need for a method to investigate the nature of the kinds of creative control being designed by the composer whilst completing a task. To this end, a descriptive process model was used to establish a dialogue, this was both through discussion and musical improvisation. The study was centred on three creative tasks, which were completed by each of the participants separately.

Task 1: Single Sequencer, Single Voice, at least one source of uncertainty, 15 mins

Task 2: Percussion/rhythmic sequence, Multi-voice, at least one source of uncertainty, 15 mins

Task 3: Ambient/Textural, Slow Rate of Change, at least one source of uncertainty, 15 mins

Each of these tasks required the synthesist to complete a musical task within the constraints of the description. This functioned as a set of arbitrary motivations to push the participants into a constrained space which they would then have to develop a strategy to solve. This could be likened to a set of decisive constraints as outlined by (Biskjaer et al. 2014). Decisive constraints are based on the observation that there are certain intentional creative actions which prune the creative solution space, but act as an enabling factor in advancing a creative course (Halskov, 2011).

The study took a novel approach to the evaluation of real-time experience, with the goal of capturing nuanced detail about the user's creative focus and sense of progress in a given task. This was conducted with the use of two primary quantitative methods, firstly a descriptive process model adapted from the self-assessment manikin (Bradley et al. 1994a) and secondly, an informal speak-aloud method based on the think-aloud protocol (Boren, 2000).

The self-assessment manikin

The self-assessment manikin (SAM) is a non-verbal, graphic depiction of three affective dimensions. This method was developed as a novel and less cumbersome way to directly asses pleasure, arousal and dominance in response to an object or event (Bradley et al. 1994a). The method is often used when reliance on verbal information is not easily accessible, for example with children or when the researcher and participant do not speak the same language. The method has been applied to a wide variety of subject matter for example, Bradley (Bradley 1994b) has demonstrated the method's validity in assessing participants effective responses to sound.

In our study, the user was given a series of five dimensions, each dimension tracked a bipolar adjective pair. These were, as seen in Figure 1*: surprising to unsurprising, interesting to uninteresting, I'm responsible to it's responsible, simple to complex and powerful to limited.* As the user was completing one of the three musical tasks, they were asked to update the markers on each of these dimensions. Updating the dimensions marker *during* the task was central to type of data this method is designed to capture. As was discovered in a preliminary practice-based workshop where this method was initially developed, the composer builds a simple mental model of the interconnectedness of the systems architecture. This model is constantly being iterated on, functionally through encapsulation and feedback of the system, sonically through sonic feedback and conceptually as the range of possibilities is explored. The descriptive process model is designed to capture the participants iterative 'mental model' throughout the duration of a composition, by requiring the composer to provide quick, self-reflective responses to a series of simple creative boundaries.

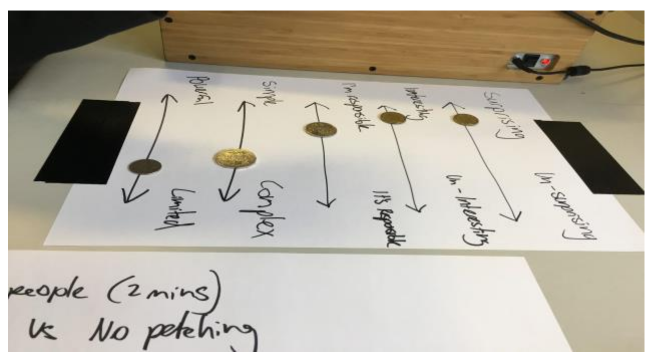

Figure 1 - The five dimensions of the process-model

This method was mixed with the second method, engage with the participant, and ask them to reflect and discuss verbally about their process and experience. This sometimes took on a think-aloud type form, as the participant explained the connectivity of their patching, as well as their musical ideas as they performed the actions of constructing a patch (Newell & Simon 1972). But overall, remained an informal discussion around the descriptive process model and the musical task being completed.

Think-aloud Protocol

Protocol analysis is a research method which requires participants to verbally describe their thoughts during a task (Ericsson, 2006). It was initially used in the fields of cognitive psychology and as a method to derive accurate, repeatable verbal reports in a laboratory setting. The data resulting from verbalisation when using protocol analysis is defined by `levels', each level indicating a decreasing amount of interference resulting from non-task-related cognitive processing. Level 1 verbalisations are those that have a clear relationship to how they are represented in short-term memory, the example given by Boren et al. (Boren, 2000) is that of solving a math equation. Level 2 verbalisations are those that require the participant to transform images or abstract concepts into words before they can be verbalised, an example of this is a chess player verbalising as they make a move. Finally level 3 verbalisations are those that require additional cognitive processing before they can be translated into words, these verbalisations require the participant to access knowledge beyond what is contained in short-term memory.

Protocol analysis was the basis for the think-aloud protocol (Boren, 2000) used extensively in the field of user-experience and human-computer interaction, where it is somewhat decoupled from the laboratory setting and is not necessarily implemented with the scientific rigour required in the field of cognitive science. Nielsen, a significant figure in the field of usability engineering remarks, ``thinking aloud may be the single most valuable usability engineering method" (Nielsen, 1994). Its usefulness is evident in the broad range of studies it can be found in, both more loosely implemented as an investigative tool and as a rigorous way to way to uncover the cognitive state of a user completing a task.

When used without clear consideration for the protocol and the nature of non-reactive verbalisations as outlined by Ericsson and Simon (Ericsson, 1984) the validity of the type of generalisable data must be considered. Boren et al. (Boren, 2000), examine the discrepancies between common practices in think-aloud protocol and the verifiability of the data it generates. Importantly they arrive at several key considerations which work towards reconciling the inconsistent use of think-aloud protocol in a research context. These include: giving detailed initial instructions for thinking aloud, regularly reminding participants to think-aloud and not intervening once a task has commenced.

Evaluating real-time algorithmic composition

In this study the focus is on a self-reflective account of the composer's experience. This importantly maps onto our understanding of the nature of our cognitive processing skills (Mumford, et al. 2003, Sawyer, 2011). The iterative, real-time nature of patching, parallels models of improvisation (Gifford, et al., 2017) in that it requires the synthesist to actively reflect on actions made, then refine them in order to achieve a perceived creative outcome.

To closely examine creative search strategies employed by modular synthesis and to better understand the conceptual model developed in real-time by synthesis, this study adopted a blended method. Combining both a refined process-model tracking five dimensions over the duration of a task and asking the participant to speak-aloud during this process.

However, we propose that in real-time algorithmic decision-making where the composer is engaged in the active processes of deciphering meta-creative control over the system's configuration — through strategies such as black-boxing or encapsulation, a process of *evaluative* reflection (Fig 2) takes place. With the intention of better understanding evaluative reflection (discussed further below), users were spoken to during the process of completing a task. These interventions by the researcher modify the nature of the data produced from what could be considered an example of the think-aloud protocol, to what is an account of the user speaking-aloud, with the researcher during a task.

Schön (1983) defines this type of real-time action as "reflecting-in-action" a "reflective conversation with the situation". Schön's model describes *refinement* as 'back-talk', that by responding to actions through refinement the practitioner "reflects-in-action on the construction of the problem, the strategies of action, or the model of the phenomena, which has been implicit in his [or her] moves." (Schön, 1983). In real-time algorithmic decision-making, the practitioner is engaged in deciphering strategies of action in both proto-creative and meta-creative contexts. *Evaluation* is used here to describe those additional meta-creative decisions that are being made about the systems configuration.

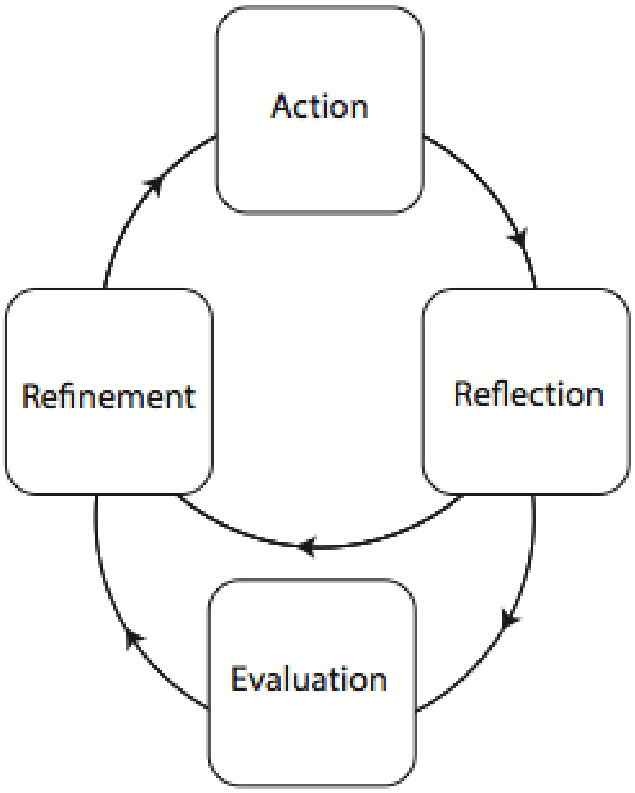

Figure 2 - A model of reflection in meta-creative action.

Findings

Core building blocks of Modular Strategies

After reviewing the audio-visual documentation and transcripts of the sessions, a thematic analysis was conducted and some themes were identified amongst the 3 participants in approaching the management of algorithmic complexity in their work, these themes are identified below.

Filtered/tamed use of multiple 'sources of uncertainty'

Participant 1 chose to work on their own modular system. As with all of the participants, the first participant's process involved the deployment of multiple automated forms of control, including the use of clocked sequencers, looping envelope generators and low-frequency oscillators. In addition, from the outset the participant made use of multiple sources of uncertainty in his work. As the participant explains, throughout their process they seek to include these sources of uncertainty as a way of providing something to 'tame' throughout the development of a patch:

'...quite often that's what I'll do, like so I'll get something like this that just, like, does stuff itself and then finding ways of taming its output.'

Importantly, the participant began the session without a clear musical goal in mind. Their working method evolved from the development of a small yet complex musical seed which then developed into a sophisticated control network. This musical seed consisted of a series of fast clicking sounds created by sending a rapid clock signal into the trigger input of a resonant bandpass filter. The digital clock module used, the Make Noise 'Tempi', was patched to output a series of curated clock divisions at random, creating a flurry of short, unpredictable percussive events. This was further augmented by driving a sequencer from the same clock source, to change the cutoff frequency of the filter.

This simple three module algorithm provides a good example of the integration of a 'source of uncertainty' with predictable automated control mechanisms. Whilst the clock divisions and sequence values were chosen by the synthesist, a randomised shuffling of clock divisions ensured that these events would be ever changing.

Devising nested, contingent control processes through blackboxing

Participant 1 then sought to use these events to devise broader algorithmic structures that were contingent upon this initial rhythmic material. This resulted in two, slower layers of sound, both of which were created using rising and falling envelopes initiated by much slower divisions of the randomised clock source. The first layer was created by periodically processing the click gestures through a comb-filter, creating a metallic resonance, the second layer occasionally revealed a sustained tone from an analog oscillator. Whilst fine-tuning the shape of the envelopes controlling the volume of these layers, the participant was asked to update the 'Responsibility' dimension of the process model:

'The specifics of what's inside them (the new textures) it's responsible for - I feel now - but I'm responsible for its global sound.'

The process described above illustrates well the process of 'encapsulation' and 'conceptual black-boxing' in patch design. Whilst the output of these shuffling clocks were put to use to drive other events, manipulation of the internal mechanism of this algorithm was no longer required. By conceptually 'black-boxing' this compositional algorithm, the musician was then able to shift their attention to higher level concerns such as sound quality, dynamic contour, and the development of more contingent events:

"It's kind of a process of taking a large amount of activity and then stripping it down. Because I've got a lot of fast clocks that are doing the really quick jittering I've then taken something that is sweeping the comb filter and I'm taking that out to then trigger another envelope."

From these reflections we can observe that for this participant, the initial algorithm had a great bearing on subsequent decisions made down the chain. their inclusion of sources of uncertainty in the early stages of the design suggest that a high degree of control over the developing structure of their patch is delegated to the initial algorithm. After subsequently adding more nested layers of control, the participant began to query the overall algorithmic design by changing the rate of the original clock driving the patch. This is a good example of the way in which the low-level building blocks of such black-boxes can still be accessed by the synthesist when required:

"So I usually get to this point like with a patch like this where there are many parts that are contingent upon each other, where I start to kind of lose myself a little bit in the system, and I start to kind of poke and prod a kind of go 'what do I like about this' - if I take something away, do I still like it?"

Prioritising automation and sequencing as fundamental control structures

Participant 2 chose to use the modular system made available to them by the researchers, meaning that this participant had never used this particular modular synthesiser. They had an opportunity to explore modules that were not familiar to them, and discuss with the researchers prior to commencing the study. The participant was an experienced modular synthesist who had the capacity to navigate the system and build successful musical compositions. However, unfamiliarity with the system did influence the range of possibilities they were able to pursue.

The participant was asked if they would like a musical task, but decided to start with no clear musical description in mind. This became an important distinction between the compositional strategies employed by participants 2 and 3. Participant 2, on completion of a piece was asked to describe their compositional approach:

"I'm not trying to be specific about when something should happen or how something should sound like, but letting that be really chaotic, and then kind of just reducing that chaos, and kind of tuning it until things are falling into a range."

They paraphrased a quote by Iannis Xenakis that they identified with:

"One establishes an entire range between two poles — determinism, which corresponds to strict periodicity, and indeterminism, which corresponds to constant renewal...This is the true keyboard of musical composition." (Xenakis 1985)

The participant described how this idea translates into their compositional strategy:

"I'll test out and input first and think, hey this sounds good between there and there, and when it leaves this range i'm not into it. So i'm gonna attenuate between these ranges and switch over to a generated source. I'll just keeping going like that with about eight parameters"

The participant, having patched a series of modules, often uses a gate sequencer to change the position of parameters within the patch, meaning they can *"hit a single button"* and change the pitch, length, tone of each generated source. The participant will use this technique in the context of live performances, auditioning as they search through the tuned-random parameter space until they like something. They will then record that to a loop in a sampler and continue to search.

The participant describes a method which uses low-level building blocks to define a clear parameter space. This was described as 'tuning' the algorithm. This has a clear parallel with the first participant's use of uncertainty as a way of providing something to 'tame'. Their method is a clear depiction of reflection in meta-creative action, as the algorithm functions as an encapsulated system within which they search using an iterative process of refinement. To get to this point, the participant has engaged in a process of developing a meta-creative strategy. Without a clear picture of the full range of sonic possibilities capable within the system, the participant uses this to their advantage, allowing the system to generate endless compositional material for use in a broader musical context.

Devising parameterised "upstream" control

The 3rd participant was involved in designing the modular synthesiser that was being used for this day of the study, so had familiarity with the system. The participant used a variety of strategies when working with the system, adopting several forms of control, ranging from no use of sources of uncertainty to completely automated algorithmic structures. The participant was clear at the outset in describing how they saw patching as a process:

"Really there is just three things, amplitude, phase and frequency, if you can control those three, you win."

The participant was confident in their capacity to design a patch mentally, and begin working towards a goal. They began by patching with no sound on at all. Simply connecting the modules that would be required for the patch they had in mind. Later in the session the researcher asked specifically about this method. The participant described this as *"planning",* they knew they would need a certain range of modules to achieve their musical intention. They described this process of moving from a patch with a simple set of controls, a *"simple, basic thing"* and working towards *"subverting it"*. The participant had the capacity to envisage a simple musical goal, without hearing the musical content. Which they would then audition, once they believed they had arrived at a premeditated point, and move onto creating interest. At the conclusion of one of the three tasks the participant went into more detail about this distinction, between the simple patch they had in mind, and how to make it interesting:

"I need to make it more humanized, more randomised, but this is where the interested comes for me, from doing randomness in a controlled way."

Here we see the participant identify a similar strategy to participants one and two. Humanizing, tuning and taming all describe a similar process involved in the deployment of multiple automated forms of control within the context of an algorithmic composition. Additionally, the participant described one feature of this simple planning method, as allowing them to build the opportunity for interconnectedness within the patch, to *"automatically apply a relationship between two parts of the mix".* This is through the use of multiplying and dividing both clock and modulation signals and applying them to the existing aspects of the patch. What is most interesting about this approach is that the participant, through their familiarity with working in an algorithmic way would preempt the use of complex algorithmic controls by leaving open potential opportunities, at key points throughout the patch.

Discussion

The process of 'encapsulation' and 'conceptual black-boxing' in patch design is present in the discussions that took place with the participants. But what is most clear is that synthesists use a widely divergent range of conceptual strategies when patching a modular synthesiser. Each participant took a different pathway, even when using the same modular configuration.

During our study we designed a process model, intended to provide detailed insight into the state of the participant as they completed a task. Instead, in this small study the method served as means by which to discuss with the participant about their strategies. As the participant arrived at a junction in their patch, for example a moment where they had 'finished' a sound, the participant would begin to discuss their decision-making process, using the dimensions and a means to communicate their strategies. This was an unintended result of the use of this model, but it provided a wealth of dialogue with the participant, much of which was useful in identifying their creative strategies, in this way the method resulted in a semi-structured interview.

Conclusion

In this paper we have presented and analysed some initial findings from a practice-based study into how modular synthesists manage complexity in their algorithmic designs. The three synthesists studied used reflective practice to describe their working methods and responded to a novel process model in order to categorise specific aspects of their relationship to their growing designs as they worked. In order to manage complexity in their designs, each synthesist has engaged in what we have termed 'conceptual black-boxing', a cognitive strategy used to hide complexity, and gain broader knowledge about the context of the automated control processes they were employing in their work. Whilst each synthesist worked with a different system architecture, and made highly contrasting music, some core algorithmic strategies were identified throughout the three sessions. Most commonly, each synthesist described a process where-by they mapped or limited the range of possible actions an algorithmic agent was able to make. This is reflected in our model of reflection in meta-creative action. Where we describe the process the synthesist is undertaking when working with an algorithmic composition system. Constantly accessing knowledge about the broader context of their system and by observing the results of algorithms they have black boxed throughout their process.

In future, we plan to continue the iterative process of developing the model, to better understand how we can capture information about composers creative reasoning when working with algorithmic composition systems. Our goal is to develop a method that will help both developers and practitioners understand how to explore the creative depths of algorithmic music making, with a focus on a practice-lead understanding of these tools.

References

A. Newell. and HA Simon, Human problem solving. Englewood Cliffs, NJ: Prentice Hall., 1972.

B. L. Carey. derivations and the performer-developer: co-evolving digital artefacts and human-machine performance practices. PhD thesis, 2016.

Boren, T., & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. IEEE transactions on professional communication, 43(3), 261-278.

Bradley, M. M., & Lang, P. J. (1994a). Measuring emotion: the self-assessment manikin and the semantic differential. Journal of behavior therapy and experimental psychiatry, 25(1), 49-59.

Bradley, M. (1994b). Emotions: Essays on emotion theory. In Emotional memory: A dimensional analysis (pp. 97-134). Lawrence Erlbaum Associates, Hillsdale, NJ.

C.-C. Chen and N. Crilly. From modularity to emergence: a primer on the design and science of complex systems. 2016.

D. A. Schon. The reflective practicioner: how professionals think in action. Basic Books New York, 1983.

Ericsson, K. A., & Simon, H. A. (1984). Protocol analysis: Verbal reports as data. the MIT Press.

Ericsson, K. A. (2006). Protocol analysis and expert thought: Concurrent verbalizations of thinking during experts' performance on representative tasks. The Cambridge handbook of expertise and expert performance, 223-241.

I. Xenakis. Music composition treks. Composers and the Computer, pages 170–92, 1985.

J. McCormack. Evaluating improvisational interfaces.

K. Bjørn and C. Myer. Patch & Tweak - Exploring Modular Synthesis, Bjooks, Denmark, 2018.

Klahr D. Exploring science: The cognition and development of discovery processes. Cambridge, MA: MIT Press; 2000.

K. Halskov. Design research lab (day in the lab series). ACM Interactions, xviii, 4:92–95, 2011.

ModularGrid 2019, accessed 24 February 2019, <http://www.modulargrid.net>.

M. Akrich and B. Latour. A summary of a convenient vocabulary for the semiotics of human and nonhuman assemblies, 1992.

M. D. Mumford, W. A. Baughman, and C. E. Sager. Picking the right material: Cognitive processing skills and their role in creative thought. 2003.

M. Mose Biskjaer and K. Halskov. Decisive constraints as a creative resource in interaction design. Digital Creativity, 25(1):27–61, 2014.

Nielsen, J. (1994). Usability engineering. Elsevier.

T. Gifford, S. Knotts, S. Kalonaris, and J. McCormack. Evaluating improvisational interfaces.