Composing computer generated music, an observational study using IGME: the Interactive Generative Music Environment

Samuel J. Hunt
Creative Technologies Laboratory
UWE Bristol
Samuel.hunt@uwe.ac.uk

Thomas J. Mitchell
Creative Technologies Laboratory
UWE Bristol
Tom.mitchell@uwe.ac.uk

Chris Nash
Creative Technologies Laboratory
UWE Bristol
Chris.nash@uwe.ac.uk

ABSTRACT

Computer composed music remains a novel and challenging problem to solve. Despite an abundance of techniques and systems little research has explored how these might be useful for end-users looking to compose with generative and algorithmic music techniques. User interfaces for generative music systems are often inaccessible to non-programmers and neglect established composition workflow and design paradigms that are familiar to computer-based music composers. We have developed a system called the Interactive Generative Music Environment (IGME) that attempts to bridge the gap between generative music and music sequencing software, through an easy to use score editing interface.

This paper discusses a series of user studies in which users explore generative music composition with IGME. A questionnaire evaluates the user's perception of interacting with generative music and from this provide recommendations for future generative music systems and interfaces.

Author Keywords

Generative Music, User Evaluation Methods, Computer-Supported Creativity, Sequencers and DAWs

CCS Concepts

•**Applied computing** → **Sound and music computing;**
•**Human-centered computing** → *Empirical studies in interaction design; User interface design;*

1. INTRODUCTION

Computational tools for automatically compositing music have been thoroughly explored in literature [1]. However little research has explored using such techniques alongside human composition inside traditional music sequencing software.

The Interactive Generative Music Environment (IGME) is a music sequencer that supports the exploration of generative and algorithmic music techniques. Unlike code or patch-based systems, it provides an easy to use interface for exploring computer generated music techniques, that is built on common music software paradigms. Many existing generative music systems use workflows that are not familiar to non-programmer music composers [3].

This paper discusses the results of 23 users interacting with IGME. With the overall aim of understanding: are the interfaces we have designed suitable for engendering interactive generative music, and what role does computer generated music take alongside human composition?

The paper is broken down as follows. The proceeding section discuses related work, followed by more specific information on the IGME software. Sections 4 and 5 discuss the methodology and findings of the study.

2. BACKGROUND

An impediment of simple generative music systems is that they often fail to form high level structure, and are highly stochastic in nature [12]. Cutting edge machine learning technologies are often presented as tools capable of composing human-esque music autonomously with minimal direction from humans. While solving some of the limitations of computer generated music they mostly focus on replacing the composer. This is seen in existing systems such as Jukedeck [13], Aiva [18], and Melodrive [7].

There are limited examples of empirical user studies that examine the way in which users interact with music composition software. Even less research exists in studying users of generative music system. Where studies have taken place heavy emphasis is placed on evaluating the quality of the generative output, not necessarily the human factors relating to the use of such systems.. Even then such evaluation remains occasional rather than commonplace [2]. In summarising user experience studies from 132 papers from the NIME, SMC and ICMC conferences, Brown, Nash and Mitchell [6] found the composers' perspectives are rarely evaluated. Work by Nevels [17] studied an individual student composing a song using off-the-shelf music software. The study is limited by its small sample size (n = 1) . Nash [14] completed a large scale observational study of tracking software, summarising 1000s of hours of interaction data. Work by Duignan [8] studied 17 music producers and their use of abstraction mechanisms in common music sequencing software.

Bellingham, Holland, and Mulholland [4] observe that *"most existing software requires the user to have a considerable understanding of constructs in either graphical (e.g Max, Pure Data) or text-oriented (e.g. SuperCollider, ChucK, Csound) programming languages: such knowledge requires a significant learning overhead."* The authors additionally note that existing system impose working practices that simply don't relate to existing compositional practice.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

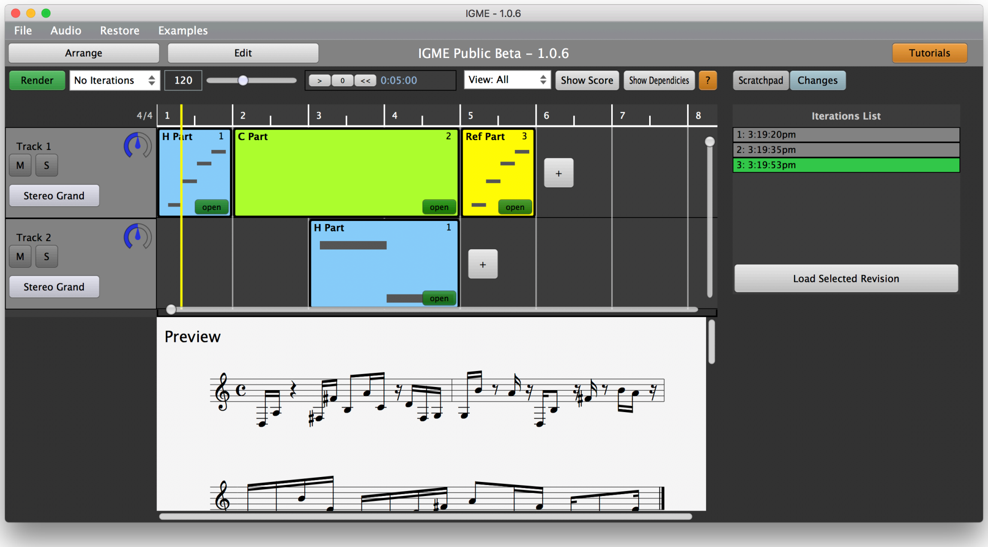

Figure 1: IGME's arrange view where parts are arranged on the timeline

More information around the IGME project is given in the following papers. [12] details the design of IGME and highlights the various design requirements needed for engendering interactive generative music composition. A Cognitive Dimensions of Notation analysis of the software is evaluated in [11]. Whereas previous papers have discussed the design of IGME this paper focuses on the evaluation of IGME with end users.

2.1 Cognitive Dimensions

Green and Petre [10] proposed the Cognitive Dimensions of Notations framework, as an evaluation technique for visual programming environments, interactive devices and non-interactive notations. Nash [15] has adapted this framework for use in designing and analyzing music notations and user interfaces for digital and traditional music practice and study. Bellingham [3] presents similar work, using the dimensions approach for analysing a representative selection of user interfaces for algorithmic composition software. Finally, the cognitive dimensions can more generally be thought of as discussion tools for designers [5].

A table of each dimension and its description is given in Table 1. The descriptions of each dimension are adapted from Nash's [15] work. IGME was designed to 'fit' an ideal profile, whereby features of the software were designed to maximise certain dimensions. More details of this process and overview are given in [11]. We will discuss both the features of IGME and the questions of the user study in reference to the Cognitive Dimensions.

Dimension	Description
1. Visibility	How easy is it to view and find elements of the music during editing?
2. Juxtaposability	How easy is it to compare elements within the music?
3. Hidden Dependencies	How explicit are the relationships between related elements in the notation?
4. Hard Mental Operations	How difficult is the task to work out in your head?
5. Progressive Evaluation	How easy is it to stop and check your progress during editing?
6. Conciseness	How concise is the notation?
7. Provisionality	How easy is it to experiment with ideas?
8. Secondary Notation	How easy is it to make informal notes to capture ideas outside the formal rules of the notation?
9. Consistency	Where aspects of the notation mean similar things, is the similarity clear in the way they appear?
10. Viscosity	Is it easy to go back and make changes?
11. Role Expressiveness	Is it easy to see what each part of the notation means?
12. Premature Commitment	Do edits have to be performed in a prescribed order, requiring you to plan or think ahead?
13. Error Proneness	How easy is it to make annoying mistakes?
14. Closeness of mapping	Does the notation match how you describe the music yourself?
15. Abstraction Management	How can the notation be customised, adapted, or used beyond its intended use?

Table 1: Terms of the Cognitive Dimensions of Music Notations framework used in this paper [15].

3. IGME

IGME is a linear music sequencer that promotes arranging parts [1] (sequences of notes) of music onto a timeline of different tracks (see figure 1), sharing many parallels with existing music software. However the parts are dynamic with their music content composed either by a human operator or configured from a series of generative and algorithmic effects (see figure 2).

The key design features of IGME are as follows:

[1]Parts are metaphorically similar to MIDI clips found in other music software, but overall more complex.

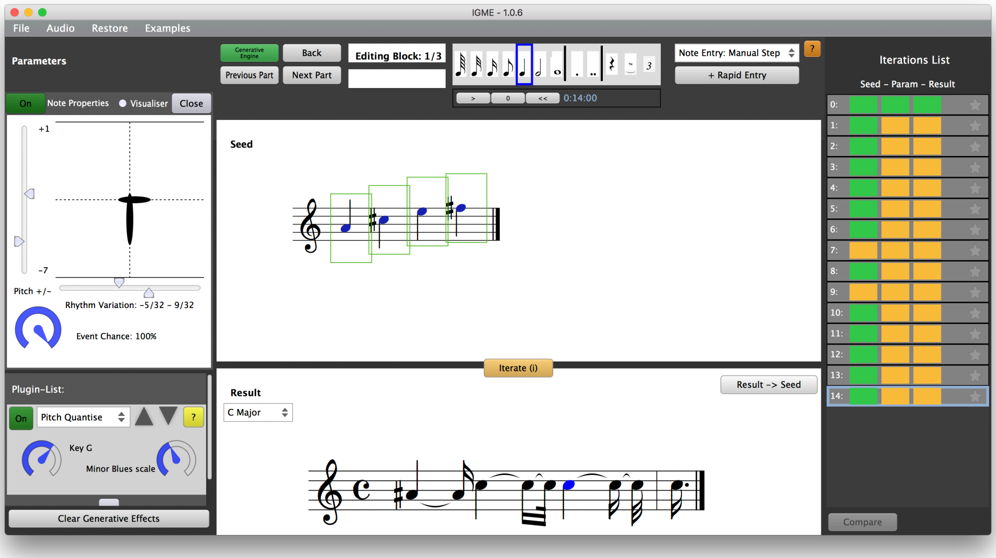

Figure 2: IGME's edit view where individual parts are edited. Left: plugin editor. Centre: note editor. Right: version control system

- Uses existing music sequencing paradigms and design metaphors.
- Provides a full version control system, for tracking edits and iterations.
- Uses graphical widgets, rather than code based interfaces.
- Takes a modular approach to composition, while retaining a linear timeline

Each time an edit is made within IGME it is registered using the inbuilt version control system. When working with stochastic music as a user is free too rapidly create many permutations of the music knowing that if they prefer an earlier version it is easily recalled. This rapid edit-audition cycle contributes to having a high state of flow, a desirable mental state for users engaging with creative exercises such as music [16]. Duignan [9] notes that version control systems are mostly absent from mainstream music software.

Most of the individual components and workflows in IGME borrow heavily from existing music sequencing software. Notes are edited using score notation and the generative effects are set-up similarly to audio plugins. The system is ultimately designed so that a user already familiar with Logic Pro or Sibelius can learn IGME quickly, and immediately experiment with generative effects. Music is arranged on a linear timeline of tracks, whereby the order of the music is explicit. Code or patched based generative music system are often difficult to predict the order of events [15].

Many existing generative music systems require a user to be literate with programming languages, or learn an alien interface [3]. IGME provides access to the underlying generative effects engine through a series of simple graphical interfaces. A downside of the system is the user is currently unable to design their own effects or processes, although there is scope to permit this in future versions.

3.1 Part

A part is composed of 3 distinct sub-components; the seed, parameters, and result. The seed is the music material edited by the user, using a traditional musical score notation editor. The parameters are a series of algorithmic effects that are applied to the seed material, these are contextualised through a series of plugins with various controls for modifying the underlying processes. The result is simply the outcome of the parameters applied to the seed material. The result is the music sequence that is auditioned by the user. Should the user not configure any parameters then the result is a carbon copy of the seed, in this situation we refer to the part as a human composed part. The seed material can also be supplied from a previous parts result (reference part) or by a seed generator (generative process) [12].

When editing the basic music material (seed) the user is free to avoid the formalities of working with bar lines, as such IGME automatically adds bar lines in the result stage. A user is therefore free to edit and arrange sequences with few constraints. Secondly when working with the built in generative effects, the seed material is processed using various effects before becoming the resultant music. This editing paradigm is referred to as the two-stage editing process.

The two-stage editing process materializes the ability to easily work with generative music. The contents of the music (the result) is dynamic, editing the seed is simply a blueprint editor, with the final result computed and subsequently auditioned when an iteration is created. As an example a user can simply select given notes in the editor and apply one of 3 simple individual note parameters, these are chance (the chance of the note happening), pitch range (the range in which the notes pitch will be randomised) and duration (the length the note will be randomly increased or shorted by). As these effects are stochastic each iteration is different each time. A full list of effects is given in [12, 11].

4. METHOD

23 music technology students (aged between 18-25) where invited to participate in a workshop whereby they spent an hour experimenting with the IGME software. Users were given a series of tutorials that they worked through and could spend any remaining time to freely experiment should they wish. After the hour session participants were asked to complete a 10 minute questionnaire evaluating their experi-

ence of using IGME. Of the 23, 9 had used generative music before this study and the other 14 had not. The primary objectives of the study were to assess: are the interfaces we have designed suitable for engendering interactive generative music, and what role does computer generated music take alongside human composition? We additionally wanted to discover what are the key features that should be considered when developing future generative music software.

Many of the questions in the questionnaire are aligned with a dimension of the Cognitive Dimensions of Notation Framework. Others are aligned with the expected uses of generative music. Work by Nash [14] used a similar methodology for comparing experiences of users interacting with music trackers, and evaluating the software under the cognitive dimensions.

5. RESULTS

Each of the question sections was designed to evaluate the novel features and workflows offered by IGME. The features are designed to be generalised and transferable to other music making (specifically generative) software. Each of the key features of IGME (our proposals for making an interface for interactive generative music) was split into a subsection of the questionnaire . Another part of the questionnaire explored what the participants would/would not use generative music for. Finally the creativity section asked questions relating to authorship of computer generated music as well as supporting more open ended qualitative answers.

5.1 Version Control System

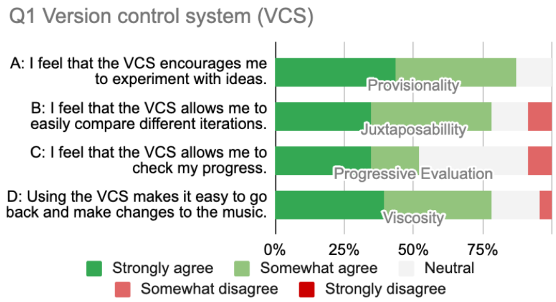

Figure 3: Version control results

Looking at the results for the version control system (VCS), (see figure 3), 87% agreed the VCS supported *provisionality* (A) and 78% agreed it supported *juxtaposabillity* (B). A large proportion of participants did not find the system overly useful for checking progress (progressive evaluation, C): with an open-ended task such as music composition it is not easy to define 'progress'. The version control system enables a user to evaluate progress by loading previous versions and comparing them, however in practice the results suggested this is perhaps not often used. 78% of participants though the VCS made it easy to make changes, therefore having a low level of *viscosity* (D).

5.2 Two Stage Editing

The two stage editing model is designed to increase *provisionality* (A), while reducing *premature commitment, viscosity, and hidden dependencies* (B,C,D respectively). Looking at the results most participants (92%) agreed that provisionality increased (A), premature commitment (B) decreased (78% agree) and viscosity (C) decreased marginally (60% agree). The last question for this section discussed *hidden dependencies*, some participants found that they had

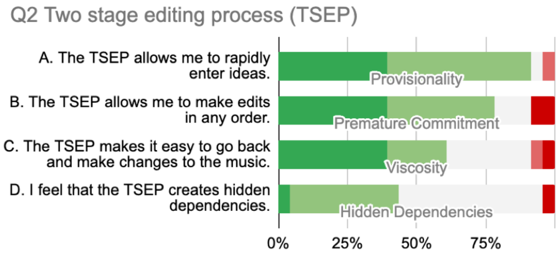

Figure 4: Two stage editing process results

increased (43%), but the results would also suggest that some participants did perhaps not understand the question given half answering neutral. The wording of this question was perhaps to *vague* leading to a grey area for this dimension. Increasing *hidden dependencies* is correlated with an increase in *Visibility*. Many such trade-offs exist within the cognitive dimensions, whereby changing one dimension affects another [16]. Furthermore, the result of each part is dependent on the seed and any generative effects, which increases *hidden dependencies* through a non linear stochastic mapping, between input parameters and resultant music.

5.3 Generative Plugins

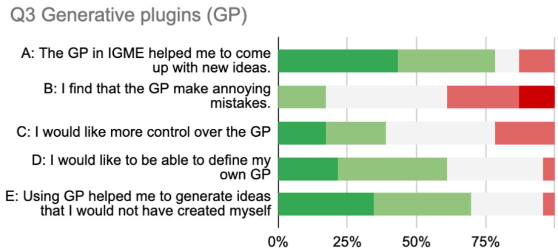

Figure 5: Generative plugins results

The questions in this block were not intentionally aligned with any of the cognitive dimensions but rather intended to evaluate IGME's generative plugins. Looking at the results in figure 5. 78% of participants for question A agreed that the generative plugins helped them *"to come up with ideas"*. Similarly when asked *if the program generated ideas they would not have come up with on their own*(E) 69% agreed this was the case. The majority of participants would like to be able to define their own generative effects (D), but remain generally neutral when asked if *they would like more control over the generative processes* (C). When asked if the software made annoying mistakes (B) answers were mixed. A given musical pattern may sound terrible to one person while being wholly acceptable to another. It is difficult to draw out meaningful conclusions when assessing the musical quality of music given its subjective nature.

5.4 Explicit Parts

Looking at the results for the 4th question (Figure 6), in general 68% of participants agreed that it was easy to distinguish between various parts (A) and 60% agreed it was easy to find the part they were looking for when editing (B). When asked if breaking the music into parts makes it easy to try out new ideas (C) 82% were in agreement.

5.5 IGME System Questions

This question block looked at generalised questions about the IGME software. One shortcoming of the cognitive di-

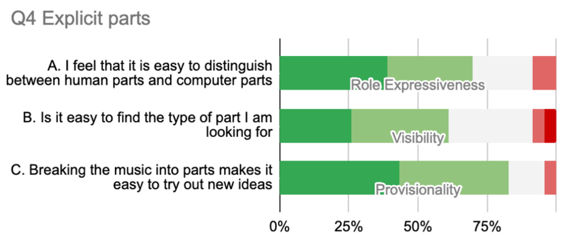

Figure 6: Explicit parts results

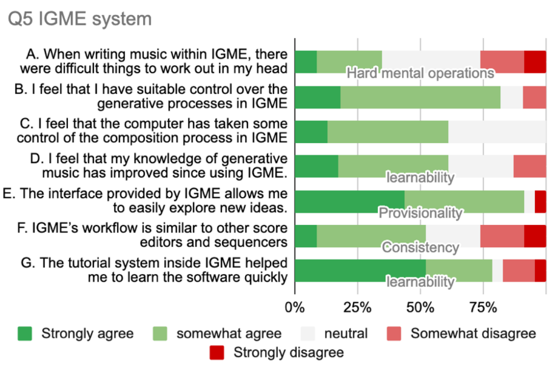

Figure 7: IGME system result

mensions is that they fail to capture *learnability*. Work by Nash [16] addresses this by defining a *virtuosity* dimension, however given the short amount of time each user spent with IGME it is unlikely they would become virtuosic with the software. However we can align questions D and G with our definition of learnability, which is "how easy is it to learn the interface". Finally questions B and C do not map strictly to any dimension.

Analysing the results to this question (Figure 7), question A was balanced in its results, suggesting that if there are *hard mental operations* they are relatively forgiving. 78% of respondents for question B felt that they had suitable control over the generative processes, this contrasts with a similar question about control in Q3-C whereby users where more neutral, suggesting therefore that there is enough control but users would like more. Question C asked if they felt some of the control had been handed over to the computer, the results indicate that users indeed thought this was the case with 60% in agreement and the remainder neutral.

In terms of learnability, the majority (60%) agreed their knowledge of generative music had improved (D), and that IGME's inbuilt tutorial system (G) helped them learn the software (78% agreed). Responses for question F were mixed with 52% agreeing, 21% neutral and 26% in disagreement. IGME was designed to be similar to other music software, borrowing design metaphors and workflows to aid with learnability. However given its unique generative workflows it is also sufficiently different. Finally question E evaluates IGME's ability to explore new ideas generally with almost all participants in agreement 91%. *Provisionality* is one of the more prioritised dimensions for the IGME software.

5.6 Using Generative Music

The questions in this section looked at what a user might use generative music for, and not related to the cognitive dimensions. Results for Question A where promising with 56% strongly agreeing, 30% agreeing and 13% neutral. Responses for question B where less strong, but remained pos-

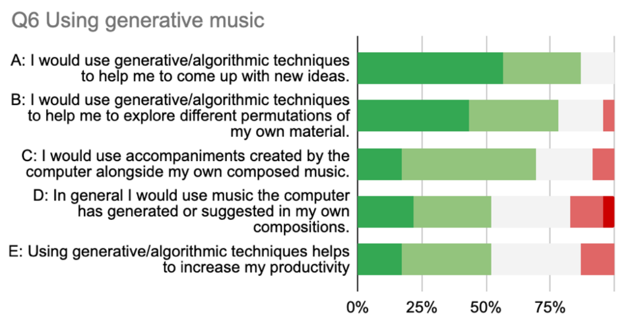

Figure 8: Using generative music results

All	Some	Neutral	Somewhat me	All me
0	8	8	6	1

Table 2: Q. Having used IGME, how much of the musical creativity do you attribute to the computer?

tive with 78% agreeing overall. Overall 69% agreed with Question C, slightly more participants answered neutral and disagree than with other questions in this section. Responses for question D were more polarised, with a 52/17% split between agree and disagree.

Overall it might be suggested that generative music is best suited as a catalyst for composition. Whereby it is used to influence and motivate an individual rather than replace them. The contrast between the strongly agrees in questions A and B and the more neutral and disagrees in question D somewhat emphases this. Participants would use generative music to come up with and alter ideas but not necessarily directly use them. Furthermore the results for question E are not conclusive, generative music can be used to compose music, but can also take many iterations to make something useful which can be a hindrance to productivity.

5.7 Creativity

When asked *"Having used IGME, how much of the musical creativity do you attribute to the computer?"* produced balanced results (Table 2). Notably no one responded *"All"* for computer, suggesting that when using generative music a degree of authorship remains with the user. Another question asked *"Is the authorship of your creative output a concern when using generative techniques?"* with 10/23 answering *yes* (13/23 *no*), participants could optionally explain further. One participant answering yes gave this additional response: *"Once the system begins to be an AI, issues of authorship arise, but for adding what is essentially a curated random chance, then selecting one, authorship and credit belong to the user."*

Following on from this participants were then given a sliding scale from 0-100 in which they gave their response to the following question *"What is the maximum percent of automated creativity you would tolerate?"* (table 3). The average score was 50.1% with a median of 50%. Along with answers to previous questions users would appear to accept some of the computer's creative input. Users of generative software are not just composers of notes but rather composers of parameters and constraints. So despite not being exact about the resultant music, users can express authorship of parameter configurations.

The final two questions asked what *"are the positive and negative aspects of using generative music"* with answers

<20%	20<40%	40<60%	60<80%	<=100%
2	6	7	6	2

Table 3: Q. What is the maximum percent of automated creativity you would tolerate?

given as a textual response. Many respondents discussed that the program created ideas that they would not have come up with themselves. Another mentioned that it can help people that are stuck in a creative rut. Finally one mentioned that *"you don't have to type different sequences in over and over to try things out."*

Many more responses where given for the negative aspects. Many noted that the music created would not be considered their own and they had lost some control over the composition process, although one response noted that *"you aren't forced to use exactly what's generated"*. A few responses observed that it could take a while to create something that did not sound rubbish, with chance playing a key role. Another noted that generative music has a lack of identity compared with music that they would create themselves. One mentioned that it can be too easy, therefore taking away part of the challenge of composition. Music composition can be considered a problem solving activity. In generative music the problem solving activity is perhaps the challenge of configuring parameters to produce a good result, rather that arranging a sequence of notes.

6. CONCLUSIONS

Users of generative music show somewhat restrained views on authorship. We argue that authorship can still be expressed through the arrangement of initial music material, selection of parameters and plugins and finally the curation of material from different output iterations. There is also notably a degree of creative flair attributed to configuring various music creation processes. This research has shown that user's would use generative music to inspire and motivate them to create initial music ideas. Users remained somewhat apprehensive around directly using the output of generative music in their own music.

Participants were in agreement of the merits of the version control system, and we strongly argue such a feature is crucial for a generative music system, and music software in general. The two stage editing process democratises access to generative music. Although unorthodox it presents a genuinely novel attempt at solving the problem of creating an interface that enables access to generative music while retaining similarities to common music sequencing workflows. Participants of the study where mostly in agreement that this feature was positively received.

A shortcoming of this research is that user's only spent an hour each with the software under controlled conditions. Future longitudinal studies with IGME are planned in which users will focus on developing longer pieces of music.

7. ETHICAL STANDARDS

This research involved consenting human participants who received remuneration for their participation in the study. Ethical clearance for the study was granted by the University's ethics board.

8. REFERENCES

[1] C. Ariza. *An Open Design for Computer-Aided Algorithmic Music Composition*. Universal-Publishers, Boca Raton, Florida, 2005.

[2] C. Ariza. The interrogator as critic: The turing test and the evaluation of generative music systems. *Computer Music Journal*, 33(2):48–70, 2009.

[3] M. Bellingham, S. Holland, and P. Mulholland. A cognitive dimensions analysis of interaction design for algorithmic composition software. In *Proceedings of Psychology of Programming Interest Group Annual Conference*. University of Sussex, 2014.

[4] M. Bellingham, S. Holland, and P. Mulholland. Choosers: designing a highly expressive algorithmic music composition system for non-programmers. In *2nd Conference on Computer Simulation of Musical Creativity*, 2017.

[5] A. F. Blackwell, C. Britton, A. Cox, T. R. Green, C. Gurr, G. Kadoda, M. Kutar, M. Loomes, C. L. Nehaniv, M. Petre, et al. Cognitive dimensions of notations: Design tools for cognitive technology. In *International Conference on Cognitive Technology*, pages 325–341. Springer, 2001.

[6] D. Brown, C. Nash, and T. Mitchell. A user experience review of music interaction evaluations. In *17th International Conference on New Interfaces for Musical Expression*, Aalborg University, Copenhagen., May 2017. NIME.

[7] N. Collins. '... there is no reason why it should ever stop': large-scale algorithmic composition. *Journal of creative music systems.*, 3(1), 2018.

[8] M. Duignan. Computer mediated music production: A study of abstraction and activity. *Computer Music Journal*, 34(4):22–33, 2010.

[9] M. Duignan, J. Noble, and R. Biddle. Abstraction and activity in computer-mediated music production. *Computer Music Journal*, 34(4):22–33, 2010.

[10] T. R. G. Green and M. Petre. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.

[11] S. Hunt, T. Mitchell, and C. Nash. A cognitive dimensions approach for the design of an interactive generative score editor. Fourth International Conference on Technologies for Music Notation and ..., 2018.

[12] S. Hunt, C. Nash, and T. Mitchell. Thoughts on interactive generative music composition. In *2nd Conference on Computer Simulation of Musical Creativity*, 2017.

[13] R. Langkjær-Bain. Five ways data is transforming music. *Significance*, 15(1):20–23, 2018.

[14] C. Nash. *Supporting Virtuosity and Flow in Computer Music*. PhD thesis, University Of Cambridge, 2011.

[15] C. Nash. The cognitive dimensions of music notations. In *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2015*. Institut de Recherche en Musicologie, 2015.

[16] C. Nash and A. F. Blackwell. Liveness and flow in notation use. In *NIME*, 2012.

[17] D. L. Nevels. Using music software in the compositional process: A case study of electronic music composition. *Journal of Music, Technology & Education*, 5(3):257–271, 2013.

[18] H. Zulić et al. How ai can change/improve/influence music composition, performance and education: Three case studies. *INSAM Journal of Contemporary Music, Art and Technology*, 1(2):100–114, 2019.