Liveness and Flow in Notation Use

Chris Nash
Centre for Music and Science (CMS),
Faculty of Music, University of Cambridge,
11 West Road, Cambridge CB3 9DP, UK
research@nashnet.co.uk

Alan Blackwell
Rainbow Research Group, Computer Laboratory,
University of Cambridge, William Gates Building,
15 JJ Thomson Avenue, Cambridge CB3 0FD, UK
alan.blackwell@cl.cam.ac.uk

ABSTRACT

This paper presents concepts, models, and empirical findings relating to liveness and flow in the user experience of systems mediated by notation. Results from an extensive two-year field study of over 1,000 sequencer and tracker users, combining interaction logging, user surveys, and a video study, are used to illustrate the properties of notations and interfaces that facilitate greater immersion in musical activities and domains, borrowing concepts from programming to illustrate the role of visual and musical feedback, from the notation and domain respectively. The *Cognitive Dimensions of Notations* framework and Csikszentmihalyi's *flow* theory are combined to demonstrate how non-realtime, notation-mediated interaction can support focused, immersive, energetic, and intrinsically-rewarding musical experiences, and to what extent they are supported in the interfaces of music production software. Users are shown to maintain liveness through a rapid, iterative edit-audition cycle that integrates audio and visual feedback.

Keywords

notation, composition, liveness, flow, feedback, sequencers, DAWs, soundtracking, performance, user studies, programming

1. INTRODUCTION

Notation plays a crucial role in music, especially in musical composition. With technology, however, it becomes possible to record and process sound without the visual medium, using a digital instrument or microphone. Both tools and research have thus increasingly focused on the production and capture of live, realtime musical performances, in which notation plays only an ancillary role in the creation of a piece of music. Leman [11] observes that this avoids the "indirect involvement" associated with interacting with the musical domain through a layer of abstraction, and enables deeper, embodied, and more intimate musical experiences, such as those enjoyed by musicians using conventional acoustic musical instruments.

The dependency on realtime creativity, however, requires virtuosity in performance, limits the scope of what can be expressed live, and makes it difficult (or impossible) to go back and make changes or choose a different creative path, and thus conflicts with the principle of supporting creativity, that tools offer a "low threshold, high ceiling, and wide walls" [15]. By contrast, a notation's decoupling of musical time and editing activity allows for greater flexibility with regards to usability, expression, and the support of experimentation. The remaining challenge concerns how notation can also support an immersive feeling of "direct involvement" in the musical domain.

This paper reviews empirical findings and models concerning perceptions of *liveness* in a user experience, as determined by the availability of domain feedback in notation editing [17]. The example of soundtracking [13] is used to illustrate a UI and notation that supports high liveness through a rapid edit-audition cycle, contributing to conditions that support *flow* [5], an enjoyable mental state characterised by high focus, intrinsic motivation, and total immersion in an activity, often associated with creativity. Comparisons with linear MIDI sequencers and loop/pattern-based DAWs are also detailed.

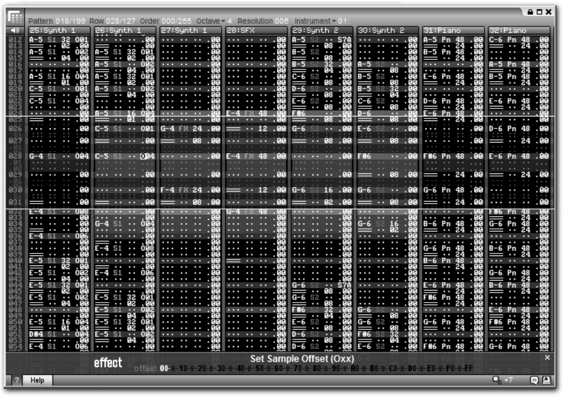

Figure 1. The *reViSiT* tracker plugin.

2. BACKGROUND

This work is based on the results of a two-year user study of real-world sequencer and tracker interaction, which captured data from over 1,000 individuals [12]. Using a tracker UI running as a plugin (Figure 1) within the user's choice of sequencer host, the study combined detailed interaction logging (including program events, user activity, and screen layout), supplemented by user surveys and an in-depth video study, as part of an investigation into virtuosity and flow in music software. The approach and methods used, as well as findings relating to motor skill and virtuosity, are discussed in [13], and are only further detailed here in the context of specific findings.

Trackers are music composition tools based on a text notation (e.g. Figure 1), almost exclusively controlled through the computer's QWERTY keyboard. Music is represented in fixed grids of text (or *patterns*), visually similar to a spreadsheet table – where columns represent separate *tracks* (or *channels*) and the rows represent fixed time slices, like a *step sequencer*. Each cell in the pattern has a fixed number of spaces to specify pitch, instrument, volume (or panning) and one of an extensive set of musical ornaments (or *effects*), for example: **C#5 01 64 D01** starts playback of a note [C#] in octave [5]; instrument [01]; maximum volume [64]; with a slow [01] *diminuendo* [D].

Unlike other music editors (score editors or sequencers) and more like code editors, trackers avoid visual metaphor and traditional music notation abstractions, focusing on a concise textual representation of musical 'source code', for realtime interpretation and playback by a synthesizer. A tracker notation

NIME'12, May 21-23, 2012, University of Michigan, Ann Arbor.

is quickly and efficiently manipulated and navigated by the computer keyboard, for which a significant number of shortcuts and macros are provided. Significantly, the architecture allows a single note, musical part, passage, or the whole piece to be auditioned instantly, using single keystrokes. Our previous work [13] provides a detailed description of trackers, focusing on how flow is supported through the user's development of motor skills and dexterity, upon which the program's edit-audition feedback cycle also depends.

3. LIVENESS IN COMPOSITION

The term "liveness" is increasingly used to describe a subjective sense of intimacy and immediacy in live arts, as experienced between audience and performer [1]. In live electronic music, research highlights the challenge of delivering liveness in the context of disembodied, acousmatic sound (e.g. from a laptop), decoupled from a performer's physical actions [4][8]. When liveness is lacking, the audience feels less a part of the performance, and may find it harder to understand what they hear or should expect, given the (limited) visual feedback.

A similar issue exists in the use of technology to mediate between the artist and their music. Leman [11] talks about the critical role of immediate feedback from an instrument, in providing the performer an understanding of cause and effect, a sense of control, and immersion in the musical domain, the lack of which can lead to disembodied, "indirect involvement" in music, such as Leman associates with the use of notation. However, notation is not only needed by composers to record and sketch their ideas, but also used to abstract details, to consider broader processes and make complexity manageable.

Drawing on similar challenges in software development, Tanimoto's concept of "liveness" in programming [17] can be used to characterise the quality and availability of feedback about a domain (e.g. a program's runtime behaviour, or how the music will sound), provided during editing of the notation (e.g. the code, script, or score). Table 1 provides a description of each level of liveness, with specific examples from both programming and music interaction (from [3]). Notations, and the environments (UIs) used to edit them, may provide a description of end product (be *informative*), define an exact specification of it (be *significant*), have editing actions offer rapid feedback (be *responsive*), or be inseparably and continuously coupled to the product itself (be *live*). Beyond these distinctions, the perception of liveness is also influenced by factors such as ergonomics in the UI, system performance (response times), and the user's ability and interaction style, as discussed in the next sections in the context of tracker and sequencer user experiences. In the latter case, the table also illustrates the divide between the liveness of recording, in sequencers, and the much lower liveness of visual interaction through other sub-devices (arrange view, score, piano roll, etc.).

3.1 Video Study: Feedback Use in Tracking

As part of our investigations into virtuosity and flow, we recorded the interaction of a professional tracker-based film composer, over a 5-hour session, allowing us to study how both the keyboard input and musical feedback were used, and how we should analyse and interpret the raw data in interaction logs from other users (see 2.2).

A tracker user of many years, the composer showed well-developed motor skills and keyboard knowledge that allowed him to remain focused, active, and in flow throughout the session, "touch-typing" music (see [13]). Rather than entering a whole passage of music and then listening to the result, as tracks might be layered in a sequencer, he worked on very short sections (beats or bars), working across all tracks collectively.

Edits were frequently auditioned by quickly cursoring up to just before the edited section, using his right hand, and pressing the *Play from Cursor (F7)* shortcut, with his left. During playback, his left hand remains over the *Stop (F8)* shortcut, ready to jump back into editing if he hears something, while his right hand works the cursors, allowing him to dovetail playback, navigation, and editing, supporting a very rapid edit-audition cycle. When interviewed, he called this "spot-on debugging", in reference to *just-in-time* (or *edit-and-continue*) debugging where a running program can be stopped and edited, and can then continue execution without needing to restart. The frequency with which he moves between editing and listening suggests triggering playback has become a well-learnt, reflexive motor sequence, possibly an instinctive response to the creation of new material in the notation. Playback, while manually triggered, thus becomes closely coupled with editing, enabling a form of Level 3 liveness.

This fine-grained, iterative composition technique enables the composer to quickly sketch and experiment with different ideas ("expand/explore small things"), in what he describes as an intuitive approach to writing music; never consciously planning ahead, but making choices based on what he hears and feels is "natural". Moreover, though he has experience with music performance (including piano tuition), his composition practice is self-taught, implicitly learnt over many years of working with trackers, similarly based on tinkering with the music and notation ("no training; just looking, listening, seeing and understanding the relation"). Such cases of experiential learning have previously been noted when the computer is used to provide progressive feedback during musical creativity [16].

Separate from editing, the composer also spends extended periods (up to an hour) simply listening to the music at length ("macro listening", in his own words). Part of this is to gain a broader perspective of the music, and allow ideas time to incubate [14], but he also cites tiredness that arises from extended periods of focused, energetic editing activity.

Table 1. Levels of liveness in programming and music.

level	informative	significant	responsive	live	description	in programming	in music
1	●				**"ancillary"** in which a visual representation is used as an aid to design, providing a basic graphical representation that is continuously visible, but not executable.	flow chart, UML diagram	composer shorthand, arrange window
2	●	●			**"executable"** in which the system uses a visual representation as an executable specification; offering input to a compiler, but not continuously interpreted.	code editor, script, compiler	score, data list, piano roll, CSound, OpenMusic
3	●	●	●		**"edit-triggered"** in which edits to the representation instantly trigger feedback, allowing users to make rapid actions and (after system response) a chance to correct mistakes.	auto complete, syntax highlights, edit & continue	soundtracking, live coding,
4	●	●	●	●	**"stream-driven"** in which the program is continually active, and where edits directly affect program execution, providing high visibility of the effect of actions.	macro recording	sequencer/DAW recording, live performance, mixing

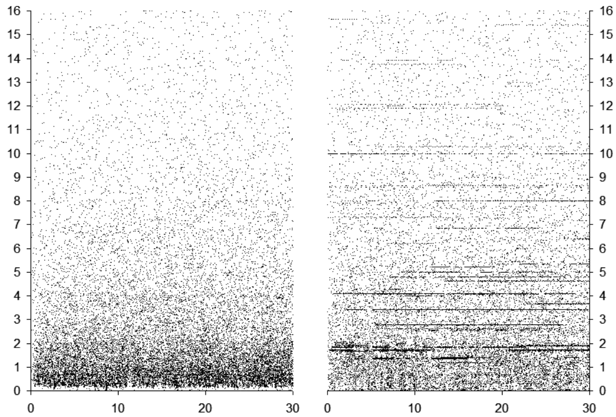

Figure 2. Session offset (mins) vs. playback length (s). First 30 minutes of interaction, for the reViSiT tracker plugin (left) and the host DAW/sequencer (right).

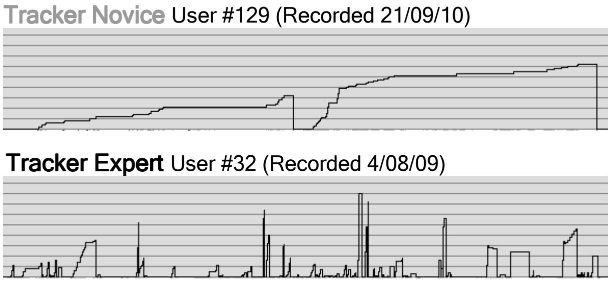

Figure 3. Cumulative data changes plotted against session time, taken from two representative session logs, showing novice and expert interaction styles.

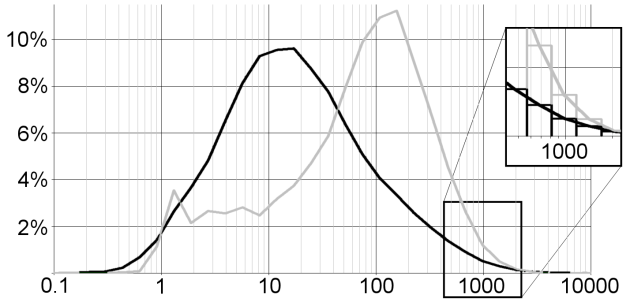

Figure 4. Histogram of editing time (in seconds) between auditions, for novice and expert users; from sessions over 30 minutes, sampled logarithmically (see inset).

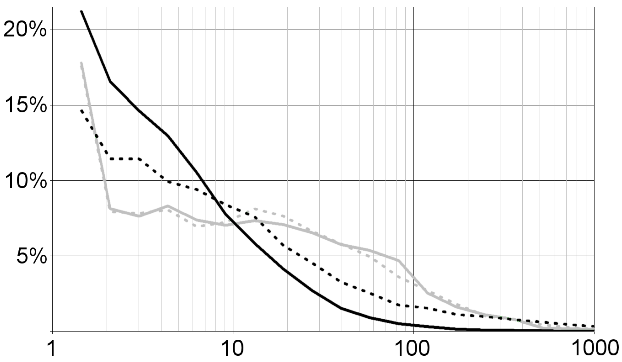

Figure 5. Histogram of edit activity (data changes) between auditions, for novice and expert users; in each case, dotted lines adjust for the increased scope of selection-based edits.

3.2 Measuring Liveness

Turning to playback habits in other users and programs, Figure 2 shows a scatter plot of the first 30 minutes from 1195 sessions recorded by 175 tracker and sequencer users, plotting the duration of the playback (in seconds) against the time it appears in the session (in minutes).[1] As with the video study, other tracker users (left) exhibit a strong tendency towards very short episodes of playback (median = 1.84s), between a beat and a bar in length (assuming 4/4, 120bpm). By comparison, sequencers show a strong tendency towards whole bars and longer phrases – at 2s, 4s, and 8s (1, 2 and 4 bars at 120bpm, 4/4), and also 10s, 20s, 30s, 45s, 60s, and 90s, for projects using digital timecode. Moreover, the plot clearly illustrates how lengths of auditions are set and then retained for long periods of time within sequencers, possibly indicating that the involved process of preparing, targeting, and playing material in the sequencer (e.g. with the mouse) hamper the use of incidental sound feedback during editing seen in tracker interaction. At the same time, longer episodes of playback (indicating broader song playback) were found to be more common in sequencers, which might be explained by the greater and more flexible scope of the sequencer's *arrange window*, compared to the relatively narrow focus of the tracker's *pattern editor* (typically 4 bars).

Unlike the sequencer, detailed information about the internal state of the tracker was available in interaction logs, allowing editing activity between auditions to be analysed. Figure 3 shows excerpts from a representative session profile of two tracker users differing in experience, plotting the cumulative amount of data changed over time, reset on the playback of the pattern or song. Not only is playback used more frequently by the expert, but often for edits of less significance. By contrast, new tracker users, and especially those from a sequencing background, engaged in longer and more extensive visual editing of the notation, before seeking musical feedback.

These trends are mirrored throughout users in the study, as shown in Figure 4, which plots the lengths of editing episodes, as used by tracker experts (Md = 13.2s, Mo = 17.1s, n = 574) and novices (Md = 67.2s, Mo = 155.8s, n = 548). In Figure 5, the number and scope of edits is similarly lower for experts (Md = 2.36 edits, 4.00 total data changes), compared to novices (Md = 5.44 edits, 5.70 total data changes). Thus, although we speculate that expert users are more capable of working longer without the scaffold of musical feedback, which might improve productivity, they choose not to. Rather than relying exclusively on the visual feedback from the notation, tracker experts learn to interlace editing with frequent, short episodes of playback, the effect of which is to greatly improve the liveness of working with the music, allowing sound feedback to guide interaction and creative choices.

4. FLOW IN NOTATION USE

Direct and immediate feedback is a central component of Csikszentmihalyi's flow theory [5], which describes the focused mental state of an individual (or group) completely immersed in an activity (the merging of action and awareness), and thus resonates with musical descriptions of "liveness" (e.g. [1], [4], and [8]). Flow has been observed in both music and programming [3] and, by integrating theories of motivation and skill development, is commonly linked with learning and creativity [5] (see also [14], in the case of musical creativity).

The nine common components of flow (listed in Figure 6) need not all be present for flow to occur (and often interact with each other), but generally describe a an intrinsically-rewarding

[1] See Nash and Blackwell (2011) for a histogram of this data.

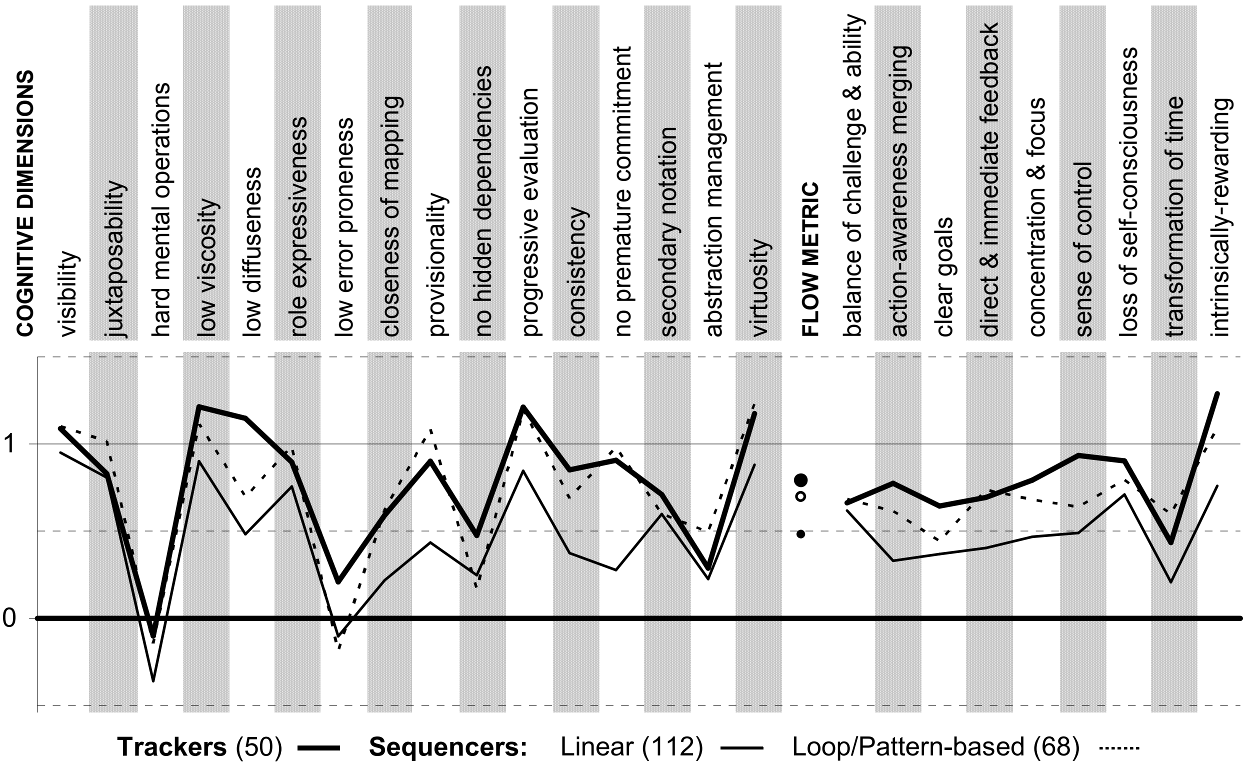

Figure 6. Cognitive dimensions of notations (left) and flow component (right) profiles for music software, based on mean survey response (sample size in brackets), scored on a Likert scale (-2/+2; strongly disagree/agree), for trackers (bold; including *reViSiT*, *Renoise*), linear sequencers (solid; including *Cubase*, *Nuendo*, *REAPER*, *SONAR*), and loop-or pattern-based sequencers (dashed; including *Ableton Live*, *FL Studio*).

activity that provides a suitable level of challenge, given an individual's ability (mediating between boredom and anxiety), allowing them to focus on the task and forget both themselves (ego) and the outside world (social pressures, sense of time).

As detailed in previous work [13], flow was observed in the tracker interaction captured in our video study, in which the composer demonstrated sustained focus and concentration, a distorted perception of time, a loss of self-conscious, and the ability to know exactly how to achieve his goals, in a challenge pursued for no external reward (composing for himself). Analysis of broader samples of interaction logs from other users also revealed indications of flow in tracker interaction, explored by looking at the user's performance, changing focus, and use of feedback. However, some aspects of flow experience are subjective, and harder to analyse in relation to notation use.

4.1 Measuring Flow

In the final months of the study, an online survey was issued to gauge users' subjective experience of the tracker notation and interface, in comparison to their experiences of a sequencer of their choice (e.g. the host sequencer).

The first section of the questionnaire presented two blocks of statements describing the 9 components of flow, which the user was instructed to score on a 5-point Likert agree-disagree scale, with respect to how they perceived them in the user experience. This section and the flow statements were adapted from the Dispositional Flow Scale-2 (DFS-2), a psychometric test to quantitatively measure flow in a given activity [10]. A second section similarly scores sixteen statements corresponding to *cognitive dimensions of the notation* [9], enabling comparisons and correlations to be made between flow components and properties of the notation. These statements were adapted from the *Cognitive Dimensions Questionnaire Optimised for Users* [2], presenting each dimension in language that can be interpreted by end-users. Questions were presented twice; respectively for the tracker and for the user's chosen sequencer, which they selected from a list of 12 popular tools or specified themselves.

Figure 6 shows the cognitive dimensions and flow profiles reported by participants, for trackers and sequencers. When broken down by product, one of two distinct profiles were exhibited by sequencers, depending on whether their main UI was based around the traditional linear timeline and recording (such as *Cubase*, *Nuendo*, *REAPER* and *SONAR*) or on the triggering of loops or short patterns (such as *Ableton Live* and *FL Studio*). Significantly, the latter variety exhibited more favourable dimensions with respect to both the cognitive dimensions of the notation and subjective experience of flow, most notably with respect to *provisionality* (the opportunity to sketch or play with ideas provisionally), *premature commitment* (being forced to think ahead and commit to decisions early) and *progressive evaluation* (the opportunity to check your work as you go along). Such differences between these types of sequencers can be attributed to their representation of time – linear sequencers show music in the order it will be heard ("eager linearisation"), whereas software based on short patterns or loops allow greater flexibility and provisionality in the order they are to be played ("delayed linearisation") [6].

The closest correlate of liveness, *progressive evaluation*, can be also explained by the narrower editing and playback focus on shorter passages of music in these programs. Trackers, which are similarly pattern based, also exhibit favourable profiles, additionally benefitting from the focus and level of control facilitated by the use of a concise text-based notation, single editing context (contrasting sequencers' multiple sub-devices, often across separate floating windows), and support for the development of motor skill and virtuosity [13].

	Program Experience	**COGNITIVE DIMENSIONS**	virtuosity	visibility	progressive evaluation	low viscosity	role expressiveness	consistency	low diffuseness	no premature commitment	provisionality	closeness of mapping	juxtaposability	secondary notation	abstraction management	low error proneness	no hidden dependencies	no hard mental operations
FLOW METRIC	.33		**.43**	**.53**	**.51**	**.46**	**.46**	**.45**	**.43**	**.40**	**.38**	**.36**	**.34**	**.32**	**.25**	**.12**	**.10**	**.09**
intrinsically-rewarding	.31		.48	.54	.57	.48	.44	.39	.36	.34	.41	.36	.29	.29	.19	.17	.16	.13
sense of control	.34		.37	.54	.44	.45	.41	.43	.46	.26	.33	.33	.44	.28	.15	.24	.11	.07
action-awareness merging	.25		.26	.47	.36	.31	.30	.31	.31	.42	.38	.34	.31	.25	.21	.07	.06	.00
concentration & focus	.29		.30	.35	.42	.38	.32	.37	.32	.30	.27	.27	.25	.23	.14	.16	.11	.04
direct & immediate feedback	.18		.22	.28	.32	.31	.37	.34	.23	.31	.25	.29	.20	.23	.19	.10	.11	.09
clear goals	.15		.20	.24	.23	.27	.28	.24	.20	.24	.15	.20	.11	.31	.29	.05	.14	.11
balance of challenge & ability	.21		.42	.32	.21	.25	.22	.19	.24	.18	.14	.22	.17	.19	.09	.02	-.01	-.06
transformation of time	.09		.19	.12	.20	.09	.09	.13	.20	.09	.14	.07	.11	-.02	.20	-.08	-.07	.00
loss of self-consciousness	.09		.08	.22	.17	.14	.20	.16	.14	.14	.11	.04	.09	.10	-.04	-.01	-.01	.12

Table 2. Correlation matrix between flow components and the cognitive dimensions of notation (n=245).

To investigate how specific properties of a notation impact flow in the user experience, Table 2 presents a correlation matrix from 245 survey responses showing the correlations between sixteen cognitive dimensions[2] and the nine components of flow.

The strongest correlation lies between *progressive evaluation* and the *intrinsic reward* present in an activity (r = .57), further iterating the importance of liveness, wherein musical (domain) feedback on the user's progress acts as a source of motivation and makes the activity more enjoyable. Visual feedback (*visibility*) is similarly important in flow (r = .53), not only contributing to *intrinsic reward* (r = .54), but also enabling a greater *sense of control* (r = .54) and immersion in the activity (*action-awareness merging*, r = .47).

To account for the internal interactions between cognitive dimensions and identify the key dimensions that contribute to perceptions of flow in the user experience, the survey data was subjected to multiple regression analysis. Table 3 presents models produced by a stepwise regression analysis, using forward selection with Mallows' Cp as a stopping rule to reduce the likelihood of overfitting. Individual factors are tested using a student's t-test (95% and 99% confidence levels are highlighted), and the model tested using analysis of variance (ANOVA). The model showed a strong goodness-of-fit, with R^2 and *adjusted* R^2 figures suggesting that between six and eight cognitive dimensions of the notation account for almost half the variation in flow indicated by users ($p < .001$).

Three cognitive dimensions stand out as highly significant in the contexts studied: *visibility*, *progressive evaluation*, and *consistency*, again highlighting the importance of feedback, from both the visual notation (UI) and musical domain (audio). Through greater liveness (Section 2), the causal effects of user actions are easily perceived, contributing to a *sense of control*, but also allowing greater *concentration & focus* to rest as much with the actual music, as the abstract visual representation. Both dimensions are fundamental to the user's understanding of what is going on in the program, and their music.

The effective transparency of the notation enabled by fast domain feedback also improves the learnability of a program, where users can experiment with commands and features to understand their function. In this respect, *consistency* in the representations used throughout a program similarly aids learning, allowing users to transfer knowledge and expertise from one part of the UI to another, and simplifying the overall handling of the system.

In general, the dimensions of the notation that correlate most strongly with flow and its components correspond to those prominent in the profiles of most music software (Figure 6). However, the strongest predictors of flow – those associated with visual feedback (*visibility*), domain feedback (*progressive evaluation*), and the support for rapid editing and sketching (*viscosity*) – are markedly stronger in the notations of trackers and pattern/loop-based sequencers, compared to traditional linear sequencers, leading to greater sense of immersion and flow in these programs.

Table 3. Flow model based on Cognitive Dimensions. Regression statistics, terms, and ANOVA results modelling flow using forward selection stepwise regression. 95% (and 99%) significance levels are highlighted in p-values for the model and its terms, where each term is also highlighted according to its significance in studies of other samples [12].

Regression	
Multiple R	.702
R^2	.492
Adjusted R^2	**.483**
Standard Error	.366
Observations	423
Mallows *Cp*	11.820

ANOVA	Reg.	Res.	Total
df	8	414	422
SS	53.66	55.33	108.99
MS	6.707	0.134	
F	50.19		
p	< .001		

Terms	**Beta**	Coeff.	Std. Err.	t Stat	p-value	95% CI
Intercept	.000	.039	.032	1.220	.223	-.024
Visibility	**.188**	.116	.027	4.335	< .001	.064
Progressive Eval.	**.169**	.104	.027	3.796	< .001	.050
Consistency	**.173**	.122	.028	4.374	< .001	.067
Virtuosity	**.148**	.084	.023	3.628	< .001	.038
Abstraction Mgmt.	**.121**	.057	.018	3.226	.001	.022
Viscosity	**.132**	.083	.026	3.158	.002	.031
Prem. Commitment	**.084**	.043	.020	2.104	.036	.003
Role Express.	**.104**	.062	.025	2.495	.013	.013

[2] An additional *virtuosity* dimension is introduced in an effort to assess 'learnability' properties of a notation, not captured by the original framework [7]. Here, it is tested using the statement "With time, I think I could become a virtuoso user of the system", corresponding to how easy a user believes a notation is to learn and master, and correlating with flow's *balance of challenge and ability* (r = .48).

5. CONCLUSION & FUTURE WORK

This paper has reviewed concepts and presented empirical methods and findings relevant to the evaluation and support of liveness and flow in computer music interaction. Focusing on notation-based composition rather than live performance, this research defines liveness with respect to the availability and quality of feedback from the domain [17] (e.g. sound [3]), and used a large study of tracker and sequencer users to investigate the use of feedback and its relation to flow [5] in real-world interaction with music production and composition software.

Findings suggest that the limitations of linear timeline UIs and the tape recorder metaphor of play-record-rewind, used by traditional sequencers, reduce the availability and quality of feedback, lowering liveness in interaction with supporting visual notations. In practice, this limitation is offset by shifting the focus of interaction to dedicated realtime hardware (instruments, controllers, and control surfaces), which supports episodes of Level 4 ("stream-driven") liveness, but which also places restrictions on the virtuosity, scope, and provisionality of creative expression. By contrast, trackers (together with pattern or loop-based sequencers) demonstrate how rapid edit-audition feedback cycles can be used to improve the liveness of notation-mediated interaction, facilitated by the development of motor skills using the keyboard [13], approaching Level 3 liveness. These programs, by narrowing the scope of editing to shorter excerpts of music, also make it easier for users to maintain focus and a sense of control, further facilitating flow.

Despite its historically-central role for both performers and composers, notation has received limited attention from digital music research. In working towards more 'live' and immersive interactions with notation, this paper used the cognitive dimensions of notations framework [2][9] to explore notational factors that affect flow, reiterating the importance of visual feedback (to support a sense of control) and domain (musical) feedback (to allow users to see the emerging product of their efforts), as well as the importance of learning and virtuosity (see also [13]). Profiles and models generated by user surveys can be used to highlight usability issues and inform the interaction design in these products, and we hope to use the methodology to further explore trends in notations used in both music and other forms of creative design.

Finally, to complement the empirical approach taken in this paper, we look towards the development of a theoretical framework that accounts for feedback and liveness in the modelling of music interaction within creative systems. Figure 7 uses the *systems of musical flow* framework [12], developed in parallel with our studies of users, which models properties of liveness and flow in notation use, based on the network of feedback loops within a user experience (pictured here, for sequencers and trackers). For further details, see [14] (and [3]).

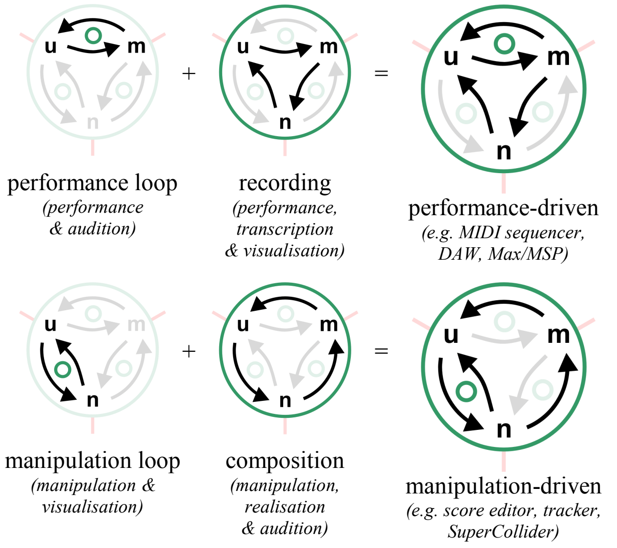
u
m
n
+
=
performance loop
(performance & audition)
recording
(performance, transcription & visualisation)
performance-driven
(e.g. MIDI sequencer, DAW, Max/MSP)
manipulation loop
(manipulation & visualisation)
composition
(manipulation, realisation & audition)
manipulation-driven
(e.g. score editor, tracker, SuperCollider)

Figure 7. Flow in music systems, modelled as feedback loops between the user, notation, and music domain. [14]

ACKNOWLEDGMENTS

This research was part-funded by the Harold Hyam Wingate Foundation. We'd also like to express our sincere gratitude to the participants in our study and the wider tracking and sequencing user communities whose openness and support made our research possible. Special thanks are also due to composer, Maarten van Strien, for providing a window into his studio; and to the people at Yamaha, Steinberg, and Cakewalk, for their valued input, support and assistance during the study.

6. REFERENCES

[1] Auslander, P. *Liveness in a Mediatized Culture*. Routledge, Abingdon, UK, 1999.

[2] Blackwell, A. and Green, T. A Cognitive Dimensions questionnaire optimised for users. In *Proceedings of the PPIG 2000*, 137-152.

[3] Church, L., Nash, C., and Blackwell, A. Liveness in Notation Use: From Music to Programming. In *Proceedings of PPIG 2010*, 2-11.

[4] Croft, J. Theses on liveness. In *Organised Sound*, 12(1): 59–66, Cambridge University Press, 2007.

[5] Csikszentmihalyi, M. *Creativity: Flow and the Psychology of Discovery and Invention*. NY: Harper Perennial, 1996.

[6] Duignan, M. *Computer mediated music production: A study of abstraction and activity*. PhD thesis. NZ: Victoria University of Wellington, 2007.

[7] Elliot, G.J., Jones, E., and Barker, P. A grounded theory approach to modelling learnability of hypermedia authoring tools. In *Interacting with Computers*, 14:547-574. Elsevier Science, 2002.

[8] Emmerson, S. *Living Electronic Music*. Ashgate, 2007.

[9] Green, T. and Petre, M. Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework. In *Journal of Visual Languages and Computing*. Academic Press, 7: 131-174, 1996.

[10] Jackson, S., and Eklund, R. Assessing flow in physical activity: The Flow State Scale-2 and Dispositional Flow Scale-2. In *Journal of Sport and Exercise Psychology*, 24:133-150, 2002.

[11] Leman, M. Embodied Music Cognition and Mediation Technology. MIT Press, Cambridge, MA, 2008.

[12] Nash, C. *Supporting Virtuosity and Flow in Computer Music*. PhD Thesis. University of Cambridge, 2011. Available from Summer, 2012 at: http://phd.nashnet.co.uk

[13] Nash, C. and Blackwell, A. Tracking Virtuosity and Flow in Computer Music. In *Proceedings of ICMC 2011*, International Computer Music Association, 575-582.

[14] Nash, C. and Blackwell, A. Flow of creative interaction with digital notations. In *Oxford Handbook of Interactive Audio (in press)*. Oxford, UK: Oxford Uni. Press, 2012.

[15] Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T and Eisenberg, M. Design Principles for Tools to Support Creative Thinking. In *NSF Creativity Support Tools Workshop*, 25-36, 2005.

[16] Scripp, L., Meyaard, J., and Davidson, L. Discerning Musical Development: Using Computers to Discover What We Know. In *Journal of Aesthetic Education*, 22(1):75-88. University of Illinois Press, 1988.

[17] Tanimoto, S. VIVA: A Visual Language for Image Processing. In Journal of Visual Languages and Computing. Academic Press. pp. 127-139, 1990.