

# MANUSCORE: MUSIC NOTATION-BASED COMPUTER ASSISTED COMPOSITION

*James B. Maxwell, Arne Eigenfeldt, Philippe Pasquier*

Simon Fraser University  
SCA/SIAT, Vancouver, Canada  
jbmawel@sfu.ca

## ABSTRACT

ManuScore is a music notation-based, interactive music composition application, backed by a cognitively-inspired music learning and generation system. In this paper we outline its various functions, describe an applied composition study using the software, and give results from a study of listener evaluation of the music composed during the composition study. The listener study was conducted at a chamber music concert featuring a mixed programme of human-composed, machine-composed, and computer-assisted works.

## 1. INTRODUCTION

Otto Laske’s notion of “composition theory” [6] focused on three fundamental principles: 1) *competence*: knowledge of the materials and syntax of music, required for the conception of musical ideas; 2) *performance*: the practical application of accumulated musical knowledge (competence) to create musical forms; and 3) the *task environment*: the field of action in which performance draws on competence for the invention of musical works. In the context of Computer-Assisted Composition (CAC), the task environment is embodied by a computer and its hardware/software. When this role is assigned to something as pliable as computer software, composers are suddenly given the capacity to tailor the task environment to their particular needs, in a manner not previously possible. Laske felt that, in a computer-based task environment, composers could access the virtual music of their imaginations in a manner unbounded by musical traditions. He identified the process of conception, design, implementation, and production of the task environment, and its iterative development throughout the compositional process, as the “compositional life cycle” [9]. It was Laske’s feeling that software developed through a compositional life cycle could gradually begin to embody the musical knowledge of the composer in an explicit, analyzable, and extensible way.

Early CAC tools like Koenig’s PROJECT systems [8], or Truax’s POD systems [19], took a top-down approach to CAC, in which high-level concepts were expressed parametrically, or graphically, and the software was responsible for generating numerical representations, or electronically synthesized performances, of musical output. Two

important observations can be made about such systems: 1) They are not corpus-based, and thus will not generally maintain an explicit connection with the user’s musical past, and 2) They deal with musical concepts at a high level of abstraction, and thus introduce significant nonlinearities into the compositional process—i.e., by substituting numerical representations for sounds, they separate composition from the act of listening (an idea that Laske very much supported, believing that such a division could lead to “unbounded” musical invention).

The vast majority of commercial CAC packages (conventional Digital-Audio Workstations (DAWs) and MIDI sequencers) are essentially bottom-up systems, which fulfill the basic tasks of recording and manipulating musical performances, or transcribing and electronically ‘performing’ musical scores. Although applications of this type are often equipped with extensive feature-sets, directed toward simplifying and streamlining the workflow for such tasks, their fundamental purpose is to *record*.

Making up the middle-ground, there are an increasing variety of CAC tools which propose different (and often quite novel) forms of musical representation, and introduce varying degrees of interactivity into the compositional process ([23, 20, 13] etc.). Among this class one could also include the increasing variety of music programming languages, graphical or otherwise ([1, 17, 21], etc.), which offer a potentially infinite variety of CAC *tools-to-be*, and propose potentially infinite mixtures of top-down/bottom-up control.

For the bottom-up tools, competence and performance are essentially unchanged from the traditional requirements of music-theoretical knowledge, instrumental performance ability, skill in instrumental music transcription, and so on. With the top-down tools, the demand placed on competence is shifted (and potentially increased) by the emphasis on abstraction, while performance becomes focused on the interpretation of numerical representations of musical materials [7], and/or on the comprehension of metaphorized descriptions of musical concepts: “density”, “timbral trajectory”, and so on [19].

## 2. MOTIVATIONS BEHIND MANUSCORE

Our initial goal in designing ManuScore was to create a music notation-based CAC tool for music-literate com-

posers, who might already possess a developed musical language and bottom-up compositional practice, but were interested in exploring a top-down interaction with their musical ideas. In contrast to Laske’s goal of freeing composers from musical tradition, we wanted the system to acknowledge the user’s existing musical practice, so that working in ManuScore need not impose any dramatic change in a composer’s musical language or compositional output. The intention was to create a task environment to augment a composer’s practice, not necessarily to dramatically alter it, and certainly not to completely automate it. In this sense, the system could be aligned with Cope’s CUE software [5], which draws from a music *recombination* database to offer “continuations” and developments of musical ideas introduced by the user.

Adding to this general conception of a non-interfering, interactive CAC tool with corpus-based generation, we were also interested in the notion of ‘object-oriented’ composition [15]. Our conception of object-orientation focuses on the notational aspects of musical ideas. That is, we are interested in what can be captured in a musical score, what the various structures on the score’s surface represent to the composer (i.e., what is their musical ‘objecthood’), and how these structures might ‘inherit’ from one another in the developing composition. In this sense, an *object* in ManuScore is somewhat analogous to a “gestalt” in the music perception and cognition literature [18]; i.e., it is an identifiable, holistic item, or concept. To whatever degree possible, we wanted to help composers explore musical ideas as gestalts, and to represent them accordingly in their compositional task environment. This approach connects ManuScore to programs like PatchWork (or PWGL [10]) and OpenMusic [1], which also help composers interact directly with musical concepts, though our focus on notational elements (i.e., leaving aside numerical operations) in the user interface clearly sets ManuScore apart.

The generative capabilities of ManuScore, in its current version, are focused on the notion of ‘continuation’: i.e., of extending musical fragments introduced by the user. Generation is also currently monophonic. However, our goal with ManuScore is to implement real-time, interactive generation, so that musical ideas may also be explored through listening and improvisation, not just through manipulation of scored musical objects.

### 3. MANUSCORE DESIGN & FEATURES

In designing the Graphical User Interface (GUI) for ManuScore we wanted to emulate a ‘pencil and paper’ workflow, while maintaining a balance between power and flexibility. Wherever possible, we chose to utilize the standard ARROW UP/DOWN/LEFT and RIGHT keys for moving objects, selecting accidentals, toggling articulation markings, and so on, requiring the user to remember only a limited set of possible interactions when learning the software.

#### 3.1. An ‘Open’ Musical Space

At launch, the ManuScore GUI presents the user with an empty space—a ‘blank canvas’, so to speak. The background displays faint vertical guides, which act as a temporal grid for entering musical events. The grid does not strictly follow the conventions of musical time signatures. Rather, it acts as a *visual* guide to subdivide the musical space, providing a similar function to the grid systems found in graphics and drawing software packages, and provides “snapping” functionality to assist the user in entering rhythmically precise material. Objects can be moved independently of the grid, so that events can be placed at any location in musical time.

At the top of the score window, “Metric Markers” can be inserted, allowing the temporal grid to be subdivided in arbitrary ways. It is worth noting that this division is strictly graphical, and imposes no formal restrictions on the music itself. The numbers in each marker indicate a grouping/subdivision of time. The top two numbers are conceptually analogous to a conventional time signature, while the bottom number indicates the number of “beat divisions” used for object snapping (and thus can create any *n*-tuplet subdivision). Figure 1 shows a sample score with two Metric Markers added. It will be noted that the markers only alter the grid for the rhythmic space *following* the marker’s position. This can be seen in Figure 1, where what *appears* to be a  $\frac{4}{4}$  time signature is cut short by a  $\frac{7}{8}$  signature, inserted part-way through the first measure. In conventional notation software, replicating the musical meaning of this structure would require the user to completely redefine the metrical notation of the music (and in most cases, to delete and re-enter the musical passage). However, because the temporal grid in ManuScore is essentially independent from the contents of the staves, this sort of structure can be created at any time, without altering the existing musical material. In this sense, ManuScore’s rhythmic representation offers a ‘smooth’ space for composition, as opposed to the highly structured metrical space of conventional notation software.

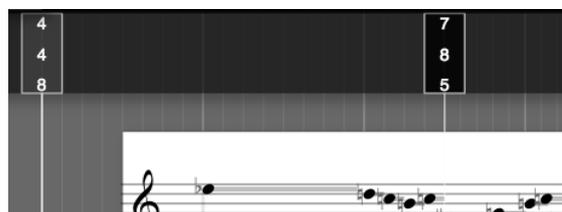
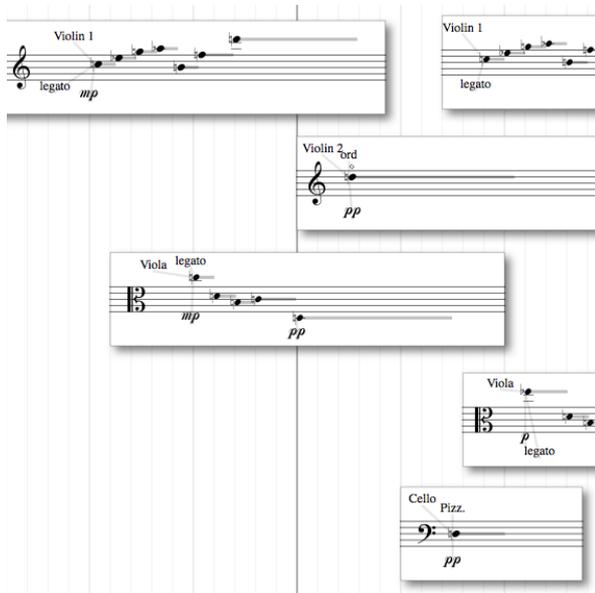


Figure 1. Metric Markers in ManuScore.

The method for adding staves in ManuScore was directly inspired by Stravinsky’s self-designed and patented *stylus*, which he used for drawing staves by hand [16]. The stylus allowed him to format manuscript to his needs, and also allowed him to insert additional musical parts into existing scores; borrowing Laske’s term, one could say that his stylus arose from a “compositional life-cycle.” In ManuScore, staves of arbitrary length can be created at

any position in the score, helping to promote the sense of an open musical space. This method of adding staves leads to a ‘cut-away’ style, in which instruments only appear when they are playing [14]. It is our feeling that the cut-away style offers a visual representation analogous to listening, while at the same time supporting our conception of gestalt-based, ‘object-oriented’ composition.



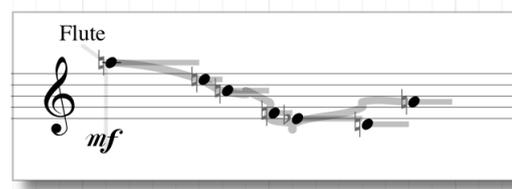
**Figure 2.** The “cut-away” score style supports the notion of musical *objects*.

### 3.2. Note Entry in ManuScore

Once a staff has been created, single notes can be entered in three ways: 1) Typing *n* and clicking on a staff, 2) Using the Step-Entry Cursor (*e* key), and 3) Using a MIDI keyboard. It may have already been noted that ManuScore attaches accidentals to all notes, following the practice used by composers like Witold Lutoslawski. In ManuScore, this is a direct result of the inherent lack of conventional bar lines, which traditionally serve to nullify previously written accidentals. Once a note is created, the accidental can be toggled, by holding the *COMMAND* key and using the *ARROW UP/DOWN* keys.

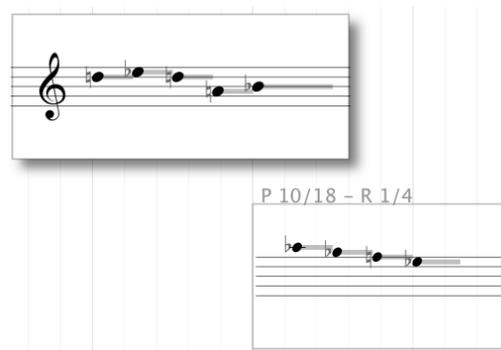
Material can also be entered in complete ‘gestures’, using the Gesture Tool (*g* key). This tool allows the user to draw a free-hand line on the staff, which is subsequently interpreted by the underlying generative system. ManuScore is backed by our “Closure-based Cueing Model” (CbCM), which is used for gesture interpretation. The process operates by using the CbCM to infer the *pitch contour* of the gesture. For each transition in the inferred “Schema” pattern [11] (i.e., contour), the algorithm selects those pitches which best approximate the position of the gesture line. ManuScore’s interpretation of a drawn gesture is shown in Figure 3. It is worth noting that, although most of the interpreted pitches follow the line

quite tightly, the low D represents a ‘best-attempt’ of the CbCM, given its training. Its failure to follow the gesture line indicates that, in the given musical context, the system did not have a learned transition that could better approximate the path of the line. For a detailed discussion of the CbCM, see [11].



**Figure 3.** CbCM interpretation of a Gesture Line.

Finally, new material can also be generated directly by the CbCM, as a continuation of a given musical context. When a note on a staff is selected, the CbCM can generate a continuation from that note, and render the continuation on a separate staff. An example of a CbCM continuation is shown in Figure 4. In the example, the top staff is the user-defined “context” and the lower staff is the generated continuation. The text above the generation “P 10/18 - R 1/4” indicates that the CbCM generated 18 pitch continuations, the 10th of which has been selected, and 4 rhythmic continuations, the 1st of which has been selected. Holding the *OPTION-COMMAND-SHIFT* keys and using the *ARROW UP/DOWN* keys will toggle through the different pitch patterns, while using the *ARROW LEFT/RIGHT* keys will toggle through the different rhythmic patterns.



**Figure 4.** CbCM continuation of a given passage.

### 3.3. Orchestration in ManuScore

Our general goal of ‘openness’ can also be seen in ManuScore’s flexible approach to orchestration. Rather than following the conventional design, in which instruments are created *a priori*, on “tracks” similar to those used in analog tape recorders, ManuScore uses an approach inspired by the practice of composing to “short-score.” When composers work to short-score, they often apply orchestration notes after the fact, assigning instruments to

specific musical gestures directly on the staff. Orchestration in ManuScore follows the same process, as shown in Figure 5. With a correctly configured MIDI system, the instrumental switch from Flute to Viola at F#4 will be played back via MIDI. Instruments can be assigned to notes by typing *i*, clicking at the desired location, and entering the instrument name.

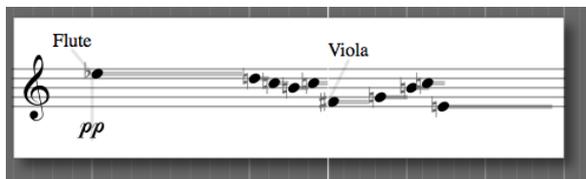


Figure 5. Assigning Instruments in ManuScore.

Users can define a custom library of instruments in the MIDI Setup window. The window has four panels: 1) MIDI Setup, 2) Instrument Library, 3) Articulation Library, and 4) Instrument Builder. The MIDI Setup panel allows users to select input/output ports for their MIDI system. In the Instrument Library panel, users can create named Instruments, each with a specific MIDI port and channel assignment. The Articulation Library is used to define named articulation settings—“legato”, “trill minor”, “snap pizz”, etc.—and to configure any program changes, keyswitches, and/or controller changes needed to select these articulations on their MIDI devices. Finally, the Instrument Builder allows users to freely assign Articulations to Instruments. A number of standard, *note-attached* articulations like “staccato”, “down-bow”, “tenuto”, and so on, are assigned to default names in the Articulation Library, and are automatically selected when the appropriate score marking is used. Some examples can be seen in Figure 6. Named articulations (i.e., those not selected directly through notation) can be added to the staff by typing *a*, clicking at the desired point, and entering the Articulation name.



Figure 6. Note-attached staccato, accent, down-bow, and tremolo articulations.

### 3.4. Sharing Staff Data with Links

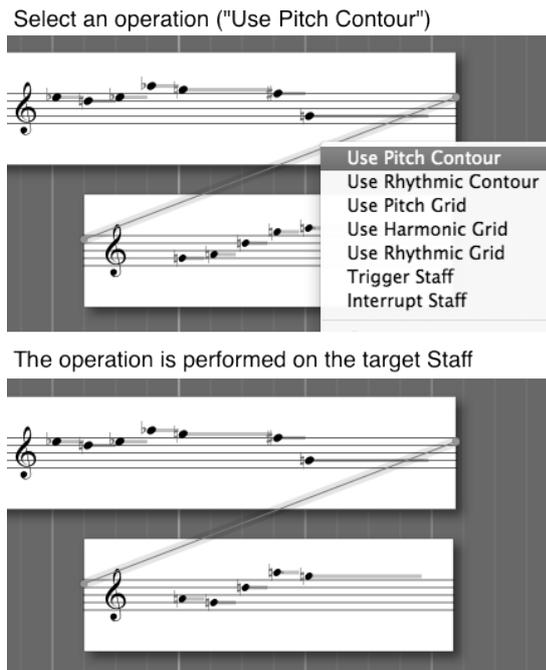
A further application of this notion of ‘objecthood’ comes in the form of *Links*. A *Link* is a graphical connection between two staves that allows one staff to become a source of information for another—a form of *inheritance*. Typing *l* and clicking on a staff will start the linking process by setting the clicked staff as the “source” of the *Link*.

Dragging over another staff and releasing the *Link* will set the staff under the release as the “target” staff. The source staff acts as a data source, and the target staff acts as a receiver of some aspect of the source staff’s data. An example of applying the pitch contour from a source staff to the target staff is shown in Figure 7. *Link* functions currently include the following operations:

- *Pitch contour*: Applies the source staff’s pitch contour to the contents of the target staff. If the target staff has a greater number of events than the source, the source contour is repeated.
- *Rhythmic contour*: Reorders the rhythmic values of events on the target staff to match the rhythmic contour of events on the source staff.
- *Pitch grid*: Provides ‘crisp’ locking of target pitches to source pitches.
- *Harmonic grid*: Provides ‘fuzzy’ locking of target pitches to source pitches. The locking algorithm uses a histogram of pitches used in the score up to the time of the target staff, weighted toward the pitches in the source staff.
- *Rhythmic grid*: Imposes the rhythmic pattern of the source staff onto the contents of the target staff.
- *Trigger Staff*: Allows non-linear playback possibilities by causing the source staff to “trigger” the target staff. When playback of the source staff ends, the target staff begins, regardless of its horizontal position on the score.
- *Interrupt Staff*: If the target staff is playing back at the time when the source staff begins, the target staff is muted; i.e., the source staff “interrupts” the target staff.

## 4. A COMPOSITION STUDY USING MANUSCORE

In the spring of 2011, an applied composition study using ManuScore was conducted by composer James B. Maxwell, working under the supervision of composer/Professor Owen Underhill. The objective of the study was to test the functionality of the software in a one-to-one composition study context. During the study, the composer was to create two short works; one using his regular software package (with which he had been working for many years), and the other using ManuScore. Both pieces were to be approximately 5:00 minutes in duration, and both were to be scored for string quartet. As a further limitation on the process, both works would draw source material from Fredrick II’s “Royal Theme” (best known as the subject of Bach’s *Musical Offering*). The two works would be premiered together, in performance, in the fall of 2011, and a listener study conducted at the concert, as described in Section 5.



**Figure 7.** Using a *Link* to apply the pitch contour from one staff to another.

We do not suggest that the above limitations provide a strict enough framework for quantitative evaluation. However, we do feel that they impose enough commonality on the two compositional processes to isolate, at least to some degree, the software itself as a potential source of difference between the resulting works. Each working process was recorded using video screen capture, in order to provide detailed documentation. An excerpt of playback from the compositional process of the work composed in ManuScore can be viewed online (audio playback in the clip is directly from ManuScore, using the “Vienna Instruments” software):

[http://rubato-music.com/home/Media/MnS\\_experiri.mov](http://rubato-music.com/home/Media/MnS_experiri.mov).

The CbCM in ManuScore was trained on three works from the composer’s catalogue: *vovere*, for flute and ensemble, *limina*, for flute, piano, and percussion, and *pensare*, for wind quintet. Most of ManuScore’s functionality was utilized during the composition process, though the composer reported particular use of the Gesture Line tool and the “pitch grid” Link function. The composer also noted that although he considered the final work to be human-composed, the software did have a strong influence on both the form and content of the piece. In particular, he found that continuations offered by the CbCM tended to provoke different possibilities for the melodic development of the work, even in cases where the continuations were not included in their original, unedited form. Continuations provided by the CbCM were often edited in both pitch content and rhythm, with the former necessitated by key/scale violations in the generated fragments.

Although ManuScore is notation-based, it is not a mu-

sic notation package, since it does not currently export standard, mensural notation. For this reason, a separate process was required to transcribe the completed work into standard music notation for performance. This transcription process revealed some intriguing effects of the ManuScore design, which we discuss in Section 7.

## 5. A LISTENER STUDY

The two works composed during the composition study outlined in Section 4 were premiered in a concert held at Simon Fraser University’s School for the Contemporary Arts, Woodward’s campus, in December 2011. The concert was presented by the Musical Metacreations project at SFU, and also included several machine-composed works by composer/Professor Dr. Arne Eigenfeldt, and one other human-composed work, “*One of the above #1*”, also by Dr. Eigenfeldt.

Participants in the study were 46 audience members from Vancouver’s new music community. The concert featured a total of ten works, written for percussion, string quartet, Disklavier, and other hybrid combinations of these instruments. Each audience member received a concert programme, which explicitly indicated that “machine-composed and machine-assisted musical compositions” would be performed. Each audience member also received an evaluation card on which they were encouraged to provide feedback. On the front side of the evaluation card, audience members were asked to indicate, on a 5-point Likert-scale from 1 to 5, their level of familiarity with contemporary music. This question was followed by ten similar 5-point Likert-scales for rating their level of engagement while listening to each of the compositions. Additionally, audience members were asked to indicate which three pieces they felt were most directly human-composed. All questions on the front side of the card were to be filled out during the performance. The back side of the card contained an additional ten 5-point Likert-scales, asking audience members to indicate the memorability of each piece. This was to be filled out at the end of the concert. However, due to low response rate, this information was excluded from the analysis. Audience members were also given space to write in their own comments. A paper outlining the general findings of the study is forthcoming.

For the purposes of this paper we will focus on the two works composed during the composition study described in Section 4. Since the primary design goal of ManuScore is introduce CAC into the compositional process without disrupting the development of a composer’s musical language, we hypothesize that audience members will not judge the computer-assisted work *experiri* to be implicitly more “human” than the strictly human-composed work, *fundatio*.

## 6. STUDY RESULTS

In order to avoid the alpha inflation that arises from multiple comparisons, statistical tests were made using *post-*

*hoc* Bonferroni-corrected alpha levels of .005 (0.5/10). For part of the analysis, the 46 audience members were divided into *novice* and *expert* groups, based on the score they indicated for the “familiarity with contemporary music” question. The *novice* group consisted of audience members who gave a score of 1-3 out of 5 on the familiarity scale ( $N = 25$ ). The *expert* group consisted of the remaining audience members who gave a 4 or 5 ( $N = 19$ ). Two audience members failed to provide a familiarity score, so their data was excluded from group comparisons.

Table 1 gives the engagement rating for all ten works on the programme. The two works composed during the composition study, *fundatio* and *experiri*, are identified in bold type. The score for “*Other, Previously*”, also written for string quartet, has been italicized to draw attention to the fact that a highly significant difference between the averaged engagement ratings for all string quartet pieces ( $M = 4.36, SD = 0.73$ ) and the “*One of the Above*” series of solo percussion pieces ( $M = 3.36, SD = 1.06$ ) was found,  $t(133) = 8.71; p < .0001$ . Similarly, a comparison between the string quartet pieces and the “hybrid” string/percussion pieces *Dead Slow / Look Left* and *Gradual* ( $M = 3.69, SD = 1.09$ ) was also highly significant,  $t(89) = 4.79; p < .0001$ , suggesting that audience members were more engaged by pieces containing strings than by those containing percussion. A comparison between the percussion and hybrid pieces failed to be significant,  $t(89) = 1.41; p = .16 ns$ .

It is worth noting that comparisons between the expert listener engagement ratings for the two works from the composition study, *fundatio* ( $M = 4.29, SD = 0.81$ ) and *experiri* ( $M = 4.47, SD = .61$ ) were non-significant,  $t(18) = 1.00; p = .33 ns$ . Novice ratings for *fundatio* ( $M = 4.24, SD = 0.83$ ) and *experiri* ( $M = 4.36, SD = 0.86$ ) were similarly non-significant,  $t(24) = .72; p = .48 ns$ .

In Table 2 we give the results for the “directly human-composed” ratings, where it is clear that both *fundatio* and *experiri* were estimated to be human-composed works. Again, there is an effect of instrumentation to be considered, as the other string quartet work was also highly rated (score in italics). However, there was once again no significant difference between the work composed in ManuScore and the work composed through the composer’s normal process.

## 7. DISCUSSION

One of our primary goals in designing ManuScore was to create an application for composers that would allow them to experiment with interactive, generative, ‘object-oriented’ composition, in a manner that would not dramatically disrupt their existing musical language. The fact that no significant difference was found in the level of listener “engagement” between *experiri*, composed in ManuScore, and *fundatio* would seem to suggest that we achieved this basic goal. Further, since listeners were not able to identify *experiri* as the computer-assisted work, it appears that the system did not dramatically alter the com-

Work	Name & Inst.	Listener Experience		
		Expert	Novice	Comb.
1(c)	<i>In Equilibrio</i> (Disklavier)	3.17 (0.99)	2.71 (1.23)	2.90 (1.14)
2(h)	<i>One of the Above #1</i> (Percussion)	4.00 (1.00)	3.36 (1.19)	3.67 (1.13)
3(c)	<i>Dead Slow/Look Left</i> (Str Qrt & Perc)	4.16 (0.90)	3.08 (1.15)	3.51 (1.16)
4(c)	<i>One of the Above #2</i> (Percussion)	3.68 (0.67)	3.16 (1.07)	3.42 (0.93)
<b>5(h)</b>	<b><i>fundatio</i></b> (String Quartet)	<b>4.29</b> (0.80)	<b>4.24</b> (0.83)	<b>4.24</b> (0.81)
<b>6(c-a)</b>	<b><i>experiri</i></b> (String Quartet)	<b>4.47</b> (0.61)	<b>4.36</b> (0.86)	<b>4.40</b> (0.76)
7 (c)	<i>One of the Above #3</i> (Percussion)	3.39 (0.76)	3.12 (1.20)	3.22 (1.04)
8 (c)	<i>Other, Previously</i> (String Quartet)	4.31 (0.75)	4.50 (0.59)	4.40 (0.66)
9 (c)	<i>One of the Above #4</i> (Percussion)	3.63 (1.16)	2.71 (1.00)	3.10 (1.16)
10 (c)	<i>Gradual</i> (VI, Perc & Dsk)	4.05 (0.85)	3.88 (0.95)	3.93 (0.89)

**Table 1.** Audience evaluation of “engagement”: (c) computer-composed, (h) human-composed, (c-a) computer-assisted (standard deviations in brackets).

poser’s musical language. It is also perhaps worth note that, of the two works, the ManuScore-composed work was rated slightly higher in ‘engagement’, though it is impossible to attribute this preference to the influence of ManuScore.

The process of composing in ManuScore introduced some important changes into the compositional process, which would be worth discussing further. Specifically, the manner in which time is represented, combined with the necessity to transcribe ManuScore documents into standard music notation should be considered more closely.

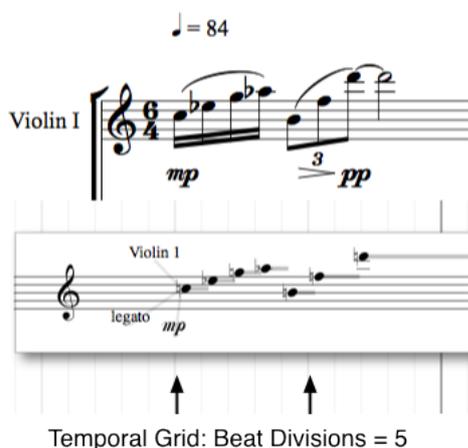
During the transcription process it was noted that the original rhythmic representation of *experiri* did not always follow an easily interpretable metrical structure. More specifically, it was found that the rhythmic representation in ManuScore did not always follow the *implied* metrical structure, as perceived through listening. An example occurs at the opening of the work, and is shown in Figure 8. Looking carefully at the example, we see that the original ManuScore phrase is written using a “beat division” of 5, suggesting a quintuplet pattern. However, it was decided during the transcription process that the aural impression of the phrase was more appropriately represented using a “4+3” grouping of sixteenth-notes and triplets, rather than the original “5+2” grouping. This change effectively increased the tempo, and shifted the entire metrical structure accordingly.

A similar effect was noticed at measure 12 of the transcription, shown in Figure 9. Here we see a passage which was created as a quintuplet pattern in ManuScore (bottom), but transcribed as a grouping of six eighth-notes, under a  $\frac{3}{4}$  metre, in standard notation.

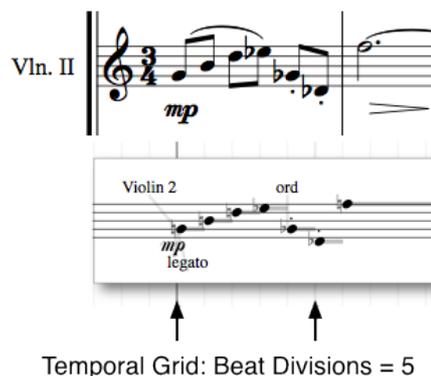
Work	Name	N
1 (c)	<i>In Equilibrio</i>	1
2 (h)	<i>One of the Above #1</i>	12
3 (c)	<i>Dead Slow/Look Left</i>	8
4 (c)	<i>One of the Above #2</i>	2
<b>5 (h)</b>	<b><i>fundatio</i></b>	<b>30</b>
<b>6 (c-a)</b>	<b><i>experiri</i></b>	<b>27</b>
7 (c)	<i>One of the Above #3</i>	2
8 (c)	<i>Other, Previously</i>	24
9 (c)	<i>One of the Above #4</i>	2
10 (c)	<i>Gradual</i>	14
Total		122

**Table 2.** Evaluation of “directly human-composed”: (c) computer-composed, (h) human-composed, (c-a) computer-assisted (standard deviations in brackets).

It was felt that such discrepancies were primarily arising as an effect of the purely graphical nature of ManuScore’s temporal grid. Since the grid does not impose a specific metrical structure on the music, the cyclical process of writing and listening tends to emphasize aural rather than theoretical principles in the developing composition. With a temporal grid of 5 beat divisions in place, pitches were easily entered into ManuScore in quintuplet patterns. However, through the iterative process of listening, entering material, editing, and listening, the musical form naturally began to unfold according to perceptual/cognitive principles, driven by the musical materials themselves. The phrasing of ideas in the musical foreground gave rise to certain types of groupings, and these naturally gave rise to accompaniments that supported those groupings. And because the temporal grid was easy to adjust to virtually any beat division value, it was simply not a priority to alter the metrical structure of the work-in-progress. In a sense, quintuplets became “the new sixteenths” for the work.



**Figure 8.** The opening phrase in ManuScore (bottom) and its transcription into standard music notation.



**Figure 9.** The music at measure 12 in ManuScore (bottom), transcribed as a metric modulation in standard music notation.

## 8. FUTURE WORK

A new version of ManuScore is currently under development. This version is backed by a modular cognitive architecture for music, called MusiCOG, which replaces the CbCM as the generative back-end for the system. An overview of this model, which is a development and extension of the CbCM, can be found in Maxwell et al. [12]. All of the features described in this paper have been included in the new version.

In response to the composer’s experience of frequently altering the pitch content of CbCM continuations to match local key/scale structures, we plan to add functionality for ‘quantizing’ the pitch content of generated material before it is rendered. Since the CbCM often generates continuations based on interval patterns, rather than pitch patterns, deviations from the local key/scale were somewhat expected. However, a pitch quantization method would help reduce the cognitive load on users, and could be implemented using the existing algorithms from ManuScore’s “harmonic grid” *Link* function.

A useful future development, which came to our minds during the present study, might involve the inclusion of methods for beat induction and metrical inference. Such methods would be useful for MusiCOG’s underlying music perception and cognition functions, and could also be used to periodically re-interpret the temporal grid of the score during creation. Such metrical interpretation could help the user avoid difficult transcription decisions after completing a score, and would also support the composer’s understanding of the structure of the work in progress. An extension of this functionality could allow ManuScore to transcribe standard music notation versions of documents, for export and printing, thus streamlining the process of moving from ManuScore to concert performance.

## 9. CONCLUSIONS

It is difficult to evaluate the role that CAC tools play in the compositional process; indeed, the influence of even the most ‘inert’ music notation software on compositional thinking is difficult to deny [2, 3, 4, 22]. ManuScore expands the field of CAC tools by augmenting common notation-based approaches with a more open conceptual design, and with the inclusion of corpus-based, generative capabilities. Although further validation of ManuScore is required, the user and listener studies outlined in this paper suggest that our goal of providing an interactive CAC tool, which enhances the compositional process, without disrupting the composer’s musical language, has been at least provisionally achieved.

## 10. ACKNOWLEDGEMENTS

This work was made possible in part by the Social Sciences and Humanities Research Council, the Canada Council for the Arts, and the Natural Sciences and Engineering Research Council of Canada.

## 11. REFERENCES

- [1] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, “Computer-assisted composition at IRCAM: from Patchwork to Openmusic,” *Computer Music Journal*, vol. 23, no. 3, pp. 59–72, 1999.
- [2] A. Brown, “Modes of compositional engagement,” *Mikropolyphony*, vol. 6, 2001b.
- [3] D. Collins, “A synthesis process model of creative thinking in music composition,” *Psychology of Music*, vol. 33, no. 2, p. 193, 2005.
- [4] —, “Real-time tracking of the creative music composition process,” *Digital Creativity*, vol. 18, no. 4, pp. 239–256, 2007.
- [5] D. Cope, “The Composer’s Underscoring Environment: CUE,” *Computer Music Journal*, vol. 21, no. 3, pp. 20–37, 1997.
- [6] O. Laske, “Composition Theory: An enrichment of music theory,” *Journal of New Music Research*, vol. 18, no. 1, pp. 45–59, 1989.
- [7] —, “The computer as the artist’s alter ego,” *Leonardo*, pp. 53–66, 1990.
- [8] —, “Composition Theory in Koenig’s Project One and Project Two,” *Computer Music Journal*, vol. 5, no. 4, pp. 54–65, 1981.
- [9] —, “Toward an Epistemology of Composition,” *Interface*, vol. 20, pp. 235–269, 1991.
- [10] M. Laurson, M. Kuuskankare, and V. Norilo, “An overview of PWGL, a visual programming environment for music,” *Computer Music Journal*, vol. 33, no. 1, pp. 19–31, 2009.
- [11] J. Maxwell, P. Pasquier, and A. Eigenfeldt, “The closure-based cueing model: Cognitively-inspired learning and generation of musical sequences,” in *Proceedings of the 2011 Sound and Music Computing Conference*, 2011.
- [12] J. Maxwell, P. Pasquier, A. Eigenfeldt, and N. Thomas, “MusiCOG: A cognitive architecture for music learning and generation,” in *Proceedings of the 2012 Sound and Music Computing Conference*, 2012.
- [13] J. McCormack, P. McIlwain, A. Lane, and A. Dorin, “Generative composition with Nodal,” in *Workshop on Music and Artificial Life (part of ECAL 2007)*, Lisbon, Portugal. Citeseer, 2007.
- [14] T. Potter, “All my children: A portrait of Sir Andrzej Panufnik based on conversations with Tully Potter,” *The Musical Times*, vol. 132, no. 1778, pp. 186–191, 1991.
- [15] C. Scaletti and R. Johnson, “An interactive environment for object-oriented music composition and sound synthesis,” *ACM SIGPLAN Notices*, vol. 23, no. 11, pp. 222–233, 1988.
- [16] I. Stravinsky, T. Palmer, N. Elliott, and M. Bragg, “Once, at a border ...: Aspects of Stravinsky,” Kultur International Films, 1980.
- [17] H. Taube, “Common Music: A music composition language in Common Lisp and CLOS,” *Computer Music Journal*, vol. 15, no. 2, pp. 21–32, 1991.
- [18] L. Tenney, J. Polansky, “Temporal gestalt perception in music,” *Journal of Music Theory*, vol. 24, no. 2, 1980.
- [19] B. Truax, “The PODX system: interactive compositional software for the DMX-1000,” *Computer Music Journal*, pp. 29–38, 1985.
- [20] J. Ventrella, “Evolving structure in Liquid Music,” *The Art of Artificial Evolution*, pp. 269–288, 2008.
- [21] G. Wang, P. Cook *et al.*, “ChucK: A concurrent, on-the-fly audio programming language,” in *Proceedings of International Computer Music Conference*, 2003, pp. 219–226.
- [22] J. Wiggins, “Compositional process in music,” *International handbook of research in arts education*, pp. 453–476, 2007.
- [23] D. Zicarelli, “M and Jam Factory,” *Computer Music Journal*, vol. 11, no. 4, pp. 13–29, 1987.