Mus*ink*: Composing Music through Augmented Drawing

Theophanis Tsandilas, Catherine Letondal & Wendy E. Mackay
In|Situ, INRIA Saclay, Ile-de-France
LRI, Building 490, Univ. Paris-Sud, Orsay Cedex, France
fanis@lri.fr, letondal@lri.fr, mackay@lri.fr

ABSTRACT
We focus on the creative use of paper in the music composition process, particularly the interaction between paper and end-user programming. When expressing musical ideas, composers *draw* in a precise way, not just sketch. Working in close collaboration with composers, we designed Mus*ink* to provide them with a smooth transition between paper drawings and OpenMusic, a flexible music composition tool. Mus*ink*'s built-in recognizers handle common needs, such as scoping and annotation. Users can also define new gestures and associate them with their own or pre-defined software functions. Mus*ink* supports *semi-structured, delayed interpretation* and serves as a customizable *gesture browser,* giving composers significant freedom to create their own, individualized composition languages and to experiment with music, on-paper and on-line.

Author Keywords
Creativity, interactive paper, participatory design, musical interfaces, end-user programming, gesture interfaces.

ACM Classification Keywords
H.1.2 [User/Machine Systems]: Human Factors, H.5.2 [User Interfaces]: Evaluation/methodology, Theory & methods, Prototyping, User-centered design.

INTRODUCTION

We are interested in developing tools that support the creative design process, in particular, the composition of original music. Contemporary music composers are an interesting user group because they combine a deep artistic sense with, often, highly mathematical and technical skills. They work with a standard notation, evolved over centuries, and then invent new musical expressions to explore and represent new musical ideas. They often express these ideas on paper, working out different aspects on different levels, over time. Yet they also work extensively with the computer, developing functions for new sounds, many of which cannot be captured with traditional music notation, in a sophisticated form of end-user programming [14]. A key challenge then is to create tools that support this creative process and to provide composers with a rich, nuanced exchange between paper and the computer. We need to enhance, rather than replace, the composer's ability to generate and explore musical ideas.

Although a number of researchers have developed paper-based interfaces using Anoto[1] technology, the usual emphasis is on improving productivity, enabling users to accomplish particular tasks by combining paper and electronic documents. Our focus is slightly different: we are interested in understanding and supporting the creative design process itself. We have been working closely with contemporary music composers, who use state-of-the-art music composition hardware and software, combined with extensive technical support. Even so, they continue to make use of paper as a tool, from earliest sketches to the final printed score. We decided to collaborate with them to better understand both the role of interacting with paper in a creative context and how to integrate paper and end-user programming tools to enhance creativity. We build upon their existing, highly personal methods of expressing ideas to support an open-ended, customizable cycle of interaction between paper and the computer.

This article begins with a description of our initial study of composers, including the insights that led to our initial design of Mus*ink*. Based on Anoto technology, Mus*ink* allows users to express, annotate and interact with personally generated musical ideas, moving back and forth between paper and OpenMusic [2], a state-of-the-art music composition system. We then present our findings from a series of mini-workshops with individual composers who experimented with Mus*ink* and our rapid, iterative redesign in response to their suggestions. We then offer insights into how creative individuals find novel ways of interacting with paper and the requirements for tools such as Mus*ink*, to support their diverse needs. We conclude with a discussion and directions for future research.

RELATED WORK

Paper is a powerful medium, easily underestimated, that provides users with a range of opportunities, from initial sketches to definitive publication. We have long been inter-

CHI 2009, April 4–9, 2009, Boston, Massachusetts, USA.

[1] The *Anoto* pen's camera captures gestures on paper printed with a computer-readable, human-invisible dot pattern *www.anoto.com*

ested in the dilemma faced by people with excellent reasons for using both paper and computers. Early work on the Digital Desk [30] and Video Mosaic [17] explored the benefits of combining paper and computers. Augmented paper has been studied in several domains, e.g., annotated engineering drawings [15], flight strips [18], text editing [9, 10] and hybrid laboratory notebooks [19]. *Anoto* technology made interactive paper practical and launched applications for scientists [27, 35] while encouraging exploration of multi-media indexing [4, 25], copy-paste between paper and documents [13] and paper-mobile interfaces [9].

Several research groups have focused on the architecture of paper interaction, notably PapierCraft [13], PaperProof [29], and ModelCraft [24]. Each approach proposes a specific pen-based set of gestures that are linked to pre-defined computer functions. Users can use these gestures to perform a command, e.g., copying a picture from one page to another, replacing a word, or editing a physical model. Although extensible, the gesture sets are defined by the application designer, not the user.

More generally, HCI researchers have begun exploring creativity, not just productivity, with specialized conferences such as *Creativity & Cognition*. Csíkszentmihályi's [7] work on 'flow' have influenced key HCI researchers [22], who advocate tools that support idea generation and sharing [1, 3]. This corresponds with an increasing interest in design, both as a method and a focus of study.

Similarly, research in interfaces to support music composition, has grown significantly since Buxton's early work [5] on interaction techniques for drawing musical notation. The *NIME (New Interfaces for Musical Expression)* conference combines music and HCI research and is a fertile area for end-user programming [14], because musicians and composers often create their own instruments and electronically generated sounds [28].

We are interested in how to create an interactive paper interface for contemporary music composers that supports individual creativity and bridges the gap between paper-based and on-line expression of musical ideas. We began by studying composers, to understand the current role of paper.

STUDY 1: INTERVIEWING COMPOSERS

Composers pose an intriguing user interface challenge: How can they use the computer as tool, but still create art? They do not want increases in efficiency, per se, but rather support for reflection and exploration of ideas. Composers would reject a system that automatically composes for them; instead, they seek tools that provide maximum individual freedom of expression but also maximum control over the computer, throughout the creative process.

Method

We interviewed 12 composers and musical assistants at IRCAM, a world-famous center for contemporary music in their offices or the labs where they composed music. Most composers are proficient or skilled computer users; they can also rely on music assistants, usually computer scientists with musical training. We also met with an IRCAM research team to discuss their longitudinal study of a single composer and how he used paper while composing [8].

Results

We found that, despite access to the latest computer-music tools, these composers continue to use paper documents. However, they were dissatisfied with the lack of connection between their off-line scores and the on-line software that generates the resulting music. Details of the study appear in [12]; here we focus on the results that affected the design of Mus*ink*, specifically chronology and choice.

Chronology of paper and computer use

Why don't composers of electronic music simply use a computer? Clearly, it is not due to fear of computers, nor is it particularly related to user interface problems, since most of their tools are designed by and for musicians. Rather, they choose the appropriate medium for the purpose at hand [21]. Composition progresses from an initial creative stage to the final piece, with much iterative development in between. Design artifacts evolve over time, from quick early sketches, through systematic explorations of alternatives, to the definitive printed work.

The desired characteristics of the design medium evolve as well. In the beginning, composers use paper because it is flexible, easy to transport and less cumbersome than a stylus on a graphics tablet. Most importantly, paper permits free associations and provides a direct link between a human gesture and a musical idea. We were struck by the innovative ways these composers found to represent their ideas. Figure 1a illustrates the relationships among different elements of a symphony; Figure 1b shows where the composer has added numbers under each instrument part to indicate loudspeaker assignments to create a specialization effect.

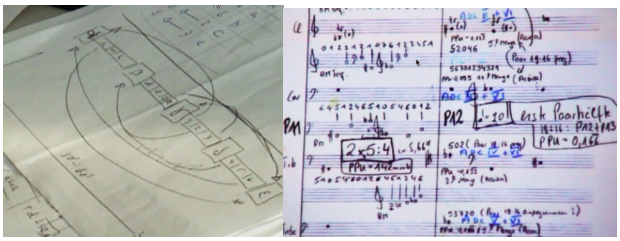


Figure 1. (a) Organization of the components of a symphony. (b) Annotations on a hand-written score

In the middle of the process, paper and computers each offer flexible, but different, modification capabilities and power of expression. However, in the final stages, paper is no longer valued for its flexibility, but rather for its permanence as a reference point and archival artifact. One composer reported that he even paid someone to rewrite his scores by hand, from the 'final' version on the computer, so as to create the true 'original manuscript'.

Choosing between paper and computer

In many cases, composers move easily back and forth between paper and computer, with no conflicts. Figure 2a shows a composer using the printed score to reference on-line musical materials: a poem used as data for building sounds, a rhythm series, drawings and code. He can browse the printed score and uses folders containing paper of various sizes to keep track of the structure of his composition.

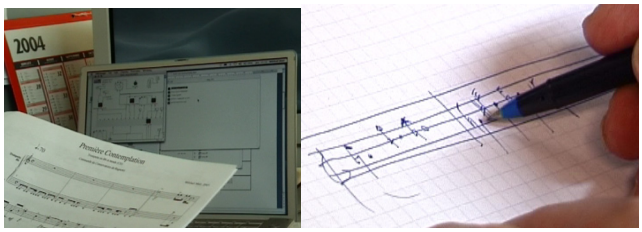


Figure 2. (a) Navigating between paper score & on-line files. (b) Extended musical notations for 1/4 and 1/8 tones.

However, some composers experience a conflict: neither medium suffices by itself, nor they do not work well together, making it difficult to choose. In electronic music, the computer is the instrument, but also a tool for creating new instruments and exploring a musical space. However, the preferred medium for imagination and writing remains paper, *because it is slow and static*. Most composers only use electronic music editors, such as Finale, when they want to implement an idea that has been already expressed on paper.

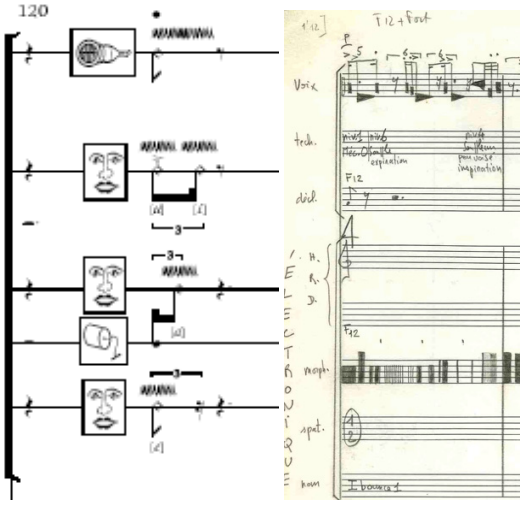


Figure 3. (a) Score with added symbols to show actions to be performed by the choir. (b) 'Electronic' is written vertically, with links to various computer programs that are played in addition to the more traditional score to the right.

In electronic music, the choice of the final format is more complex than in classical music: with an electronic format, the composer has to deal with non-conventional notations, such as in Figure 3a, where the composer includes graphical notations that indicate particular sounds for a choir. In contrast, Figure 3b shows a hand-written score with links to various electronic documents and statements that trigger particular computer programs while the piece is being played. This hand-written paper score serves as the key reference point, with links to the computer, but the final work is fundamentally located both on-line and on paper.

Implications for design

These interviews provided us with useful insights about the role that paper plays in creativity and suggested new ways that interactive paper can support composer's inventive design process. Each composer has a unique creative process, with personal strategies for expressing sounds, rhythms, variations and structures. Each composer also has a set of custom-made computer tools, e.g., created with AudioSculpt [28] and manipulated with OpenMusic, a visual programming environment to support composition [2]. At the most basic level, it was clear that a future system must provide a flexible way of linking composers' drawings to their music composition tools. However, we also wanted to create a testbed for exploring creative design with interactive paper. One of our system's guiding objectives was to give users maximum control over the assignment of meaning to their gestures, which required a trade-off between openness and recognition. Offering a blank slate with complete openness and perfect recognition is impossible. The challenge was to create enough scaffolding to permit sufficient recognition to be useful, while allowing users to invent a wide variety of different representations of their musical ideas and add meaning over time, as required.

MUS*INK*: CYCLING BETWEEN PAPER MUSIC SCORES AND COMPUTER COMPOSITION TOOLS

We designed Mus*ink* as an extensible gesture-based language that uses a common musical structure, the 5-line staff of a musical score and a small number of basic, recognizable gestures. Users can define a personal interaction vocabulary and associate it with computer tools such as OpenMusic. Figure 4 illustrates a scenario in which a composer expresses a musical idea on paper and manipulates it using Mus*ink*'s *Gesture Browser* and OpenMusic.

Scenario: *Leonard has invented a new type of crescendo that vibrates according to a particular pattern he defined in OpenMusic. He prints an earlier version of his composition onto Anoto paper and uses an Anoto pen to draw angled lines over several series of notes. He uploads the pen data and opens the Mus*ink Gesture Browser*, which displays the hand-annotated version of his score. He selects an instance of a crescendo and right-clicks to open a gesture-definition dialog box. He then specifies the details, including a link to the relevant OpenMusic function. Mus*ink *automatically recognizes most instances of the crescendo gesture. For the rest, Leonard points to the unrecognized gesture and uses a marking menu to assign the crescendo function. Later, Leonard explores several implementations of the new crescendo in OpenMusic's workspace and tests their outcome as different variations of his piece.*

Design Considerations

The main goal of Mus*ink* was to provide a tool that would stimulate our design explorations with composers. We be-

gan with a basic design goal, i.e. to optimize the trade-off between openness and recognizability but also remained open to new ideas that emerged as part of our participatory design process. Given the prevalence of OpenMusic as a tool at IRCAM, we decided to integrate several aspects of OpenMusic's design philosophy into our approach. Specifically, we treat gestures as functions that can take properties of musical objects as arguments, e.g., their rhythm or pitch, and generate new objects. We started in the middle of the design process, when musical scores are already present, and explored how to augment them, using Mus*ink* to create new scores. We knew that most composers do not simply write a finished section of a score, but rather work progressively, adding layers of nuance over time. Thus, Mus*ink* is designed to allow composers to reprint their scores in multiple cycles, annotated with new gestures and reprocessed together with other evolving musical objects.

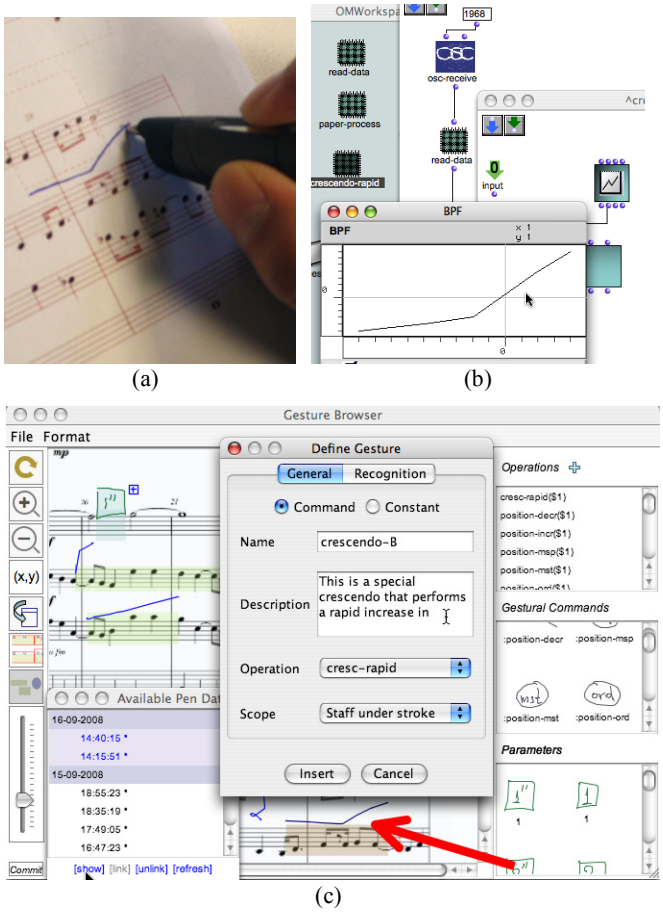


Figure 4. A Mus*ink* scenario
(a) Drawing on paper: expressing a new type of crescendo
(b) OpenMusic: defining the crescendo's vibration pattern
(c) Mus*ink Gesture Browser:* defining the crescendo class

The challenge is how to provide the above functionality while respecting the natural role of gestures on traditional paper. Although paper-based gestures sometimes serve as commands that perform specific software operations, they are also declarative, with a representation designed to be recognizable by people. We thus studied existing forms of annotations of musical scores and integrated them into Mus*ink* gestures.

Interacting with Mus*ink* on Paper

The basic Mus*ink* syntax supports the three elements identified by Winget [31]: *symbols* drawn over or under individual notes or phrases, *numbers*, usually representing fingering or tempos, and *text*. This is consistent with Chapuis et al.'s classification [6] and enables us to handle both simple (single-trace) gestures and complex gestures in which multiple traces are logically linked together. Mus*ink* gestures thus include: a *scope,* to specify the elements of the score to which a function is applied, a *temporal* range within the score, and (3) a graphical representation. Figure 5 illustrates several examples of complex Mus*ink* gestures.

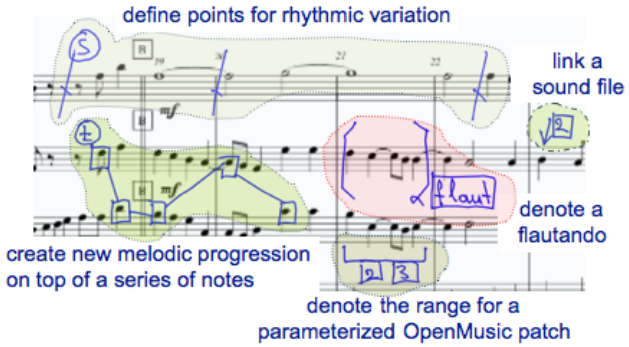


Figure 5. Examples of operations expressed with Mus*ink*

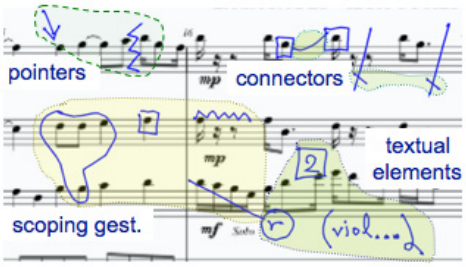


Figure 6. Examples of Mus*ink* basic gestures

Figure 6 shows Mus*ink's* basic, generic gestures:

- *Pointers:* describe specific locations within the score's timeline. *Forms:* vertical curves; arrows.
- *Scoping gestures:* define a range within the score, either as a set of musical symbols or a temporal range. *Forms:* closed curves; horizontal strokes under or over a staff; parenthesized scopes (as in PapierCraft [13]).
- *Text and Parameters:* may be an annotation, an identifier or a parameter. *Forms:* parentheses; any gesture enclosed within a closed curve, e.g. a circle or a rectangle. May be linked to pointers or scoping gestures if the surrounding curve touches or is close to the gesture.
- *Connectors:* are supplementary strokes that group elementary gestures. *Forms:* line segments that visually connect the traces of two gestures; marks indicating a group of traces with a series of small line segments, (useful when traces are distant, e.g., they appear on different pages).

Semi-structured delayed interpretation

Users do not need to have a formal semantic definition of a Mus*ink* gesture as it is being drawn on paper. It may act solely as a structural element in the score or as a symbol that represents an abstract idea. The user can revisit it later to, for example, assign it semantic meaning or link it to another gesture. We refer to this as *semi-structured delayed interpretation*. Mus*ink* uses identifiers to define semantics. A pointer or scoping gesture may use its own graphical representation as an identifier. For example, the zigzag shape of a horizontal line may act as the identifier for a "tremolo" gesture, distinguishing it from other horizontal lines. Alternatively, a text 'tag' may act as an identifier when attached to any pointer or scoping gesture. Any identifier can represent a computer function, e.g., an OpenMusic patch. The function can take any of the following as arguments: score positions, musical symbols, temporal ranges, text and numeric parameters associated with the identifier, either directly or through connectors.

Interacting with Mus*ink* on the Computer

Figure 4c shows how users can assign semantics to gestures via Mus*ink*'s *Gesture Browser*. The interface is implemented in Java 6 using the PaperToolkit framework [34] to load pen data. The main pane of the Gesture Browser displays the PDF of the printed musical score augmented with an interactive layer that shows strokes drawn on paper. A smaller pane lists the pen data available to load. The toolbar to the left includes tools for updating the recognition of gestures, zooming in and out, visualizing the underlying score model, adapting the sensitivity of recognition, and directing recognized function calls to OpenMusic. The three panes to the right provide: a list of user-defined operations or functions that can be linked to gestures; a list of defined gesture identifiers; and a list of textual/numerical elements, intended for use as function parameters. Thumbnails for the classes in the two lists are automatically generated from the associated gestures, as they are first defined.

Interaction with Gestures

The Gesture Browser lets users define new gestures and refine gesture recognition results. Users can right-click on the gesture's trace to display a marking menu [11] (Figure 7a). Users can remove a basic gesture, revise its recognized scope, define a new gesture class (Figure 7b) and associate (or disassociate) the gesture with a previously defined class. Gestures classes can function either as gesture identifiers or as parameters. They can also be linked with user-defined OpenMusic functions as they are defined. Figure 7c shows how the user can modify the scope of a gesture and Figure 7d shows how the user can review the assignment of function arguments on complex gestures.

Function Definition

Users can define new functions on the fly, by assigning a unique function name and specifying its arguments. Supported argument types correspond to the data types that a gesture can represent: sets of musical symbols, pointers in the score, temporal ranges, and textual/numerical values. The actual implementation of functions is beyond the scope of the Gesture Browser. Instead, it sends the function name and arguments of the recognized gestures to OpenMusic from which the user can create a function or patch to process a call.

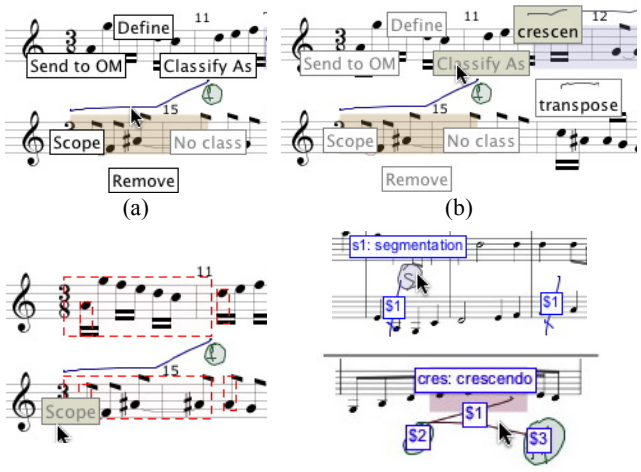


Figure 7. Interacting with Mus*ink* to classify a gesture
(a) right-click on a gesture to activate a marking menu
(b) choose 'classify as' and specify *crescendo*
(c) modify the scope of the *crescendo* gesture
(d) left-click on identifiers to view the assignment of function arguments (numbers indicate recognized arguments)

We use the Open Sound Control (OSC) protocol [33] to establish the connection with OpenMusic. OSC is a platform-independent communication protocol used to share data in real time between musical instruments, multimedia devices and computers. Note that this system architecture allows for connecting the Gesture Browser with other music applications that support OSC, e.g., Max/MSP, but we have only tested it with OpenMusic.

Technical Details about Gesture Recognition

Mus*ink* separates recognition into multiple steps, to simplify both recognition and customization. Recognizing elementary gestures is relatively easy, since it involves only a few fixed gestures. However, this small set can produce multiple alternative representations for a given function. Mus*ink* recognizes gestures in three steps:

- identify elementary strokes: pointers, scoping gestures, connectors, and textual elements
- recognize identifiers and parameters
- match gestures grouped under recognized function identifiers using arguments of their associated functions

The first recognition step takes the score structure into account and uses several heuristics, including the $1 recognizer [32]. The latter performs particularly well when distinguishing among open and closed curves. We use Rubine's algorithm [20] as implemented by the iGesture framework [23] to recognize identifiers and parameters.

This fit to our design goals better than the $1 recognizer [32], because it provides a reliable mechanism for rejecting gestures that do not belong to a defined gesture set.

The third recognition step applies only to gestures that have been recognized as function identifiers. The goal is to match function arguments with compatible data elements represented by pointers, scoping gestures, and textual parameters. The algorithm follows connections within a group of gestures, starting from the identifier gesture and moving to other connected gestures based on simple heuristics.

Preliminary Evaluation of Gesture Recognition

We ran a small user study to test the accuracy of recognition of Mus*ink*'s basic vocabulary and assess the technical viability of our approach. We recruited six participants with basic knowledge of classical musical notation. In each 20-30 minute session, we asked the participant to use an Anoto pen to interact with a pre-printed musical score. They tested various examples of Mus*ink*'s elementary gestures in series of five controlled annotation tasks.

	elements to be recognized	correct	total	accuracy
Basic Strokes/Gestures	1. closed scopes	48	50	96%
	2. parenthesized scopes	60	60	100%
	3. text/parameter elements	168	179	94%
	4. direct line connections	45	49	92%
	5. connections with small lines	40	48	83%
	6. connections with parameters	145	179	81%
	7. arrows	39	61	64%

Table 1. Average recognition accuracies for elementary gestures tested in the study

Table 1 summarizes the results. With the exception of arrows, other basic symbols were recognized at 80% or better accuracy, and most at over 90%. Many of the errors that we observed were due to various unpredictable ways that some participants drew on paper. For example, we detected line connectors that were drawn by repeatedly moving the pen from one point to another. Also, some closed curves were drawn in two steps, creating two strokes rather than one, which was not anticipated by the recognizer. Although we can expect that users would eventually learn how to draw gestures 'properly' to avoid such errors, such situations are not always preventable. The study helped us assess fixable limitations of our recognizer and improve our heuristics accordingly. We also believe that the introduction of digital pens with direct feedback, such as audio, currently available in Livescribe® pens, could minimize the problem. However, as 100% accuracy may not be feasible, our continuing goal is to support powerful online interactions that give users the option of post-hoc specification of gestures.

EXPLORATORY EVALUATION WITH COMPOSERS

After releasing the first version of Mus*ink*, we conducted a series of mini-workshops with individual composers. Our first goal was to get their reactions on Mus*ink* and Mus*ink*'s Gesture Browser. We also wanted to stimulate a design exploration and reflect on the potential of augmented paper in music composition.

Method

Participants: We met with five composers: MM, DC, JH, MS, and PL. Two had participated in the first study. Four were senior composers with long experience in music composition. The fifth was just completing his Ph.D.

Participatory design: Prior to each session, we collected artifacts, including scores, research articles and analyses of their work [8]. We had a second meeting with four out of the five composers. In the interim, we modified Mus*ink* to provide novel functionality that they had suggested. This allowed us to better capture the needs of each individual composer and explore several design alternatives. We asked them to bring drafts of their current compositions on paper and their laptops, including OpenMusic patches. We printed several pages of their compositions printed on Anoto paper. We also scanned the score of a composer and removed his inked gestures, so we could explore how he would use Mus*ink* to re-annotate his work. We gave the composers a NOKIA digital pen, and, in the final four interviews, we also brought a Livescribe® pen to observe how composers would react to the audio feedback, and reflect on new possibilities that this pen would permit.

We encouraged composers to use the digital pen and Anoto paper, and let them interact with paper as they normally did. Our goal was not to enforce them to adapt to our approach, but rather adapt our approach to their current work practices. We demonstrated the Mus*ink* Gesture Browser and discussed the benefits and the constraints it posed.

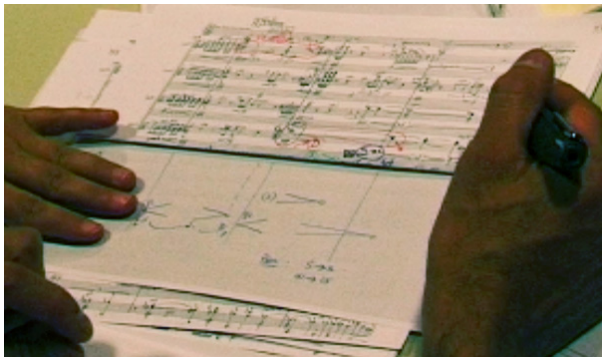


Figure 8. Directions for faders on a music console

Results

Gestures to control electronics during a performance: a layout and semantic issue

MS raised a problem he faces, i.e. to control the levels of faders on a mixing console during a performance. Currently, he draws these by hand. He demonstrated the potential use of Mus*ink* on a page of its own score together with a folded Anoto paper (Figure 8). He drew directions and linked them to the corresponding parts of the score. This led him to reflect on how to balance layout and visualization

issues, including editing, performing, printing and publishing perspectives. His reflections suggest that such design decisions should be left flexible enough to accommodate diverse types of use.

Composing with words

Figure 9 shows how PL would use interactive paper to specify profile data in a patch for generating notes. The composer uses the profile of the shape of letters in a word to parameterize the generation of notes from a chord.

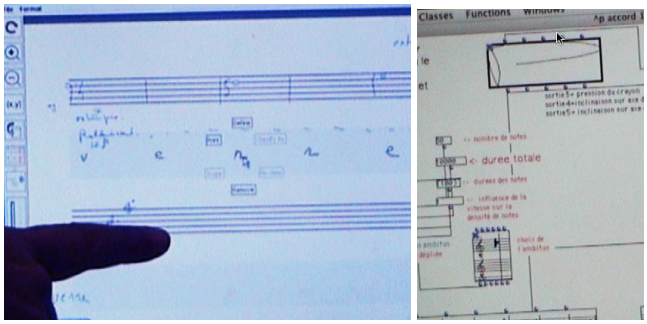


Figure 9. Words as musical parameters.
Left: The word "verre" is parameter for the Gesture Browser.
Right: OpenMusic patch: the shape of the letter P serves to generate notes.

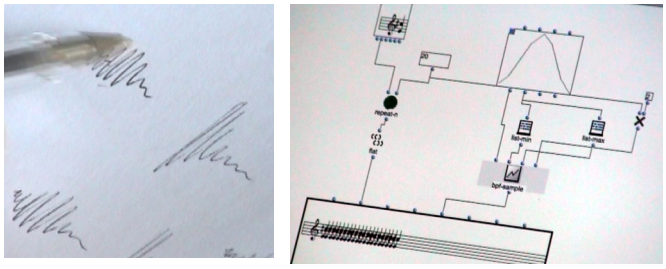


Figure 10. Representation and implementation of tremolos
Left: Gestures representing tremolos drawn on paper.
Right: OpenMusic patch that generates a sound from the envelope of a tremolo gesture.

Composing music though drawing and programming

Figure 10 shows a tremolo patch, controlled by an amplitude curve. MM showed how he would specify this with a gesture he had just drawn on paper.

MS encouraged us to create special Mus*ink* graph paper, so he could draw extremely precise curves. These were not sketches, but rather precise definitions used to generate families of curves in OpenMusic, which could then be used as parameters for a variety of functions (Figure 11a).

DC added his own gesture vocabulary, rather than inventing a new one. He explained that OpenMusic did not meet his needs, but started to explore new ideas when he realized that he could use the Gesture Browser "as if" it were Open Music (Figure 11b). He emphasized his need for "expressive" gestures in his score, as opposed to "representative" ones. He explored using them as drawings and suggested that OpenMusic would now be useful at the analysis stage, for example, to interpolate between drawing variants.

Musical notations as parameters for programming

MS explained that the electronic parts of a piece generally need a semantic representation, such as a drawing of the resulting sounds (Figure 11c). We compared a published version of one of his earlier works [26] to his current notation. Today, he has to recopy these sounds manually, but he wants to be able to move smoothly back and forth between either form: electronic patches in OpenMusic and paper.

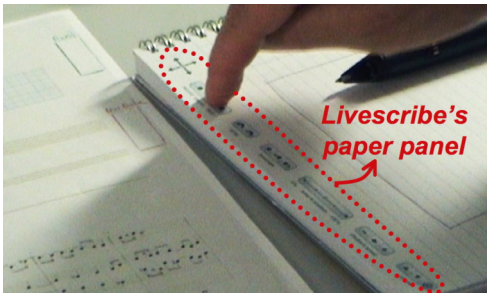
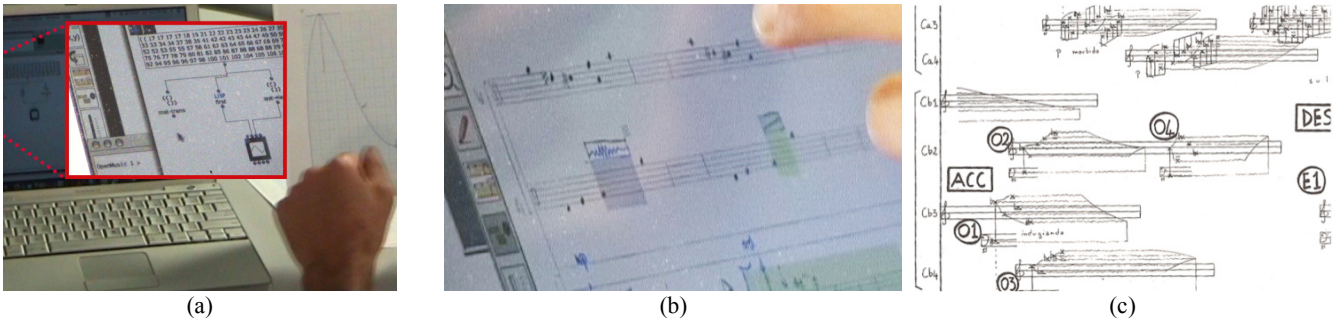


Figure 12. Thinking-aloud over a Livescribe notebook and a score paper interface: exploring how to extend Mus*ink* with a paper palette for gestures.

Brainstorming on interactive paper interfaces

We used a Livescribe® pen to explore with composers alternative ways to interact with Mus*ink*. When we showed the interactive pen and its audio feedback, MS quickly sketched how to define gestures on paper rather than online, creating a customizable paper panel of gestures, similar to those in Livescribe notebooks (Figure 12).



(a) (b) (c)

Figure 11. (a) Linking the data points of a precisely drawn curve with OpenMusic. (b) Directions displayed on the Gesture Browser on how to play the congo (percussion). Such directions could potentially become drawings (graphical data) for OpenMusic.
(c) Graphical representation of electronic parts in a published composition, Traiettoria.

Insights for the Composition Process

Role of Paper Drawings

We were surprised to learn that these composers emphasized drawing, rather than sketching, in the composition process. Their disciplined gestures express concrete musical ideas. We observed three types of drawings throughout the interviews:

- *Directions* to instruct either human performers or electronic equipment. Each has a distinctive graphical representation and scope within the musical score.
- *Symbolic representations* of musical objects and ideas. Several composers have two versions of each score: performers see the notes; composers see their personal representation of musical ideas.
- *Graphs*, rather than notes. Composers can specify musical input along various dimensions, such as nuances in sound, rhythm, or instrument. Once on graph paper, these drawings are translated into coordinates onto the computer. This explains why they are more controlled and precise than initial sketches: the composer is aware of their meaning for future computer operations. By drawing graphs on paper, composers add natural input that mitigates the deterministic nature of computation.

Role of Computers

Composers viewed the computer as an important tool for generating variation. Here, drawings and programs play a complementary role, since tools like Mus*ink* and OpenMusic let them program with their drawings. However, it is important to emphasize that, while the computer may generate alternatives, it is always the composer who is responsible for the final result.

Evolution of Mus*ink*

Extended Data Representations

We found that conventional forms of printed musical scores, as supported by tools like OpenMusic, were unable to support the rich data representations that composers normally use. We extended Mus*ink*'s model to support rich musical notation drawn on paper. This allows users to draw arbitrary symbols along the timeline of a musical piece (Figure 13). Users can interact with them using Mus*ink* gestures as they do with printed symbols like notes and rests. Even if not recognized, such symbols can be linked with other parts of a musical score through their position in the score's timeline.

We also support the use of graphical gestures and parameters, and graphs (Figure 14), treating them as a special type of musical object. In Mus*ink*, graphical data are represented through lists of x-y coordinates. This representation can be transmitted to OpenMusic through Mus*ink*'s user interface. OpenMusic provides advanced tools for editing graphical data and processing them with other musical objects. Accordingly, Mus*ink*'s syntax was enhanced to enable interactions with graph data and link it with other musical representations. Users can use Mus*ink* to point to and select graphical data, specify temporal ranges, and attach parameters and function identifiers.

Mixed Paper Formats

In order to support the new data representations, we experimented with new paper formats, such as pages with empty staffs, pages with simple parallel lines defining the timeline of musical events (Figure 13), and graph paper for drawing graphs (Figure 14). We also prototyped paper layouts that mix multiple data representations, e.g., pages with strips of graph paper lying over staffs with regular notation, printed from the computer. These layouts and their online models were created manually by the researchers although several composers asked to be able to create their own.

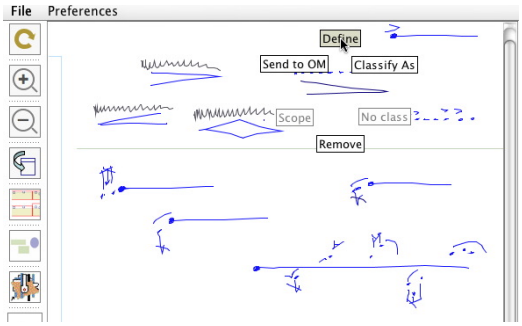


Figure 13. A score created by MM during an interview is shown on the Gesture Browser (notice that 5-line staffs have been replaced by single-line timelines). Sophisticated musical notation can stay unrecognized and coexist, through the score's timeline, with symbols defined as Mus*ink* gestures.

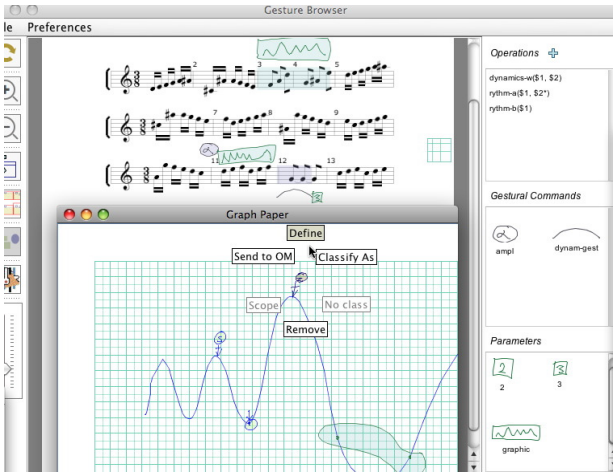


Figure 14. Support of graphical data in Mus*ink*'s Gesture Browser. Scores can contain graphical parameters, and Mus*ink* gestures can be linked with detailed graphs. Mus*ink*'s syntax has been extended to support functions over curves.

Differentiating between Different Roles of Gestures

The coexistence of different representations (notation, graphs, directional Mus*ink* gestures) makes recognition harder. We have explored how to differentiate between them based on context-specific information, e.g., the size and form of curved lines drawn by users or their position, but this solution is not general. Other approaches [10, 13]

have addressed similar problems by introducing additional writing modes. Switching between modes can be handled by using a different pen, e.g., a pen with a different ink color, pressing a pen button, or alternatively, ticking on specialized paper areas. As reported earlier, we explored with composers the potential of defining gestures by using paper palettes and an interactive pen. Although this strategy facilitates gesture recognition, switching modes has drawbacks, and we are still exploring other alternatives.

CONCLUSIONS

We are interested in the role that paper plays in creativity and how interactive paper technologies can support the creative design process. We conducted a series of interviews and mini-workshops with music composers, which revealed an astonishing variety of strategies for creating and representing musical ideas. Each composer has a unique creative process and a set of their own custom-made computer tools, created with software like AudioSculpt and manipulated with tools like Open Music. At a basic level, we created Mus*ink* to link paper-based drawings to these music composition tools.

However, our more ambitious goal was to create a testbed for exploring creative design with interactive paper, giving users complete control over the assignment of meaning to their gestures. We began with a basic structure, the musical staff, and a small set of gestures that Mus*ink* could recognize without training. Mus*ink* provides the user with a flexible set of scoping techniques for specifying groups of musical objects, both with respect to each other and to the underlying score. In addition, the user can draw any gesture, which Mus*ink* will either recognize or ignore. At any point in the future, the user can select a gesture and assign it a meaning on the computer, either a custom-made function or a pre-defined command. Mus*ink* tries to interpret similar gestures in the way, but the user retains control over the results. This notion of 'semi-structured delayed interpretation' gives users a powerful combination of freedom of expression when drawing musical ideas, while retaining the possibility of adding computational power at any time.

Five composers donated examples of their personal compositions, which we printed on Anoto paper, and participated in collaborative design sessions with us, to explore new possibilities for Mus*ink*. During this period, Mus*ink* evolved very quickly, to incorporate specific suggestions from one composer to the next. Although each composer was unique, tools to support one composer were often directly relevant to the others. We came to view Mus*ink* as an interactive paper interface to Open Music and co-invented new Mus*ink* functionality with the composers. For example, we provided different paper structures (5-line staff, variable-width graph paper, space devoted to drawing curves) and began to explore how the print-edit-reprint cycle could significantly affect the composition process.

We admit to a certain bias in our approach; after many years of studying interactive paper in different domains, we fully expected to find interesting uses of paper/computer interaction. However, we were genuinely surprised by the incredible diversity and cleverness these composers evidenced in their drawings, which we came to understand are in fact, musical objects in their own right. Unlike in other creative domains we have studied, including from architecture to fashion design, these composers rarely 'sketch'. Instead, they make very precise drawings in which the physical characteristics of each gesture has meaning with respect to the musical idea being expressed and the computation that will result. This is an unusual user group, with skills as end-user programmers, extreme creativity, and, at the same time, highly disciplined and precise gestures. (Many of these composers are musicians, which was evident as we watched them move and make gestures in the air.)

Future work will assess Mus*ink* in real work environments. Several composers have expressed interest in trying Mus*ink* and we plan to follow at least one composer from the beginning to the end of a new musical composition.

Summary

Our studies of composers have demonstrated the important role that paper continues to play in the creative process, even when composers are adept computer users. Paper remains the optimal choice at the two extremes of the creative process: drawing initial ideas for the first time and creating an archival record of the finished work. Between these two extremes, composers mix paper and computers in a variety of ways, influenced by the particular input/output characteristics of each.

We have explored how the use of Anoto technology expands the options available to composers in this interim stage of creation and supports *semi-structured delayed interpretation*. Mus*ink* allows composers to add innovative notations to traditional music scores, a common theme in contemporary music, a framework in which paper, printouts and files can be linked together and support the process of co-adaptation [16] in which users are encouraged to adapt technology to meet their own unique needs.

ACKNOWLEDGMENTS

We are grateful to Carlos Agon, Nicolas Donin, Aurélien Tabard, as well as the composers and musical assistants who participated in our interviews.

REFERENCES

1. Abrams, S., Bellofatto, R., Fuhrer, R., Oppenheim, D., Wright, J., Boulanger, R., Leonard, N., Mash, D., Rendish, M. & Smith, J. QSketcher: an environment for composing music for film. In Proc. Creativity and Cognition, ACM Press (2002), 157-164.
2. Agon, C., Assayag, G. & Bresson, J. OM Composer's Book. 2006: Editions Delatour France, IRCAM.
3. Amitani, S. & Hori, K. Supporting musical composition by externalizing the composer's mental space. In Proc. Creativity and Cognition, ACM Press (2002), 165-172.

4. Arai, T., Aust, D. & Hudson S. PaperLink: a technique for hyperlinking from real paper to electronic content. In Proc. CHI 1997, ACM Press (1997), 327-334.
5. Buxton, W., Reeves, W., Baecker, R. & Mezei, L. The use of hierarchy and instance in a data structure for computer music. Computer Music Journal, 1978. 2(4): p. 10-20.
6. Chapuis, Y., Fober, D., Letz, S., Orlarey, Y. & Daudin, C. Annotation de partitions musicales dynamiques. In Proc. JIM (2007), 18-27.
7. Csíkszentmihalyi, M. Flow: the psychology of optimal experience. 1990: New York: Harper and Row.
8. Donin, N., Goldszmidt, S. & Theureau, J. De Voi(rex) à Apocalypsis, fragments d'une genèse. Exploration multimédia du travail de composition de Philippe Leroux in L'inouï revue de l'Ircam, no 2 2006.
9. Heiner, J., Hudson, S. & Tanaka, K. Linking and messaging from real paper in the Paper PDA. In Proc. UIST 1999, ACM Press (1999), 179-186.
10. Hinckley, K., Baudisch, P., Ramos, G. & Guimbretière, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In Proc. CHI 2005, ACM Press (2005), 451-460.
11. Kurtenbach, G. & Buxton, W. The limits of expert performance using hierarchic marking menus. In Proc. InterCHI 1993, ACM Press (1993), 482-487.
12. Letondal, C., Mackay, W. & Donin, N. Paperoles et musique. In Proc. IHM 2007, ACM Press (2007), 167-174.
13. Liao, C., Guimbretière, F. & Hinckley, K. PapierCraft: a command system for interactive paper. In Proc. UIST 2005, ACM Press (2005), 241-244.
14. Lieberman, H., Paternò, F. & Wulf, V. End user development (Human-Computer Interaction Series). 2006: Springer-Verlag.
15. Mackay, W. Augmented reality: Linking real and virtual worlds: a new paradigm for interacting with computers. In Proc. AVI 1998, ACM Press (1998), 13-21.
16. Mackay, W., Responding to cognitive overload: Co-adaptation between users and technology. Intellectica, 2000. 30(1): p. 177-193.
17. Mackay, W. & Pagani, D. Video Mosaic: Laying out time in a physical space. In Proc. Multimedia, ACM Press (1994), 165-172.
18. Mackay, W., Fayard, A., Frobert, L. & Médini, L. Reinventing the familiar: Exploring an augmented reality design space for air traffic control. In Proc. CHI 1998, ACM Press (1998), 558-565.
19. Mackay, W., Pothier, G., Letondal, C., Bøegh, K. & Sørensen, E. The missing link: augmenting biology laboratory notebooks. In Proc. UIST 2002, ACM Press (2002), 41-50.
20. Rubine, D. Specifying gestures by example. In Proc. SIGGRAPH 1991, ACM Press (1991), 329-337.
21. Sellen, A. & Harper, R. The Myth of the Paperless Office. 2003: MIT Press.
22. Shneiderman, B. Creating creativity: user interfaces for supporting innovation. ACM TOCHI, 2000. 7(1): p. 114-138.
23. Signer, B., Kurmann, U. & Norrie, M. iGesture: A General Gesture Recognition Framework. In Proc. ICDAR (2007), 954-958.
24. Song, H., Guimbretière, F., Hu, C. & Lipson, H. ModelCraft: capturing freehand annotations and edits on physical 3D models. In Proc. UIST 2006, ACM Press (2006), 13-22.
25. Stifelman, L., Arons, B. & Schmandt, C. The audio notebook: paper and pen interaction with structured speech. In Proc. CHI 2001, ACM Press (2001), 182-189.
26. Stroppa, M. Un orchestre synthétique: Remarques sur une notation personnelle, in Le timbre: Métaphores pour la composition. 1991, Bourgois, Paris. p. 485-538.
27. Tabard, A., Mackay, W. & Eastmond, E. From Individual to Collaborative: The Evolution of Prism, a Hybrid Laboratory Notebook. In Proc. CSCW 2008, ACM Press (2008), 569-578.
28. Vinet, H. & Delalande, F. Interfaces homme-machine et création musicale. 1999: Hermes Sciences Public.
29. Weibel, N., Signer, B., Ponti, P. & Norrie, M. PaperProof: A paper-digital proof-editing system. In Proc. CoPADD (2007), 9-21.
30. Wellner, P. Interacting with paper on the DigitalDesk. Communications of the ACM, 1993. 36(7): p. 87-96.
31. Winget, M. Heroic frogs save the bow: Performing musician's annotation and interaction behaviour with written music. In Proc. ISMIR (2006), 73-78.
32. Wobbrock, J., Wilson, A. & Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proc. UIST 2007, ACM Press (2007), 159-168.
33. Wright, M. & Freed, A. Open Sound Control: A new protocol for communicating with sound synthesizers. In Proc. ICMC (1997), 101-104.
34. Yeh, R., Paepcke, A. & Klemmer, S. Iterative design and evaluation of an event architecture for pen-and-paper interfaces. In Proc. UIST 2008, ACM Press (2008), 111-120.
35. Yeh, R., Liao, C., Klemmer, S., Guimbretière, F., Lee, B., Kakaradov, B., Stamberger, J. & Paepcke, A. ButterflyNet: A mobile capture and access system for field biology research. In Proc. CHI 2006, ACM Press (2006), 571-580.