Human Computer Interaction meets Computer Music: The MIDWAY Project

Marcelo M. Wanderley
CIRMMT - McGill University
Montreal, Qc, H3A 1E3, Canada
Marcelo.wanderley@mcgill.ca

Joseph Malloch
Inria; LRI, Université Paris-Sud
CNRS; Université Paris-Saclay
Orsay, France
joseph.malloch@inria.fr

Jérémie Garcia
Goldsmiths, University of London
London SE14 6NW
J.Garcia@gold.ac.uk

Wendy Mackay
Inria; LRI, Université Paris-Sud,
CNRS; Université Paris-Saclay
Orsay, France
mackay@lri.fr

Michel Beaudouin-Lafon
LRI, Université Paris-Sud, CNRS;
Inria; Université Paris-Saclay
Orsay, France
mbl@lri.fr

Stéphane Huot
Inria, Lille, France
stephane.huot@inria.fr

Abstract
This paper describes research carried out in the MIDWAY project at the interface between Computer Music and Human-Computer Interaction (HCI). Our goal is to bridge models and tools from both domains by combining recent developments, and providing a *musical interaction design workbench* to facilitate the exploration and definition of new interactive technologies for both musical creation and performance. Such models and tools can expand the means available for musical expression, as well as provide HCI researchers with a better foundation for the design of tools for "extreme" users.

Author Keywords
Musical Interaction, Human-Computer Interaction, Digital Musical Instruments, Instrumental Interaction.

ACM Classification Keywords
H.5.2. User interfaces (Interaction styles); H.5.5. Sound and music computing (methodologies and techniques);

Introduction
Although Human-Computer Interaction and Music both rely heavily on interactive technologies, their goals sometimes differ. For example, HCI often focuses on improving human performance with interactive

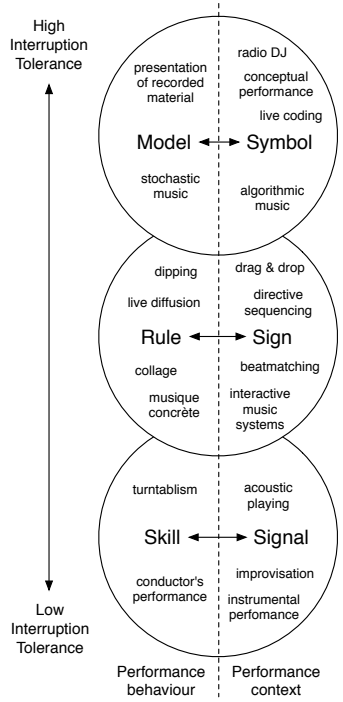

Figure 1: Rasmussen's model applied to musical interaction: the three performance behaviors are: skill-based, rule-based or knowledge- (or model)-based (left column). Performance contexts are shown on the right column. At the bottom, contexts demand close temporal coupling between performer and instrument, with little tolerance for interruption. At the top, contexts have much looser coupling.

technologies, measured in terms of efficiency and accuracy. Music creation lacks such easily quantifiable and measurable goals, and instead focuses on concepts such as creativity, engagement, personalization, and appropriation, all difficult to account for in design.

We believe that the creative context of music provides opportunities for putting cutting-edge HCI models and tools into practice. Expert musicians push the boundaries of system design through their personalization and appropriation of music applications, rendering standard HCI performance measures insufficient for evaluating musical tools. We argue that computer music technology should support and embed existing design models and methodologies, and that new designs built from a well-defined design space will, in turn, facilitate validation and evaluation, as well as reuse and appropriation from other fields, making exploration and extrapolation possible [6].

Methods and Models

HCI researchers have proposed many models to support the design and evaluation of interactive systems. Two are particularly relevant for musical interaction, with direct implications for our work:

Rasmussen's *Human Information Processing* [8,10] is a useful theoretical framework for making sense of the various interaction possibilities in music (Fig. 1). It has implications for the choice of mappings, e.g., from the static and deterministic mappings of digital musical instruments (DMIs) to the dynamic mappings of interactive music systems. It also suggests how to design feedback for performers, including continuous signals for skill-based performance, signs for rule-based interaction, and symbols for model-based behavior.

Beaudouin-Lafon's *Instrumental Interaction* [2] describes how users interact with objects of interest, mediated by 'Interaction Instruments' similar to interaction with physical tools. Instruments are treated as first class objects, and can be defined in terms of *reification*, *polymorphism* and *reuse* [3]. Instrumental interaction relates more directly to musical interaction than earlier HCI models such as direct manipulation.

Even so, musical instruments are usually inflexible: DMIs typically involve static mappings between sensor outputs and sound synthesis inputs that performers must adapt to. Although this limits their use, it allows musicians to master their instruments, sometimes with virtuosic skill. However, musicians also *co-adapt* [7]—not only do they adapt their behavior to the instrument as they learn it, they also adapt or modify it to achieve creative goals. We hope to create *co-adaptive instruments* that are explicitly designed for appropriation by musicians, using rule-based approaches together with the principles of *polymorphism* and *reuse*.

Examples of Applications

We describe two projects for which we designed modular and reusable tools that meet the idiosyncratic needs of musicians. These applications illustrate and support concepts defined in the above models, opening new possibilities for flexible musical applications.

LibMapper & ICon

Input Configurator (*ICon*) [1] and *libmapper* [9] are open-source software tools intended for the design and (re)configuration of modular interactive systems. Both focus on mapping and configuration as a top-level task

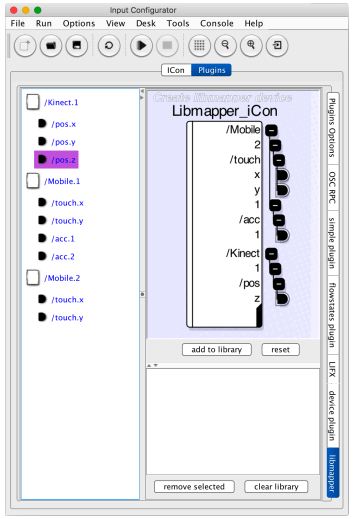

Figure 2: A compound device is created in *ICon* from various *libmapper* signals coming from different physical devices. This enables the construction of instrument models from high-level components; here a device consists of two mobile phones and a Kinect depth camera. Routing parts of the libmapper network through the bridge also allows the use of ICon's library of interaction models and the integration of complex state machines into the mapping configuration.

separate from the task of designing input devices, target applications or media engines.

ICon is an interactive data-flow editor in which various input and interaction resources, such as input devices and communication protocols, can be instantiated as data-flow processing devices, and connected to each other and to interactive applications. *Libmapper* was designed to support the creation of DMIs. Applications and input devices declare their local resources, which are remotely discovered and connected over a local network. Various session-management tools can be used to interact with the resulting distributed network to add, modify or remove connections between producers and consumers of real-time data.

ICon and *libmapper* are complementary. *ICon* features a much richer visual programming interface with a large, extensible library of data processing devices, as well as both data-flow and state-machine approaches for describing and prototyping advanced interaction techniques [1]. It follows the principles of Instrumental Interaction by *reifying* interaction techniques into data-flow processing devices and configurations that can be manipulated as first class objects and applied to other contexts or systems (*polymorphism* and *reuse*). *Libmapper* has similar properties, but its distributed nature also natively supports collaborative design, since an arbitrary number of GUIs can interact with the same mapping network. It also adds support for supervised machine learning tools through the ability to query the value of any signal in the network, including "destination" signals (synth or application inputs).

We have built a prototype bridge between the two tools in order to exploit their complementarity: special *ICon* devices can give access to any *libmapper* signal available on the network. This enables the design of "compound devices" that include signals from a variety of sources gathered together into a logical collection (Fig. 2). By leveraging the benefits of the two approaches, especially their support for the interaction models discussed above, we envision the design of musical applications that treat mappings and interaction as first-class objects.

Paper Substrates and PaperComposer

PaperComposer [5] is an "interface builder" enabling composers to create, manage and use their own interactive paper interfaces with *Paper Substrates*. *Paper Substrates* [4] are interactive paper components that support the creation and manipulation of complex musical data through user-defined representations, and a paper-based interface, using components such as note containers, tone networks or graph paper (Fig. 3).

Instead of radically replacing current tools, *Paper Substrates* extend existing music programming environments that composers are familiar with. *Paper Substrates* rely on existing data processing and storage functions to provide a new layer of user interaction based on handwritten input. They can be seen as *interaction instruments* rather than as independent software components: they modify or create application objects by communicating directly with the application.

Communication is based on the OSC protocol over data channels, allowing flexibility and independence. Other substrates can subscribe to these data channels to create chains of components acting on each other's data or characteristics before redirecting the result to a particular application. For example, a composer can

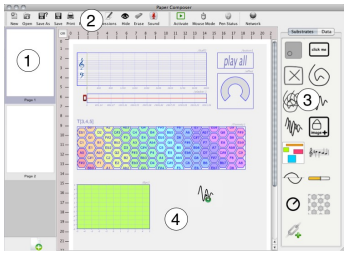

Figure 3: PaperComposer interface. (1) Current document pages. (2) Toolbar to manage document and set parameters. (3) Thumbnails of available component classes can be dragged into the virtual page to create new instances. (4) Virtual page with components: musical staves, play-bar, button, knob slider, tonnetz, curve container.

Figure 4: Interacting on paper to explore computer-based musical processes. Photo H. Raguet © Inria.

combine a component for entering pitches that uses symbolic notation with another that defines the pitches' amplitudes with a curve, to create a more complex object.

Composers can use the modular components of *PaperComposer* to create, customize and use their own paper interfaces for a range of musical creation tasks, such as drawing control curves or writing musical sequences with music notation (Fig. 4). New paper components can be developed and integrated within *PaperComposer* using a *Java* API, and then used and re-used by composers with any compatible application, supporting co-adaptation.

Conclusion and Future Work

The tools introduced above are examples of *technology that embeds knowledge*. By bringing together models and tools from HCI and computer music, they facilitate the exploration and definition of new interactive technologies for musical creation and performance. They have been used by performers, composers and instrument designers, and to support the creation of public performances. We are now working on the next generation of our *design workbench*, which will include more concrete and practical design guidelines. For example, *libmapper* could suggest signal connections based on well-known HCI theories and models such as integrality and separability of input devices or bimanual interaction. *ICon* could provide visual feedback on the properties and "quality" of mappings based on similar theories and models, and support more advanced visual programming tools, e.g. for specifying how a *Paper Substrate* should interpret pen input.

References

1. Appert, C., Huot, S., Dragicevic, P., and Beaudouin-Lafon, M. FlowStates: prototypage d'applications interactives avec des flots de données et des machines à états. *In Proc of ACM/IHM 2009*, 119–128.
2. Beaudouin-Lafon, M. (2000) Instrumental inter-action: an interaction model for designing post-WIMP user interfaces. In *Proc. ACM CHI 2000* pp. 446-453.
3. Beaudouin-Lafon, M. and Mackay, W.E. Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. *In Proc of ACM/AVI 2000*, 102–109.
4. Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. Interactive Paper Substrates to Support Musical Creation. *In Proc of ACM/CHI 2012*, 1825–1828.
5. Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. PaperComposer: Creating Interactive Paper Interfaces for Music Composition. *In Proc of ACM/IHM 2014*, 1–8.
6. Huot, S. 'Designeering Interaction': A Missing Link in the Evolution of Human-Computer Interaction. *Habilitation à Diriger des Recherches*. U. Paris-Sud. 2013.
7. Mackay, W.E. Responding to cognitive overload: Co-adaptation between users and technology. *Intellectica*, 30(1):177-193, 2000.
8. Malloch, J., Birnbaum, D., Sinyor, E., and Wanderley, M.M. Towards a new conceptual framework for digital musical instruments. *In Proc of DAFX 2006*, 49–52.
9. Malloch, J., Sinclair, S., and Wanderley, M.M. Distributed tools for interactive design of heterogeneous signal networks. *Multimedia Tools and Applications 74*(15): 5683–5707, 2014..
10. Rasmussen, J. *Information processing and human-machine interaction: an approach to cognitive engineering.* North-Holland, 1986.