Creative Interaction Paradigms for Machine Learning-Based Design of Gestural Musical Interfaces

Hugo SCURTO[1]

Supervisor: Rebecca FIEBRINK[2]

ARPE Report

October 5th 2015 — August 5th 2016

[1]Department of Physics
École Normale Supérieure de Cachan
61 avenue du Président Wilson
94235 Cachan, Fr

[2]Department of Computing
Goldsmiths, University of London
25 St James's, New Cross
London SE14 6AD, UK

Contents

Acknowledgements

First, I would like to thank my supervisor, Rebecca, for having been so kind and efficient when guiding me all along the year. I am immensely grateful for the invaluable personal and professionnal experience I have accumulated thanks to her, be they working with disabled children at a musical centre, leading hands-on demos with kids at a BBC radio event, or writing and submitting my first scientific paper. Also, I would like to thank Atau and Mick for having invited me to join their regular research meet-ups. My warmest thanks to Adam, who generously let me perform at a Goldsmiths EAVI night with my software, allowing me to merge my passion for music and research on stage. Finally, I thank Jérémie for the great French interludes, and all the members of the Embodied AudioVisual Interaction Group for the very pleasant atmosphere we shared.

I would like to conclude by heartily thanking Frédéric Bevilacqua, who largely contributed to the realization of this research year, and with who I am eager to begin my PhD thesis in September. Lastly, I would like to express my deep gratitude to the ENS Cachan and to the Department of Physics, who have always been considerate of all my specific aspirations: hopefully I will be able to give back what I have received.

1 Introduction

Over the last century, the introduction of technology in musical practices has brought about dramatical changes in human creativity toward sound. Recording devices enabled the infinite replaying of acoustic sounds, and sound synthesis allowed the generation of previously-unheard sounds without needing any physical sound sources. Nowadays, advances in machine learning let us imagine even more innovative prospects for music-making and creativity, and in particular for designing new digital musical instruments controlled with gesture.

Machine learning algorithms are widely recognized for their ability to model relationships in data. An important paradigm shift has been made recently, when researchers found that the interactive use of machine learning algorithms (namely, having users interactively supply data to an algorithm) could turn the latters into design tools, allowing rapid prototyping, iterative modification, and embodied interaction. These actions are deemed essential in the design of new musical instruments, as expressive interfaces mapping gestures to sounds.

Our study will thus focus on the interactive use of machine learning as a creative design tool for gestural musical interfaces. More specifically, we aim at designing novel interaction paradigms with machine learning algorithms, with a particular attention to enhance accessibility and discovery during the design process. A complementary goal is to evaluate these methods in real-world applications, in order to understand how they could provide people with effective mechanisms to better exploit the creative potential of machine learning techniques.

In a first part, we define essential notions of machine learning, mapping, and creativity for non-expert readers, along with giving an overview of related works for both expert and non-expert readers. We conclude this state of the art in a second part, by extracting our study's methodology from it. In a third part, we detail the final implementation of our software, resulting in several design iterations made throughout our study: we propose four data-driven methods to support interactive and creative uses of machine learning as a design tool for gestural musical interfaces.

In a fourth part, we report two successive studies done with a first user group, namely professionnal composers: we give insights on how these methods might be useful for fostering creative processes in compositional practices. In a fifth part, we present work done with a second user group, namely disabled children: two studies helped us understand how this approach might be used by both music therapists and children, gaining accessibility in the design of adapted interfaces. Finally, we discuss future work to be done with both user groups, as well as general prospects in interactive, user-centered machine learning.

2 Background and Related Work

This chapter will first give an overview on human interaction with Machine Learning (ML), starting from basic principles of ML to specific aspects involved within interactivity. We then present fundamental issues and works related to the design of computer music interfaces. Lastly, we review definitions and studies about creativity.

2.1 Interactive Machine Learning

2.1.1 On Machine Learning

Machine learning (ML) is a field of Computer Science that explores the study and construction of algorithms that can learn from, and make predictions on data. Concepts underlying the building of such algorithms mostly derive from probabilistic theory, decision theory, and information theory. In the context of our study, it is not necessary to develop these bases in details: we thus propose a few informal definitions and report the reader to [Bishop, 2006] for a complete overview.

Definitions. The main problem of ML is the building of a function $\mathbf{y}(\mathbf{x})$ the task of which is to take a given vector $\mathbf{x}$ as input, and to generate an output vector $\mathbf{y}$. If the desired output vector is one or more continuous variables, then the task is called *regression*; if the aim is to assign each input vector to one of a finite number of discrete categories, the task is known as *classification*.

The form of the function $\mathbf{y}(\mathbf{x})$ is determined during a *training* phase, in which the parameters of an adaptive model are tuned using a *training dataset*. Applications in which the training data comprises examples of the input vectors along with their corresponding output vectors are known as *supervised learning* problems. Applications in which the training data consists of a set of input vectors without any corresponding output values are known as *unsupervised learning* problems. When the goal of such unsupervised problems is to discover groups of similar examples within the data, it is called *clustering*.

Challenges in ML research. In the last decades, the use of machine learning has dramatically spread throughout computer science and several other scientific fields, mainly because of its defying efficacy to approach or accelerate the solving of given problems.

However, a lot of "folk knowledge" does exist when researchers and developers want to really master the use of ML, even if the theoretical basis is easy to reach in mainstream textbooks [Domingos, 2012]. In the same time, an important part of ML research has been pointed at because of its focus on perfect algorithmic performance on synthetic and identical training data [Wagstaff, 2012]. Thus, there has been a recent trend for ML research to focus on novel uses of existing algorithms, as a way to be efficient on real-world datasets and to remain usable by non-experts as well as by experts.

2.1.2 Human Interaction with ML

At the crossroads between Human Computer Interaction and Machine Learning research, a field has emerged in the last decade. It studies human interaction with machine learning algorithms, where the human does not only choose or modify algorithms and evaluate models, but also builds the training dataset by himself.

Interacting with ML. A first possibility to support user interaction with ML is to allow him to modify algorithms' internal behaviour. In this sense, [Talbot et al., 2009] proposed interactive visualizations of algorithms' internal behaviour to optimize interaction with them. Also, user feedback has been shown to have the potential to significantly improve machine learning systems [Stumpf et al., 2007], the only condition being the ability for algorithms to receive such feedback. In such a situation, it is important to notice that users not only want to provide feedback, but also want to directly guide or *demonstrate* how to proceed tasks [Thomaz and Breazeal, 2008].

Interactive Machine Learning. Another possibility to support user interaction with ML is to give him the possibility to create and edit the training dataset. The first study was initiated by [Fails and Olsen Jr, 2003], who proposed to improve an image classification model by allowing its user to iteratively edit its training dataset and evaluate the newly-created model, based on his expert judgment of how the model should work. Rather than having users dwelling on algorithms' technical aspects, this "Interactive Machine Learning" (IML) paradigm adopts a *data-driven* approach to machine learning: the user directly provides examples of what she wants the system to do, and let a given algorithm generalize from these examples. IML is thus the study of a tight loop between machine learning and the human, where the human fulfills a teaching role for the machine.

This general approach has been successfully applied to several other problems, such as handwriting analysis [Shilman et al., 2006], recommender systems [Amershi et al., 2012], and computer music interface design, which is the central topic of the following section.

2.2 Computer Music Interface Design

2.2.1 Mapping Gestures to Sound

Acoustic instruments can be sketched as devices converting gestural input values (for example, the soft hitting of a given key of a piano) to sonic output values (the corresponding note of the piano at a given volume). Digital musical instruments falls under the same sketch, except that these devices are not constrained by physical laws: thus, every gesture-sound relation can designed in a process called *mapping*, opening new innovative prospects and musical issues.

The "mapping" process. Designing a digital musical instrument implies three elements: (1) an input device, (2) an output module, and (3) a module linking the input device to the output module. The input device can either be chosen among existing gestural controllers, or can be explicitly built; in both cases, this hardware measures user parameters (such as position, acceleration, or hand pressure). The output module can also be chosen among existing sound synthesis algorithms, or can be explicitly built; in both cases, this software receives control parameters (such as volume, pitch, or rhythm). The third and last element, called *mapping* [Hunt and Wanderley, 2002], consists in designing the links between such gestural input values to such sonic output values: this is the focus of our study.

Musical issues. The process of designing a mapping, as introduced by [Schnell and Battier, 2002], is a creative and intentional act of composition. Thus, computer tools for designing mappings should ideally support such compositional processes. In this sense, several properties of interaction have been valued and suggested by professionnal composers, notably encompassing *physicality* and *abstraction* [Fiebrink et al., 2010].

The recent research field of "embodied music cognition" offers a global framework for the study of such physical, corporeal processes (see [Leman, 2008] for a complete overview). It notably highlights the importance of bodily experience in sound perception and cognition. According to this theory, it is primarily through the body that performers convey inner information about their artistic intentions and emotions; this bodily information is encoded into and transmitted by sound to listeners who can in turn attune to the performer's inner intent. Thus, design tools for mapping should support the design of *embodied interactions*.

2.2.2 Machine Learning as Design Tool

On mapping techniques. Historically, computer programming has been a core technique to the design of mappings [Hunt and Wanderley, 2002]. While programming allows a mapping to be specified precisely, the process of translating an intended mapping function to code can be frustrating and time consuming [Fiebrink et al., 2010], even for expert programmers, and it is inaccessible to non-programmers.

Machine learning has been used as an alternative mechanism for generating mappings since the early work of [Lee et al., 1991]. Most work that has employed machine learning for mapping creation has employed *supervised learning* algorithms [Caramiaux and Tanaka, 2013], which can create a mapping from gestural input values to sound synthesis output values using a set of "labeled" training examples. In this labeled training set, each example consists of one vector of gestural input values, plus the "labels"—the vector of sound synthesis parameter values the designer would like to be produced in response to those sensor values. Such methods have paved the way for a switch to an analytical view of mapping design to *interaction-driven* approaches.

Interactive Machine Learning end embodied interactions. Research by Fiebrink and collaborators has focused on understanding the impact of using interactive machine learning (as opposed to programming) on the mapping design process [Fiebrink et al., 2010], and on designing user interfaces to allow instrument builders to use machine learning effectively and efficiently, without requiring prior machine learning expertise [Fiebrink et al., 2011]. Results have suggested that interactive machine learning offers a useful alternative to programming, for instance by making mapping creation *faster*, by enabling designers to encode an *embodied understanding* of the desired gesture-sound relationships in the training examples, and by making mapping *accessible* to non-programmers.

Other research has explored the development of new modeling approaches that are tailored to the design of gesture-sound interaction by demonstration [Françoise et al., 2014]. Such approaches are particularly successful when the task is to recognize and track given gestures. Still, further research is needed to address other fundamental needs of instrument designers, for example, when the user does not have a specific relationship between movement and sound already in mind, or when other properties of the instrument (e.g., playability, comfort) supersede any preference for a particular sensor/sound relationship [Katan et al., 2015]. Thus, new creative approaches to mapping generation might accelerate the discovery and realisation of new design ideas.

2.3 Creativity Support Tools

2.3.1 Designing Creative Softwares

Creativity support tools have been around for as long as people have been creative. As for music instruments, the introduction of the computer has enabled a plethora of new design possibilities for them, as softwares and interfaces allow new types of interaction and reasoning.

On User Experience. First, it is important to underscore that our productivity and creativity when realizing a given task, being researchers, engineers, graphical designers, or musicians, is highly influenced by the tool we use for this task. *User experience*, as the practical, experiential, affective, meaningful and valuable aspects of human–computer interaction, has to be carefully designed in order to enable and facilitate innovative and creative paths of use [Buxton, 2010].

Design principles for Creativity. In this spirit, [Resnick et al., 2005] have summed up twelve general design principles to enhance the creative potential of new softwares and user interfaces. Specifically, such tools should be able to propose many *different alternative* choices, as well as to support iterative design using *prototypes*. They also should be simple to use, compatible with other softwares, and should support different styles and paths of use.

A general example of such an innovative design for iterative prototyping is proposed by [Dow et al., 2010], who found that when people create multiple alternatives in parallel, they produce higher-quality, more-diverse work and experience a greater increase in self-efficacy.

2.3.2 Machine Learning for Musical Creativity

Recently, machine learning has been widely used in creative softwares, either as computer assistants (or feedback agents) or as a way to automatically generate content. Studying algorithms' generativity in creative processes is again another aspect of ML research that is promising.

On Generative processes. ML outputs have the widespread function of being predictions for a given input. However, when one does not aim at making predictions, the same outputs can be used as *creative generated content*. In this spirit, [Kumar et al., 2011] used machine learning to automatically generate alternative Web page designs. It adopts a data-driven approach to the generation of new designs, by combining previously-used design elements to create original page designs that are the outputs of a ML algorithm.

Generating Expressive Mappings. ML-generated mappings have been highly valued for their great *expressivity*, as opposed to explicit mappings [Caramiaux and Tanaka, 2013]. Composers interacting with ML also valued the quick "playing" of newly-created prototypes, and strongly emphasized the creative benefits of *discovering* new sonic gestures when playing with such generated mappings [Fiebrink et al., 2010].

Going further in this direction, [Laguna and Fiebrink, 2014] proposed to use ML to generate mappings either with or without providing training examples, which resulted in more rapid prototypings. Such data-driven approaches to interactive machine learning could be a possibility to exploit algorithms' behaviour in an even more creative and abstract way.

3 General Aim — Methodology

General aim

The aim of this research project is to design and evaluate creative data-driven approaches to users' interaction with machine learning algorithms, in the context of gestural musical interface design. Our wish is to enhance accessibility and discovery during the design process, so that people could better exploit the creative potential of machine learning techniques. We evaluate the benefits of these approaches with professionnal composers, as well as with disabled children.

Motivation

In the previous chapter, we saw that interactive machine learning approaches to mapping generation support well the design of **embodied interactions**. Our first aim is to provide an understanding of how our data-driven methods could improve such design processes. We believe such insights could also benefit other research fields, such as robotics or human-computer interaction.

We also saw that ML was praised for its capacity to generate new outputs, facilitating discovery. A second aim is thus to study to what extent data-driven methods support, or improve such **creative processes**. Overall, such understandings could also inspire more general technology design for creativity, such as graphic design tools or innovative user-centered systems.

Finally, our third aim is to study what methods are better adapted to **real-world applications** with a priori distant user groups: composers, and disabled children. We believe the needs of both user groups could overlap somehow, broadly informing further research and software design.

Methodology

Following a user-centered methodology, we have alternated design, implementation, and user feedback to explore design possibilities, and progressively converge on the creation of a software, its evaluation, and a better comprehension of creative mechanisms supported by our paradigms.

1st step	Design	RR method
$\Downarrow$	Evaluation	Disabled children (workshop)
	Evaluation	Composers (interviews)
2nd step	Design	RC, CR, CC methods
$\Downarrow$	Evaluation	Disabled children (case study)
	Evaluation	Composers (workshop)

Table 3.1: User-centered methodology.

In the present report, we made the choice to describe the final implementation of the software in one chapter, as well as to separate the results for each user group in two different chapters. However, it should be clear for the reader that each part of the work has been informed by other parts, as exposed in the timeline presented in table 3.1. Working in parallel with the two user groups indeed helped us reveal and address several design issues we will discuss in next chapters.

4 Software Description

This chapter reports on the design of the new software used throughout our study. We start by characterizing the software on which the new one builds, called the Wekinator. Then, we present our new data-driven extensions, the design of which was motivated by several iterations through our user studies. We finally describe the user-facing interface of our software.

4.1 On the Wekinator

The Wekinator[1] is an open-source software developed by Fiebrink since 2009. It supports an iterative approach to interactive machine learning (see figure 4.1), by allowing its user to interactively create a training dataset, feed it to a given supervised learning algorithm, and directly evaluate the newly-created model.

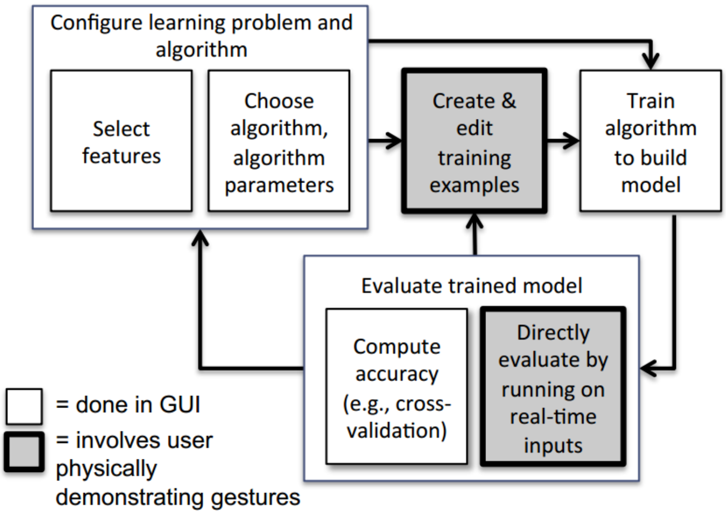

Figure 4.1: The interactive machine learning workflow supported by the Wekinator (image: R. Fiebrink).

The building of the training dataset consists in having a user provide several training examples, by physically demonstrate different inputs in real-time while providing the corresponding output labels. Users can experiment with different algorithms for classification (AdaBoost.M1, J48 decision trees, k-nearest neighbor, and support vector machines) and multilayer perceptron neural networks for regression. The running of the model produces a real-time stream of output values in response to the incoming input values.

Used in a musical context, it allows instrument builders and composers to create training datasets by demonstrating performer gestures alongside the instrument sounds the designer would like to be produced by those gestures. From this data, a given supervised learning algorithm then creates a model that defines a mapping between gestural input values and sonic output values.

[1] www.wekinator.org

Other capabilities of Wekinator are listed below:

- Any type of input controller or sensor system can be used to control sound, provided data about the input is sent as an OSC message [Wright and Freed, 1997].
- Any sound synthesis software can be used to play sound, provided it can receive synthesis parameter vectors as OSC messages.
- The GUI allows users to switch immediately and repeatedly between generating mappings and playing the generated instruments in real-time.
- The GUI allows users to easily change mappings by deleting and adding training examples.
- Advanced or curious users can customise aspects of the machine learning process, e.g., changing the learning algorithm or its parameters, changing the selected features, etc.
- Learning algorithms are set to default configurations that have been shown to work well for many mapping problems, so novice users never have to make an explicit choice of learning algorithm or algorithm parameters.

4.2 Data-driven Extensions

Using the Wekinator, an instrument designer who wants to explore many different prototypes using machine learning must still create many different sets of several training examples, and explicitly choose the type of relationship between gestures and sounds that should be embedded within each set. To creatively enhance the design process, we propose a novel interaction paradigm as well as several different implementations to exploit machine learning generativity.

4.2.1 General Paradigm

We propose a new paradigm for mapping creation, called "grab-and-play mapping" (see figure 4.2), that enables the very rapid creation of new instruments from a very small amount of data communicating some minimal, soft design constraints — namely, the way the user might want to move while playing this new instrument. This minimal set of data allows the creation of mappings which are customised to a controller and/or to a sound module in a loose sense, by aiming for a mapping that is playable using whatever range of motion and types of variation are present in the gestural stream provided by the designer.

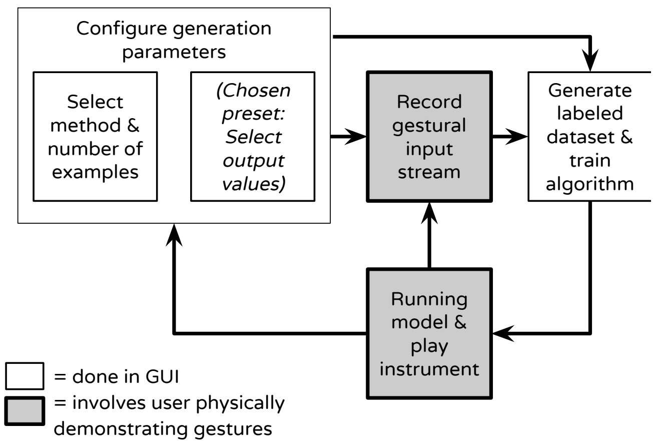

Figure 4.2: Grab-and-play paradigm workflow. The user only has to provide one continuous, unlabeled stream of gestural data to automatically generate a new mapping. Inputs and outputs are respectively drawn from the user's recorded gestural stream, and the sound parameter space..

This process does not require a designer to specify other information about the instrument than this unique gestural stream, other than potentially the range of legal values for each sound synthesis parameter that will be controlled by the mapping. Our approach thus shifts the designer's focus from one of imagining and then implementing new gesture-sound relationships, to a focus on discovering new relationships that have been designed partly by the computer, and on embodied exploration of those relationships.

In this paradigm, the user must first demonstrate how she will physically interact with a musical controller; this results in one recorded, continuous stream of gestural input data. Next, the computer transforms this stream of unlabeled inputs into a labeled training set that feeds a supervised learning algorithm for mapping creation. This transformation is implemented using a number of methods for automatically generating a mapping.

4.2.2 Implementations

Input data: Random selection. A first possibility to select input examples from the user-provided gestural stream is to select one or several vector of input values at random times. Such an implementation is able to adapt a mapping to a user's particular way of moving (for example, a user making small gestures would provide different values than if interacting with large gestures), and also to reveal the most frequent gestural values provided by a user, probabilistically speaking. However, in the same probabilistic perspective, it risks missing important gestural values, or even to select the same values several times.

Input data: Clustering. A more elaborated way to select examples is to apply unsupervised learning algorithms on the user's gestural stream. Specifically, using clustering algorithms is a possibility for identifying the most "important" aspects of a user's way of moving (namely, the clusters), and to select them as input vectors. The benefits could be an even more adapted mapping, rid of the previous method's temporal randomness issues; the user could also make an intentional use of the clusters when recording a gestural stream. Limitations go with usual unsupervised learning limitations, which are the choice of the algorithm and its parameters to obtain subjectively satisfying clusters. We implemented the K-means algorithm as a common centroid-based method, which allow users to specify the number of clusters they wish to be computed.

Output data: Random generation. In our paradigm, the user only provides an unlabeled stream of gestural data. The problem of selecting labels as output vector values is more about how to generate such ones. A first possibility is to randomly generate each parameter from a uniform distribution over the range of all legal parameter values (e.g., [0,1]). This has the advantage of letting the computer make choices in place of the user, thus speeding the design process. In compensation, same or close values may be generated, resulting in an instrument with little sonic possibilities.

Output data: Chosen by the user. Another possibility to generate labels is to let the user choose a given number of output vector values (as "points" in the output space the user would like to be passed through), store them, and successively assign them to the previously-selected input vector values. The advantage is an almost certainty to have the computer scan subjectively interesting parts of the output space; a disadvantage is that it reduces the strength of our rapid approach as it implies the user to make more choices during the design process.

4.2.3 Data-driven methods

Combining these implementations finally bring about 4 different data-driven methods:

RR – *Random Inputs, Random Outputs*
The user almost fully lets the computer design the system.

RC – *Random Inputs, Chosen Outputs*
The user designs what the system will do, but let the computer choose how.

CR – *Clustered Inputs, Random Outputs*
The user designs how to use the system, but not what it will do.

CC – *Clustered Inputs, Chosen Outputs*
The user almost fully designs the system.

4.3 USER-FACING INTERFACE

We implemented our new tool in Java as a branch of the Wekinator software. All code is available online[2]. The new tool adds the following additional functionality to Wekinator:

- Mappings can be generated using the procedure above, requiring only that the user demonstrate a brief input sequence.
- New mappings can be generated indefinitely from the same unlabeled gestural stream.
- The user can change the data-driven method used to generate a labeled training dataset.
- The user can interactively change the number of supervised training examples/clusters generated from the gestural stream.
- The user can switch between the current mapping and the previously built mapping.
- The user can switch between the grab-and-play extension and the original Wekinator setup.

We conceived its interface based on common design heuristics [Nielsen, 1994] (see figure 4.3).

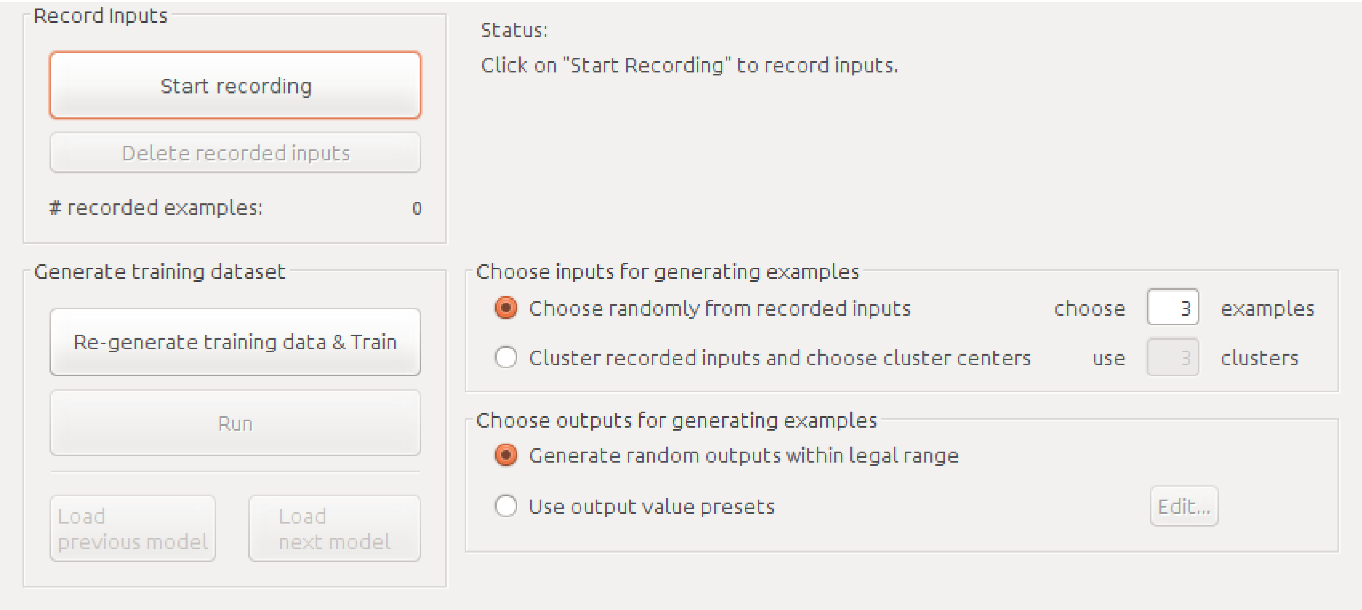

Figure 4.3: User-facing interface. The upper left tab lets the user record a gestural stream. The bottom left tab lets the user generate a new training dataset, build and run the model. The bottom right tabs let the user choose the data-driven method.

[2] http://github.com/hugoscurto/GrabAndPlayWeki

5 Evaluation with Music Composers

The following chapter reports on two studies we led with professionnal composers. We first explain the choice of both this user group and our methodology. Then, we present results from (1) three semi-structured interviews about interacting with the RR method, and (2) a workshop using the four data-driven methods implemented. We finally discuss these results and suggest research prospects that could follow this work.

5.1 Methodology

5.1.1 Motivation

As seen in previous chapters, interactive machine learning has been valued as a design tool among composers. A first challenge is to gain a better understanding of what composers would like to do with such data-driven methods — how they might fit into their practices, and what they would envision as ideal future implementations. Such experiential feedback would inspire our current study as well as future work to be done in the domain of music computing.

In the course of our user-centered methodology, we designed a first data-driven method, then three other ones. A second challenge is to provide an understanding of how these methods could improve the design of embodied interaction — digital musical instruments seen as gestural interfaces. We thus wish to have composers try and compare original Wekinator learning setup with our new interaction paradigms.

Finally, we hypothesize that our four data-driven methods focus more on exploration than on explicit definition. A third challenge is thus to study to what extent they could support or improve creative processes in play among composers — digital musical instruments seen as composed artifacts. Overall, such understandings could also inspire more general technology design for creativity.

5.1.2 Work schedule

These studies follow Fiebrink's early work on using interactive machine learning to design musical instruments. We had the opportunity to collaborate with other members of the Department of Computing, as well as with Goldsmiths' Department of Music. We first planned semi-structured interviews with 3 composers using our RR method. The collected feedback informed the design of RC, CR, and CC method, along with other improvements that were not developed in the context of our project. We then organized a workshop with the help of the Department of Music: 8 composers tried and experimented our 4 data-driven methods to build custom musical instruments.

5.2 User Studies

5.2.1 Interviews

We report on interviews held with three professional computer musicians using the RR method. Our wish is to analyze how our grab-and-play approach could influence their music practice and/or composition processes, and to inspire future implementations.

Interview setup

Participants. We held individual interviews with three professional computer musicians. All three were composers and performers, as well as active in teaching computer music at university level. One reported previous experience with the Wekinator. We hoped to gather feedback to better understand how our grab-and-play approach could support embodied exploration processes and rapid mapping generation. We also aimed to collect information on ways to improve our first RR method. For instance, we wondered how much control the random generation method would leave to composers.

Structure. Each interview was a 30-minute exchange in which experimentation with RR method alternated with semi-structured interview questions. The musician was presented with a one-stringed GameTrak which allows the sensing of a user's 3D hand position (see figure 5.1), while we controlled the computer GUI and led the interview. Experimentation started with our grab-and-play paradigm, spanning regression and classification algorithms; it ended with the original supervised learning setup, using the same regression and classification algorithms. When first trying the grab-and-play setup, composers were not told about its implementation: they thus had no presuppositions when experimenting with it. They were asked about their playing strategies and how they thought it was working. Then, they were asked about ways they could imagine improving this RR method. Finally, they used the original Wekinator supervised learning setup, allowing them to experiment and compare the two approaches. For regression, we used a digital synthesis instrument based on similarities between physical models of the flute and electric guitar [Trueman and DuBois, 2002], potentially allowing for vast sound space exploration. Experimentation with classification relied on a sample trigger implemented in Max/MSP, playing pre-recorded sound samples in a "funk" style.

Observational study

Grab-and-play setup. The exploratory aspect of our grab-and-play approach was praised by the three composers. One of them described the system as *"kind of an enigma to solve"*, and was interested in the fact that *"it kind of challenges you: you have to explore a bit, and try to understand the criteria, or how to deal with these criteria"* to perform with it. Also, the possibility of rapidly prototyping a new instrument allowed them to experiment with very different gestural and sonic interactions. Using the same recorded gestural stream to build two instruments with the RR method, one composer reported when comparing their playing that *"[he doesn't] feel any consistency between them in terms of gesture and sound: they felt like completely different mappings"*, saying he could explore them *"endlessly"*.

Different strategies were adopted to exploit the system's capabilities. One composer first spent time exploring the sonic parameter space, then tried to regain control and to replicate certain sounds. He then decided to reduce the space he was exploring by moving the controller in a given plane rather than in 3D, allowing him to learn certain gesture-to-sound relationships in a *"pleasant"* way. In this sense, one composer reported he could eventually learn how to play such an instrument. After having been told gestural data was randomly selected, one composer tried to exploit this aspect by spending more time in certain locations in his gestural space to increase the likelihood of their inclusion in the mapping. He indicated he was interested in *"playing more"* with this exploit.

The random selection also had some weaknesses: for example, a composer reported he had too little gestural space to explore between two interesting sounds in a given mapping. Another composer said he would require more control over the selection of sound parameters while agreeing that randomly selecting could *"definitely"* go with his vision of composing (*"the embodiment of being able to control the sound with enough level of control, regardless of what the movement is"*). Ways to modify a given mapping would be required as an improvement (this is discussed in section 5.2.1).

Original Wekinator setup. When testing the original Wekinator setup, one composer underlined its effect on his expectation of how a given instrument would work: *"it sets up all the run expectations, and it also affects the way I play it, because now I start to remember these poses, rather than just exploring in an open-ended way"*. Choosing gestures when building a mapping can thus be a responsibility composers want to avoid when creating meaning through sound. In this sense, a composer even mentioned that he *"never care[s] about gesture"* in composition, rather seeing these gestures as movements that are related to his own instrument practice: *"actually, what I care about is the exploration process afterwards"*.

On the other hand, one composer liked the fact that he could immediately replicate a given sound as he *"kind of see[s] what's being mapped there"*. He enjoyed the idea of spending less time on exploration and having more control, as *"in some kind of performance, you want to be very meticulous"*. Comparing the grab-and-play and original Wekinator setups, composers seemed to agree that *"both are useful"*, depending on what they would want to achieve. *"If you set up the mapping yourself, and the system yourself, you have more control, but then again maybe it's too predictable"*, one composer summed up.

Suggestions for improvement. Talking about ways to improve such setups, one composer evoked the idea of *"a hybrid approach"*, where one could record specific gesture-sound relationships and add some randomness in between: *"some points could be manually controlled, and some points automatically"*. This would be a way to address the previously-mentioned trade-off between control and exploration: one could then explore and discover the control space during performance, while having access to predetermined gesture-sound relationships in the mapping.

The random selection was praised for its rapidity in prototyping and experimenting, as for *"most trainings, actually, you're not really so concerned about the specific thing that's done: you just want stuff mapped out"*. However, composers would like to have a bit more control over both gesture and sound when building such a mapping. In this sense, one could imagine clever ways to select gestural and sound parameters that would still enable rapid instrument prototyping. Going further, one composer suggested incorporating the design process within the performance. Instead of being *"a static thing"*, the design process would become a real-time evolution of one's control space (*"me creating the control space in real-time"*). For example, such a performance could entail repeating the same gesture to tell the machine to add new sounds to this gesture. This idea is reminiscent of Fiebrink's play-along mapping approach [Fiebrink et al., 2009].

Finally, one composer noticed the difficulty in editing a newly-generated mapping: *"It's really frustrating when you're working musically because you just want to tweak that thing, and then the whole thing blows up"*. One could edit the training data, or, as the composer suggested, *"regression is just a geometry, so why can't we just start stretching things and manipulate them?"* Designing a user interface that allows the intuitive modification of an N-dimensional geometry would be necessary; however, this goes beyond the scope of our grab-and-play mapping paradigm.

Discussion

These individual interviews have clarified what kind of compositional processes could be allowed with our grab-and-play approach. Composers' opinions globally corresponded to our intuitions about the discovery and exploration processes encouraged by the RR method implemented in our tool. As mentioned by one composer, such a random process may be used when starting a piece, as a way to let new ideas emerge, then opening up a reflection on how to use them: quoting him, *"all these mapping processes are about making decisions that are rational: it's just building blocks. Then, musical decisions come as you actually walk through them..."*

As future improvements, a possibility would be to generate input data that are more equally spread through the space delimited by user's gestural extrema. The choice of output labels could also be informed by the relationship between synthesis parameters and higher-level perceptual characteristics, enabling the creation of instruments capable of accessing a desired perceptual sound space. Hybrid approaches mixing grab-and-play mapping with user-provided pairs of inputs and outputs could also be a way to encourage exploration while allowing customization.

In the context of our study and in order to remain general and applicable to other fields than music, we imagined other implementations of our grab-and-play paradigm that may also support composers' needs. For example, clustering gestural data could meet composers' need for control over their gestural space in relation to sound, while allowing rapid prototyping. Also, most composers wanted to have more control over the choice of sounds: adding a customizable list of possible output labels could alow them to do so. Such improvements allowed us to implement RC, CR, and CC methods, which we then tested and evaluated in a second study.

5.2.2 Workshop

We report on a workshop held with composers, using RR, RC, CR, and CC methods. Our wish is to analyze what data-driven methods could support embodiment and/or creativity processes, and to gain a deeper understanding of differences between Wekinator and grab-and-play setup.

Workshop setup

Participants. We held a hands-on workshop with seven practicing composers, students, and researchers in computer music. All of them reported previous experience with original Wekinator in their own practices. We thus had no difficulties in explaining them differences between Wekinator and our grab-and-play approach.

Structure. The workshop lasted 2 hours and included three parts: (1) a brief presentation of our software along with the workshop's aims, (2) hands-on, free experimentation with the software, and (3) group discussion on the software along with questionnaire filling. The presentation included brief explanations on how data is selected using the four different implementations of the software. Participants were allowed to use a variety of gestural input devices, such as a face-tracker, a 10x10 webcam grid, a leap motion sensing fingertips air position, previously-mentioned GameTrak 3D stringed hands sensor, and mouse controllers. They mostly experimented with regression using continuous sound synthesis modules, such as FM synthesis, or previously-mentioned digital synthesis instrument based on similarities between physical models of the flute and electric guitar. Some also used classification along with drum machine modules, or custom-built sound modules. Hands-on experimentation and generation parameter modifications alternated with informal exchanges between us and them. Finally, group discussion allowed us to share our views on model-building with our grab-and-play tool. We logged participants' data for software interaction, as well as gestural data built models.

Findings

Model-Building. Participants built an average of 21 different models during the workshop (σ = 9.5). They experimented with all four implementations, with a preference for random inputs and outputs over respectively clustering and chosen presets. This resulted in 56% of models built with the RR method, 24% with the CR method, 7% with the RC method, and 13% with the CC method, and could be interpreted as implicit evidence for participants understanding the methods.

To build these models, participants recorded an average of 13 new gestural inputs (σ = 7.7). They built 2.5 times more new models than they effectively used these new gestural inputs (σ = 0.86). This means they made a consequent use of our software's ability to re-generate new data to build new models. On the opposite, participants changed the number of examples an average of 2.5 times over the workshop (σ = 2.1), which is quite low. Questionnaire data confirm this statement, as most participants have not really figured out how helpful changing the number of examples could be: rather, they mostly focused on modifying methods and evaluate them interactively.

Embodiment and Human Learning. When using our grab-and-play tool, running models was the only way for users to get feedback on how dataset modification changed algorithms' behaviour. By embodiedly interacting with newly-built musical instruments, participants learned how the system worked, and were able to provide data that allowed a more effective use of the system. Three participants thus began exploring the grab-and-play tool by recording continuous movements that reflected their range of motion: one made *"random moves"*, and another *"played like a silent instrument, like a sequence of any instrument but with no sound"*. When using the cluster implementation for gestural inputs, some participants tried to match the number of cluster with the number of gestures they performed during the recording step. For example, one participant *"aimed for some stationary points"* using the leap motion controller; another also demonstrated multiple examples of many gestures, *"[without] car[ing] about transitioning"*, which indeed helps the clustering algorithm to find patterns in gestural data. Interestingly, this sequencing of example gestures when using the clustering implementation was reproduced by one participant with the random implementation, as he *"discovered that he could also demonstrate clusters when using random ones"*, manipulating the random process by changing the number of drawn examples.

The grab-and-play tool also affected users' expectation on the software, switching from precision-related practices to open-ended learning of newly-created gesture-sound relationships. A participant liked the fact that the tool *"forces you to switch the states you have planned, a bit"* (for example, the 'relaxed' status of one of his instruments did not match the one he had in mind). By embodiedy interacting with the model, he was able to match gestures with sounds, thus learning how to play the instrument. Going further in this sense, another participant said the software would fit improvisional practices, where gestural data would evolve as he would interact with the model.

Trade-offs between exploration and explicit definition. Predictably, our grab-and-play tool was praised by all participants for its invitation to exporation and discovery of new design possibilities. We were interested in understanding the pros and cons of each data-driven method we implemented.

As foreseen in section 5.2.1, the RR method was praised by participants to explore the design space available by their setup: one participant said he would *"retrain, retrain, and retrain"* using the same recorded stream to explore design possibiities. On the other hand, the CC method allowed more objective-oriented design processes, as participants were explicitly choosing given sounds and segmenting their gestural stream during the recording step. One participant explained that after a first experimentation step with randomly-generated models, *"as [his] ideas became more refined, [he] would probably move to clusters and presets"*. However, as the CC method does not allow to specify which gesture goes with which sound, it is still proposing unforeseen outcomes (see users' expectation in previous paragraph) and do not propose the same degree of customization than the original Wekinator.

Our observations also suggest a wider, more creative use of all methods than initially expected. For example, at the end of the workshop, participants were explicitly asked to propose a solution for designing an instrument that would make a higher pitch when they do a given gesture. As expected, they all proposed to use the preset implementation for outputs. Interestingly, only two of them mentioned the use of the clustering implementation for choosing gestural inputs; others did not mentioned any specific implementation. This could suggest that participants rely on exploring their gestural input space, and then learning how to produce desired sounds, rather than making precise design choice using the cluster implementation. One participant analyzed: *"With the cluster [implementation], it was more like, you could actually pick certain positions. But with the random, if you picked certain positions it was not that precise, which was also kind of cool..."*

Participants were also explicitly asked to propose a solution for quickly discovering design ideas: five of them proposed the random implementation for both inputs and outputs, and two of them also proposed to cluster inputs and to use preset outputs. One participant propose the latter method as a way to explore output space in-between defined points, thus allowing the discovery of new sounds. This is very reminiscent of the original Wekinator results on exploration processes [Fiebrink et al., 2010]. Eventually, this could suggest that the rapid prototyping allowed by the grab-and-play tool made it appear as a tool globally aiming at exploration processes.

Overall, both implementations for choosing gestural input data fitted exploration practices while allowing soft definition: control was made possible within data provided by users. On the other hand, it was much clearer to distinguish between the two proposed implementations for choosing sound output data: randomness allowed more discovery (a participant even mentioned *"surprising"* sonic results), while choosing presets drastically reduced the output space. Combining these implementations enabled different degrees of freedom in exploration and definition that all participants seemed to have understood.

Global evaluation. Globally, all participants enjoyed using our grab-and-play tool, and were able to propose diverse usages in which it may be useful. At the conclusion of the worshop, participants highly agreed that they were able to effectively experiment with the different options provided in the grab-and-play extension (mean = 8.7 on a 10-point scale). They also strongly agreed that they understood the difference in how to use Wekinator versus its grab-and-play extension (mean = 9.3). About data-driven methods, participants highly agreed that they understood the difference in how to use cluster versus random implementation for gestural inputs (mean = 8.3), as well as chosen preset versus random implementation for sound outputs (mean = 9.5).

Discussion

Design Processes in Data-driven Contexts. Our observations suggest that our grab-and-play approach may favour a focus on performing to the design of embodied interactions. Several participants mentioned that the software would fit live performance contexts, in which they would need *"a more rapid approach to building sounds"*. One participant liked the concept of playing without sound during the recording step, *"which is using gestural instruments like dance, ballet, etc."* He suggested that *"letting the computer do the rest enables to focus on performing"*, and was very enthusiastic about using it in a project with improvising dancers to experiment with the impact of the first recorded gestural input: *"after recording a few different models with the extension, I could sense association in the feeling of the instrument, although changes were made"*.

As we have discussed, our grab-and-play approach may be useful to quickly explore embodied design ideas. One participant analyzed the souftware could be an effective way to *"assess the potential"* of both gestural input devices and sound synthesis modules, in *"early explorative stages"*. He distinguished between grab-and-play *"ideation"* and original Wekinator *"realization"* aspects to designing embodied interactions. Most participants indeed agreed the Wekinator was much more adapted to *"establish some exact instrument like a synthesizer using knobs"*, or any musical projects where they have *"particular gestures"*, *"particular sounds"*, and *"clear mappings in mind"*.

Finally, using the grab-and-play approach may also be useful when users do not have specific design constraints in embodied interactions. Two participants agreed on the fact they would use it when they would not be *"too bothered about precise inputs"*, allowing *"freeform experimentation"* with gestural input devices. Going further, two participants suggested the cluster implementation for inputs could be *"very useful"* for experimenting with specific types of gestural data in an interactive machine learning setup, namely high-dimensional data and high-frame rate data (such as electromyography signals). On the other hand, as mentioned by another participant, the original Wekinator would better fit projects working on models of specific motions such as *"walking, running, jogging, etc."*

Exploiting the Creative Potential of ML Algorithms. Our observations also suggest that our data-driven approaches to the building of embodied interactions may encourage creativity at different scales. Interactively feeding random data to machine learning algorithms has benefited participants with discoveries: randomness may be *"essential to create room for creativity"*, as coined by one participant. Another participant went beyond this, stating that random outcomes generated by machine learning could later be re-used in the original Wekinator setup. This suggests that the random implementation could allow users to interatively, iteratively discover and refine their design space in a longer-scale machine learning-based design process.

As previously seen, the clustering implementation for gestural inputs may also enable creative discoveries. First, participants sometimes used this implementation without segmenting their recorded stream: this resulted in an embodied exploration of unsupervised machine learning outputs, as they tried to discover clusters from their own gestural data. Additionnally, if users feed the clustering implementation with specific gestural data, and combine it with the preset implementation for sound outputs, they still have to find which sounds are associated with which gestures. This may allow for a gamified mapping design process, in which users interactively discover new gesture-sound relationships. Such a process could in turn be followed by exploring supervised machine learning interpolation between defined points in a way more similar to what the original Wekinator enables.

Overall, creative softwares are often designed with the goal of explicitly defining and creating given outcomes through precise options and actions. The very essence of our grab-and-play software does not follow such a rule, but rather proposes soft design constraints explicitly aiming for more open-ended outcomes and less controlled actions. Interestingly, our observations suggest participants did not bother that much from facing such a change in paradigm: rather, they seemed to find it quite adapted to some of their creative practices. One participant said that *"the fact that [the software] is still quite random is pretty cool, because i'm not that into precision"*. Thus, the autonomous-oriented philosophy of our tool in building mappings may be of interest for supporting embodied musical interactivity. Eventually, improving real-time autonomy of such a tool could provide users with even more creative, effective tools, for example by designing machine learning algorithms able to constantly adapt to new gestural inputs.

5.3 General Discussion

We explored novel data-driven approaches to instrument prototyping and music-making among music composers. Initial interviews using a random implementation to select both gestural inputs and sound outputs suggested three other data-driven method for our software, while highlighting global advantages of a grab-and-play approach to mapping generation. The four methods were then tested and evaluated during a workshop with computer music practitioners. From these observations, we conclude that all methods supported the discovery of new embodied design ideas, with each methods covering different degrees of freedom in precision and exploration. We also showed the potential of working on data-driven approaches to foster creative uses of already existing machine learning algorithms: giving more place to human direct exploration and evaluation may facilitate more pleasant, productive design processes to which such interactive approaches to machine learning seem able to support.

Further research could investigate other implementations for selecting both gestural inputs and sound outputs, as suggested in section 5.2.1. Other possibilities would be to use such data-driven methods in scenario-based design of gestural musical interfaces, where the computer would automatically generate a new mapping, have the user evaluate it, then proposing multiple-choice questions for modifying the current design. Such an approach could reduce the number of options modified consciously by the user, allowing him to focus even more on embodied, interactive exploraton rather than explicit definition. Lastly, interactive machine learning systems offer prospects for adapting existing machine learning algorithms to better fit human interaction. Going further in the scenario-based option, a possibility would be to enable users to design interfaces continuously: unsupervised learning approaches could allow a model's constant adaptation to users' way of using it. Such approaches could in turn be applied to the design of other kinds of tangible interfaces (such as multi-touch screens), or even to other application fields (such as movement rehabilitation), leading to potential impact for real-world innovative uses of machine learning.

Figure 5.1: Setup for using our grab-and-play software with a GameTrak, a webcam-based face-tracker, or a mouse tracker.

6 Working with Disabled Children

The following chapter reports on an exploratory study applying data-driven approaches with disabled children. We first motivate the choice of both this user group and our methodology. Then, we present results from (1) a workshop with disabled children to compare grab-and-play and Wekinator setup, and (2) a case study using grab-and-play setup with Jan, a music therapist working with disabled children. Finally, we discuss results and suggest future research prospects.

6.1 Methodology

6.1.1 Motivation

Interactive machine learning has been recently applied to build custom musical interfaces for disabled adults through several workshops [Katan et al., 2015]. Not only did the authors of that work find similarities between the musical goals and practices of disabled people and expert composers, but they also noted some difficulties for participants to develop a memorable gestural vocabulary. A first challenge is thus to understand whether our grab-and-play approach might circumvent some user frustration, by explicitly inviting exploration of new instruments rather than suggesting that gesture design and memorisation are important.

A second challenge is to meet the needs of everyone musically interacting with disabled children — namely, music therapists, but also health care providers and relatives. Research on technology-mediated activities for music therapy has been done for several decades; however, our data-driven approach to music practice and instrument design remains unprecedented. Our wish is thus to provide music therapists with innovative, accessible machine learning-based tools for building adapted musical interfaces. To come to a successful design and use of the tool implies investigating music therapists' practices with disabled children, as therapists and children would differ in their use and understanding of the tool.

6.1.2 Work schedule

We had the opportunity to take part in a two-year Musical Inclusion Programme[1] (managed by Simon Steptoe for Northamptonshire Music and Performing Arts Trust, and funded by Youth Music and the Paul Hamlyn Foundation), one of the aims of which is to explore creative ways in which new music technologies could enhance disabled children's musical and social inclusion. During this research project, we were able to participate to one "relaxed sing-along" session in which we led an exploratory workshop (in section 6.2.1). Then, we went two times in a primary school where a music therapist, Jan, let us lead a case study on her musical practices with disabled children (in section 6.2.2).

[1] http://www.nmpat.co.uk/music-education-hub/Pages/musical-inclusion-programme.aspx

6.2 User Studies

6.2.1 Workshop

We first led a workshop with disabled young people to gain a preliminary understanding of how our grab-and-play paradigm might be useful for building new musical instruments for people with disabilities, and of how youth might respond to the customised yet unpredictable mappings built by this tool.

Workshop setup

Participants. The young people we worked with were part of a "relaxed sing-along" session, one of the aims of which is to help disadvantaged young people take part in collective and relatively simple musical activities. “Disadvantaged” stands for a broad variety of living conditions, ranging from health, behavior or learning disorders to looked-after children. Such young people may not have the opportunity to access high-quality musical activities, thus preventing them from the benefits music can provide in a social context. Bespoke digital musical instruments have the potential to make music-making easier and more fun for many of these youth. It is also possible that using personalised instruments may reduce social pressure, since the mapping function is unique to each user. By emphasising participation as a process of exploration of instruments and sound rather than performing a piece of music correctly, we also hoped to make the experience fun and inclusive for everyone.

The 15 youth we worked with all had physical and/or mental disabilities. They were accompanied by their parents or guardians, and their level of concentration was variable depending on their disabilities.

Workshop structure. The workshop was a one-hour session during which each of the two workshop leaders led a sequence of small-group sessions with one youth participant and their parent/guardian(s).

The input device used was a GameTrak Real World Golf controller, which senses 3D position of the user’s hands using two strings. Sound was generated by Max/MSP. The following setups were available to the participants:

- Grab-and-play RR method with classification, for triggering pre-recorded sound samples, in a “funk” style.
- Grab-and-play RR method with regression, for controlling audio effects (pitch shifting and reverb).
- The same sample triggering and effects control as above, but using Wekinator’s existing supervised learning interfaces for classification and regression (i.e., requiring users to specify labeled training examples).

In each small group, the workshop leader controlled the computer (including the GUI for mapping creation), and the youth participant was given the input controller (sometimes with the help of parent/guardians). Participants therefore did not have to learn to use the GUI or other software. All participants tried at least one grab-and-play mapping, and participants who had time and expressed interest also tried supervised learning mapping.

Observational study

Grab-and-play setup. Our grab-and-play approach was very useful to build adapted interfaces. It allowed us to quickly build instruments whose gestural range was wholly dependent on the participant: during the recording step, some people made wide movements, while others with strong motor disabilities were only able to make small movements. In this sense, the adaptivity of our tool prevented it from building non-playable instrument for a given person. Some participants also seemed to find the exploratory side of the RR method very fun. They spent a lot of time trying to find where the different audio samples were in their gestural space: this activity seemed to capture participants' attention, as they usually seemed to engage in choosing which sample to trigger.

Grab-and-play classification seemed to elicit different types of interaction compared to regression. People using classification focused on triggering known sounds, whereas people using regression focused on exploration (alternating between searching for new sounds and playing them back). Both approaches thus have their own pros and cons, depending on which musical activity people and carers want to take part in.

Original Wekinator setup. Participants who had enough concentration also tried the supervised learning setup. They first recorded different GameTrak positions for each of the four classes of samples, and then tried their instrument. Several participants reported that they liked being able to choose where to place the audio samples in their gestural space, giving them even more control on what was going on. However, it was hard for some participants to concentrate on the process of choosing different gestures to trigger different samples. Even if the customization of the interface was enjoyed by some participants, it was not necessary to support meaningful musical experiences for most participants.

Both classification and regression were understood by participants, as they knew which audio effect to expect since they had chosen them during the recording step.

Discussion

This preliminary workshop has shown the utility of our grab-and-play approach to build custom musical interfaces. Our observations show this approach can be useful to build personalised devices, both for participants that were not able to concentrate for a long time, and for participants with specific motor disabilities. Also, the randomness of the RR method did not seem to interfere with disabled children's musical experience: on the contrary, it allowed even more rapid prototyping of exploratory instruments.

As future improvements, other input devices (such as joysticks, Wiimotes, or dance maps) as well as other output programs (such as instrument-specific samples or visual outputs) could be used to design instruments that are even better customised to each participant. Further, the social aspect of collective musical practices could be investigated through grab-and-play mapping, for example by having different young people exchanging their newly-created models, or more simply by having teachers sing with young people's sonic outputs.

These observations stand as an initial validation of our hypothesis on facilitating interaction for disabled children. While we saw the software allowed rapid adaptation and engaging musical activities, we did not let children use the software themselves, most of the time because their motor and/or cognitive disabilities did not allow them to do so. In other musical activities, they mostly interact with and through expert care providers and therapists. Studying music therapists' interaction with disabled children thus forms a crucial part for adapting our software to such an application domain: this is the main focus of the following of our study.

6.2.2 Case study

In the context of the Musical Inclusion Programme, we had the opportunity to collaborate with Jan, a long time music therapist working in a primary school. At the writing of the report, we did two day-long sessions there to observe her interact with disabled children.

Methodology

In our project, we wish to support more engaging and personalized musical experiences for disabled children. We do not aim at addressing a given motor or cognitive disease: rather, we are interested in facilitating interaction between several different disabled children and their caregivers/music therapists. We believe that supporting personalized, unique musical experiences, as our grab-and-play tool does, could provide disabled children with a privileged area to express his/her emotions, as well as having positive effects on the whole therapy team.

Addressing such issues implies adopting a different methodology from the one we employed with our previous user group (namely, music composers). We thus decided to focus more on the question of how to support great interaction between the music therapist and disabled children, rather than trying to understand the pros and cons of our 4 previously-designed methods. During our two day-long sessions at the primary school, we combined unstructured interviews with Jan, observational sessions of her regularly interacting with children, interactive sessions in which we helped Jan use our RR method with children, and a reduced ethnographic inquiry with caregivers. All of this acted as indirect participatory design activities, as it informed the design of novel sound modules and input devices throughout the study.

Observing Jan interacting with five different disabled children gave us precious feedback on her practices. It is arguable whether such observations could be coined "one" case study (studying Jan's interaction), or "five" case studies (studying each children's interaction). Also, our study does not cover long-term effects in learning processes, as well as larger group studies (both for music therapists and disabled children), leading in each case to discussions about results consistency and generalization. Here, we do not intend to generalize our results: however, we do intend to provide ground work for further research in related areas, exploring different disabilities through the lens of one music therapist.

About Jan

Jan is a music therapist who is registered with the Health and Care Professions Council (HCPC): she has been working regularly on one-to-one session with a few children with strong cognitive and/or motor disabilities. As a music therapist, her main focus is not about teaching music skills or music theory to disabled children, but rather to help them develop more general communication skills through non-verbal musical activities. Not only does the musical space act as a recreational and releasing medium for such disabled children, but it also helps the therapist to diagnose specific disabilities, as well as to discover unforeseen abilities.

In the school's room she is working in, there are several acoustic instruments — such as a piano, a flute, a drums' tom and cymbal, xylophone bars, bells, or tambourines. Nevertheless, the acoustic instrument she is using the most is her voice. Jan makes an extensive use of her voice to imitate disabled children's vocalizations and musical sounds, as a way to reflect childrens' behaviours and emotional states. Such improvised interaction mechanisms are crucial in her work: they foster attention, eye-contact, and listening abilities among disabled children. Imitating children also leads them to imitate her, allowing them to develop new vocal and gestural vocabulary. Overall, using non-verbal sound as medium helps such children to get rid of their usual fears towards interacting with others: the music therapist thus aims at giving a sense of control to disabled children though sound, building trust and confidence in expressing theirselves.

In the middle of Jan's instruments were very few digital musical instruments. An exception was an old portable Yamaha synthesizer — other electric devices being a camera, a 2000s portable computer, and a huge amplifier equipped with a cassette reader. When using the synthesizer, Jan focused more on her improvisation patterns with the children than on the range of different sounds made accessible by the synthesizer; in other words, her use did not make it different from any other acoustic instrument of the room. We were thus very surprised to see, put aside in the room, two cutting-edge electronic devices, namely an Eye-Tracker and a Skoog. The Eye-Tracker was mostly used by Rebecca — another caregiver from the same school — in the context of gaming activities, but was not used by Jan as the school did not have installed music softwares adapted to it.. The Skoog, a USB-plug soft cube designed as an accessible digital musical instrument for disabled people, was not used by Jan or the carers because they were not trained to use it. More generally, we observed a lack of experience with computer-mediated activities among carers, as well as a reluctance to spend time delving into instruction manuals.

When asked about what she would like to do ideally with the children, Jan did not come out with precise design possibilities, but was rather curious about anything that could provoke *"different types of sound and gestures"* among disabled children, *"other than just hitting a keyboard"*. In other words, she seemed extremely open to try new approaches to her musical practices, and also excited to actively participate to their design. As a consequence, we were able to collaborate with her very easily, which facilitated our understanding of her practices. Overall, what seemed to be her main focus when remembering other previous successful musical activities was the sense of inclusion through music. Her wish was to enable those who cannot sing, or even who can just make one sound, to benefit from the social aspect of music: as she noted, *"it is great* [for disabled children] *to play with other people, even with only 2 or 3 notes"*. Giving disabled children more control over sound is thus or prior interest for her practice.

Sessions setup

As mentioned above, we held two day-long sessions at Jan's workplace. During the first session, we mostly focused on trying different input devices and sound modules to collect Jan's feedback on it after using them with disabled children. Input devices were a GameTrak Real World Golf controller that senses 3D position of the user's hands using two strings, and webcam-based controllers (a 10x10 color grid, and a face tracker). Sound modules were the previously described Sample Trigger, a drum machine patch, a sinus generator, a FM synthesis patch, and a digital synthesis instrument based on similarities between physical models of the flute and electric guitar. Our laptop was equipped with the RR method from our grab-and-play tool.

The second session focused more on observing Jan interacting with disabled children. We only used the GameTrak stringed controller, allowing Jan to spend more time working with a given configuration. Thanks to feedback from the first session, we implemented two sound modules. The first was a sample trigger allowing real-time recording of sound samples, and replaying them at different pitches; the second one was a continuous looper allowing real-time recording, freezing, and pitch shifting of sound samples. Their design — notably the possibility to record sounds in real-time — was inspired by first session's observations, as we discussed with Jan and other caregivers the importance for disabled children to reproduce real-world sounds, or sounds that they could recognize. As said before, we kept on using the RR method to be consistent with the first session, thus reducing the risk of misleadng Jan in her practices, while letting us focus on interaction rather than method comparison.

In both sessions, we fully controlled the GUI (notably recording disabled children gestural streams), and alternated between helping Jan in adjusting instrumental setups and observing her interact regularly with the children (regularly meaning by herself or with caregivers). Our wish was to focus on interaction processes rather than on designing the interface Jan or other therapists could use on a longer scale (this next design step will be discussed in section 6.2.2, *Discussion*).

Interacting with disabled children

The first children we worked with was Ines[2]. Among other cognitive disabilities, Ines has concentration problems, and does the same agitated gesture with both arms when getting excited in a given activity. We equipped her with the GameTrak controller on both her hands and rapidly recorded her gestural stream. When we used different continuous sound modules for each hand, she seemed to be very surprised; Jan happily noticed a shift in Ines's attention. Indeed, while she was performing her agitated gesture when playing the synthesizer, she began making smaller and slower gestures with only one hand, with a surprised expression on her face. She then made faster gestures, alternating between slow "exploratory" gestures and agitated gestures as we were generating new instruments with the RR method. Jan expressed her wish to work more with the two-hand setup, as it eased her practice a lot. More nuanced results were obtained when using sound triggers, as changes in sound seemed to be too rare for Ines: she mostly gesticulated. Rebecca suggested using stereo sounds could help her realize sounds come from both her hands.

Andrew has physical disabilities (for example, his head has to be attached to his wheelchair in order to stay up); he also presents an obsessive-compulsive disorder (OCD) that made him want to bite into his right thumb. When we fastened one Gametrak string to his right hand, his first action was to go and try to bite it. We thus recorded this gestural stream and generated a new instrument. The more he tried to bite again, the more he realized that his gesture was actually making sound: the effect was he stopped trying to bite into his thumb. Jan interacted with him by imitating the computer sounds Andrew made, either with her voice or with acoustic insturments; she also sang along with him. Several eye-contacts were triggered by such interaction: Andrew would move his hands while looking at Jan, as he tried to share with her what he was feeling. As for Ines, Andrew interacted better with continuous sound modules than with triggered sounds. Additionnally, using air gesture input devices such as the webcam was none of a use with him: without tangible, tactile artifacts such as the GameTrak string, he would basically go back to his thumb biting. A notable feature was made possible when we recorded Wiliam's voice in one of our sound modules, and made him play it back with his gesture. Since that, he had not made any vocal sound, but when he began playing the instrument, he immediately reacted by vocalizing in response to his recorded voice on the computer, allowing Jan to interact vocally with him.

Gareth has strong motor disabilities along with cognitive disabilities: notably, he has difficulties to stay awake for a long time, resulting in motivation problems that caregivers and Jan had difficulties to cope with. As he was falling asleep during the first session, we noticed he had more ease to do head and body gestures than hand gestures, which is why we tried the webcam controller and record his gestural stream. The result was remarkable for Jan and ourselves, as he spent 30 minutes interacting with a continuous sound module. Jan mostly let him express himself, looking at him, waiting for his actions, and making approval and surprised facial expressions. When he seemed to go back to his sleep state, we rapidly generated a new instrument with the RR method: this had him open his eyes again, move his head and arms, remaining surprised and amused as if, as Jan coined it, he was telling himself *"hmm, I've just done that!"* We also used sound triggers to allow Gareth to play back a short flute phrase played by Jan. Again, both interacted as in a conversation: Jan would play a few notes with her flute, and Gareth would respoond by moving his hand and triggering flute sounds. Jan noticed how Gareth was waiting for her to play; she

[2] All disabled children's names have been changed for anonymous purposes.

also noticed he seemed *"more aware than initially"*. For him, triggered sounds was very useful as it was a way to alternate between silence and sound, thus paving the way for building motivation skills.

Stan is an autistic children. Jan usually does not work the same way with him, as he does not react the same way than other disabled children. For example, he sometimes gets focused on one aspect of a given activity instead of the whole activity. When he entered the room, he seemed to feel very impressed by all the new people, making no eye-contact at all and sitting on a chair. We tried to use the GameTrak string to have him trigger sounds, but what happened was he did not seem to care too much about the sound: rather, he was playing with the string's plastic extremity. We also encountered difficulties to record a representative gestural stream from him, which resulted in a fatal loss of time for Stan. He ended up putting the string out of the GameTrak, then flipping it back several times, having no interest in the new continuous sounds we were trying to use. Jan evoked the possibility that such a different activity acted as a trauma, as it upset his dynamics of playing the keyboard: she expressed a will to trigger flexibility in him by having him learn how to adapt to such practice, little by little.

Lana was the eldest children we worked with. She has strong cognitive disabilities as well as some physical disabilities. A particularity with her is that she is used to stare at people without making a move: she thus interacts by making large gestures, vocalizations, and facial expressions. After recording her large gestures and generating an instrument with the RR method, she seemed very amused by triggering sounds with the GameTrak string: she would move her hand, stop when hearing a sound, and instantaneously laugh at such sonic event. Lana was the only children that developed new creative uses of the GameTrak — indeed, she seemed very intrigued by the device itself. After having located a precise point in her control space, she would stay in that position, and play the string with her thumb, which was actually making a creative use of the decision boundaries of the ML algorithm. One notable observation was when we recorded Jan playing tambourine, and Lana's laugh in one of our sound modules. Jan then took the initiative to fasten one of the GameTrak string to Lana's right arm, and the other string to Lana's left knee. We recorded a new gestural stream and generated a new instrument. The disabled children was very stimulated by such setup, smiling and laughing at triggered sounds as Jan would play with her, gently touching her arms or legs and smiling back at her. Jan reported she was very enthusiastic to see Lana interacting with her so intensely.

Discussion

We explored several input devices as well as different musical activities, while our grab-and-play approach allowed the rapid mapping between these. Our exploratory case study gave us positive feedback about new types of interaction between Jan, the music therapist, and four children with different disabilities (an exception being the autistic children). These children all have specific ranges of motion that, according to Jan, are likely to change from sessions to sessions. Our RR method allowed the building of adapted music instruments covering such specific motion ranges, while allowing for designing new types of gestures and sound relationships. Also, the rapidity of new prototypings allowed Jan not to interrupt her work with the children, which is crucial when working with children with attention issues. Apart from mapping strategies, we also explored pros and cons of different types of input devices and sound modules, giving insights for future softwares to be developed — notably the fast recording and replaying of real-world sounds.

However, as flexible as our grab-and-play approach was, a considerable amount of work remains to be done in order to fully understand and satisfy the needs of both the music therapist and children with different disabilities. Specifically, another part of the interaction has almost been unexplored, namely, interaction between Jan and our software. During the study, Jan understood well basic effective commands of the software — for example, she was frequently proposing to record a new gestural stream, or asking if we could generate a new mapping. In future work we will lead participatory design sessions with Jan and the caregivers in order to converge with a more accessible interface (see figure 6.1). This interface would constitute the base of longitudinal studies, in which we would observe Jan interacting autonomously with disabled children on a longer time scale. We hope to collect positive feedback on learning processes that our exploratory case study suggested (for example, Ines and Andrew acquiring embodied skills, Stan learning to adapt to new activities, or Gareth and Lana strenghtening their communication and motivation abilities). Finally, we saw the RR method supported well Jan's interaction with the children, as she seemed to value exploration and instantaneous reactivity over precise customization. Further work might investigate how other data-driven methods could better support her practices.

6.3 General Discussion

Through two studies, we explored novel data-driven approaches to music-making and instrument prototyping among disabled children. A first workshop suggested valuable advantages for our new grab-and-play interaction paradigm compared to the original Wekinator setup, allowing rapid prototyping of instruments covering a specific range of motion. We then study the use of this interaction paradigm with Jan, a music therapist. An exploratory case study gave us positive feedback on musical interaction allowed by our software, and informed the needs of Jan's specific practices. In the next two years we will develop our contribution to musical inclusion through workshops and interfaces that will allow autonomous interaction between Jan and disabled children, along with adapted input devices and sound modules. Analyzing disabled children gestural data could also motivate the use of our three other data-driven methods.

Overall, we underlined the need to develop new methodologies to address the needs of disabled children. Cognitive disabilities lead to several communication obstacles that usual methodologies take as granted. Also, further longitudinal studies would imply administrative and ethics issues that might make the design process even more complicated, and time consuming. We took the decision to include a music therapist as well as caregivers in the design process. We also wanted to explore several disabilities to shed light on different technological needs such children could have, with a wish to inspire other researchers. In future work we would also like to collect data among disabled children relatives, as well as leading parallel studies with other music therapists and caregivers, as we suppose each of them would experience the software differently. Finally, it is worth remembering that studying such a user group gives us invaluable feedback for our software, be it used by disabled children, therapists, or professionnal composers.

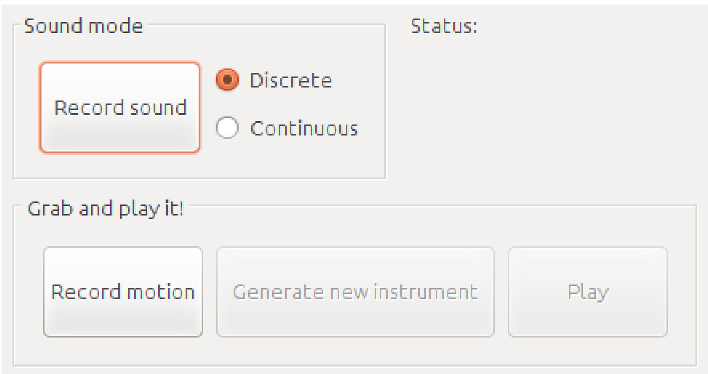

Figure 6.1: Possible future interface to study Jan autonomously interact with disabled children.

7 Conclusion

We applied data-driven interaction paradigms as creative machine learning approaches to the design of gestural musical interfaces. Following a user-centered methodology, four user studies led with two user groups informed the design of these paradigms. Composers valued an emphasis on embodied musical exploration and discovery, while the rapid adaptation afforded by our tool fitted and developed new practices for Jan, a music therapist working with disabled children.

Future work might investigate other ways to implement these data-driven methods. For example, developing high-level descriptors for gestural and audio data could allow efficient and rapid prototyping. Also, developing interactive visualizations could allow fine-tune modification of a built model. Finally, one could imagine a new mapping paradigm in which the training phase and the running phase would merge, enabling the system to adapt progressively to the user during its use: interactive *unsupervised* learning approaches could be used to discover structures in real-time through the user's motion data, allowing for automatic adaptation rather than design.

In parallel, further work should be done exploring interactive supervised learning with disabled people. Our exploratory results encourages us to lead longitudinal studies with Jan, developing an accessible interface and studying her autonomously interact with children on a longer scale. We believe data-driven approaches could allow disabled people to adaptively rehabilitate movement and cognitive abilities, for example by learning new gesture and sound relationships. On a longer scale, we also aim at making our software and its future accessible interface open-sourced, allowing more music therapists to use it and gathering more feedback for further ameliorations.

Lastly, we hope our research on data-driven approaches would inspire further research investigating creative, user-centered applications of machine learning algorithms. Our work focused on building new digital musical instruments; however, we tried to keep our approach as general as we could — indeed our four data-driven paradigms could be applied to data other than gestural or audio. Adapting our tool for designing other types of creative systems might thus be of interest. Yet, we strongly believe music, as a creative process blending improvised interaction between humans with embodied knowledge, remains an exemplary case study for such research topics.

We will present the first design/evaluation step of our research project in a paper[1] and talk at next *International Computer Music Conference* in September 2016. We are also preparing a paper on the second design/evaluation step for next *ACM CHI Conference on Human Factors in Computing Systems*, held in May 2017. As a follow-up to this project, I will spend the next three years investigating adaptive, expressive interactive systems for music performance and pedagogy in a CDN-funded PhD work at *IRCAM*, holding joint efforts with *Goldsmiths, University of London*.

[1] Scurto, H., Fiebrink, R. (2016). Grab-and-play mapping: Creative machine learning approaches for musical inclusion and exploration. International Computer Music Conference (ICMC'16), Utrecht, Netherlands.

BIBLIOGRAPHY

[Amershi et al., 2012] Amershi, S., Fogarty, J., and Weld, D. (2012). Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[Buxton, 2010] Buxton, B. (2010). *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann.

[Caramiaux and Tanaka, 2013] Caramiaux, B. and Tanaka, A. (2013). Machine learning of musical gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2013), Seoul, South Korea*.

[Domingos, 2012] Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87.

[Dow et al., 2010] Dow, S. P., Glassco, A., Kass, J., Schwarz, M., Schwartz, D. L., and Klemmer, S. R. (2010). Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(4):18.

[Fails and Olsen Jr, 2003] Fails, J. A. and Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM.

[Fiebrink et al., 2009] Fiebrink, R., Cook, P. R., and Trueman, D. (2009). Play-along mapping of musical controllers. In *Proc. International Computer Music Conference*.

[Fiebrink et al., 2011] Fiebrink, R., Cook, P. R., and Trueman, D. (2011). Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 147–156. ACM.

[Fiebrink et al., 2010] Fiebrink, R., Trueman, D., Britt, C., Nagai, M., Kaczmarek, K., Early, M., Daniel, M., Hege, A., and Cook, P. (2010). Toward understanding human-computer interaction in composing the instrument. Proc. of the International Computer Music Conference.

[Françoise et al., 2014] Françoise, J., Schnell, N., Borghesi, R., and Bevilacqua, F. (2014). Probabilistic models for designing motion and sound relationships. In *Proceedings of the 2014 International Conference on New Interfaces for Musical Expression*, pages 287–292.

[Hunt and Wanderley, 2002] Hunt, A. and Wanderley, M. M. (2002). Mapping performer parameters to synthesis engines. *Organised sound*, 7(2):97–108.

[Katan et al., 2015] Katan, S., Grierson, M., and Fiebrink, R. (2015). Using interactive machine learning to support interface development through workshops with disabled people. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 251–254. ACM.

[Kumar et al., 2011] Kumar, R., Talton, J. O., Ahmad, S., and Klemmer, S. R. (2011). Bricolage: example-based retargeting for web design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2197–2206. ACM.

[Laguna and Fiebrink, 2014] Laguna, C. P. and Fiebrink, R. (2014). Improving data-driven design and exploration of digital musical instruments. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 2575–2580. ACM.

[Lee et al., 1991] Lee, M., Freed, A., and Wessel, D. (1991). Real time neural network processing of gestural and acoustic signals. In *Proc. International Computer Music Conference*.

[Leman, 2008] Leman, M. (2008). *Embodied music cognition and mediation technology*. Mit Press.

[Nielsen, 1994] Nielsen, J. (1994). Heuristic evaluation. *Usability inspection methods*, 17(1):25–62.

[Resnick et al., 2005] Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., and Eisenberg, M. (2005). Design principles for tools to support creative thinking. In *Proceedings of the NSF Workshop on Creativity Support Tools*, pages 25–36.

[Schnell and Battier, 2002] Schnell, N. and Battier, M. (2002). Introducing composed instruments: Technical and musicological implications. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–5.

[Shilman et al., 2006] Shilman, M., Tan, D. S., and Simard, P. (2006). Cuetip: a mixed-initiative interface for correcting handwriting errors. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 323–332. ACM.

[Stumpf et al., 2007] Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. (2007). Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91. ACM.

[Talbot et al., 2009] Talbot, J., Lee, B., Kapoor, A., and Tan, D. S. (2009). Ensemblematrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292. ACM.

[Thomaz and Breazeal, 2008] Thomaz, A. L. and Breazeal, C. (2008). Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716–737.

[Trueman and DuBois, 2002] Trueman, D. and DuBois, L. (2002). Percolate. *URL: music.columbia.edu/PeRColate*.

[Wagstaff, 2012] Wagstaff, K. L. (2012). Machine learning that matters. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 529–536.

[Wright and Freed, 1997] Wright, M. and Freed, A. (1997). Open sound control: A new protocol for communicating with sound synthesizers. In *Proc. International Computer Music Conference*.