User-Centred Design Actions for Lightweight Evaluation of an Interactive Machine Learning Toolkit

Francisco Bernardo
Department of Computing
Goldsmiths, University of London
London SE14 6NW
---
f.bernardo@gold.ac.uk

Mick Grierson
Department of Computing
Goldsmiths, University of London
London SE14 6NW
---
m.grierson@gold.ac.uk

Rebecca Fiebrink
Department of Computing
Goldsmiths, University of London
London SE14 6NW
---
r.fiebrink@gold.ac.uk

ABSTRACT

Machine learning offers great potential to developers and end users in the creative industries. For example, it can support new sensor-based interactions, procedural content generation and end-user product customisation. However, designing machine learning toolkits for adoption by creative developers is still a nascent effort. This work focuses on the application of user-centred design with creative end-user developers for informing the design of an interactive machine learning toolkit. We introduce a framework for user-centred design actions that we developed within the context of an European Union innovation project, RAPID-MIX. We illustrate the application of the framework with two actions for lightweight formative evaluation of our toolkit—the JUCE Machine Learning Hackathon and the RAPID-MIX API workshop at eNTERFACE'17. We describe how we used these actions to uncover conceptual and technical limitations. We also discuss how these actions provided us with a better understanding of users, helped us to refine the scope of the design space, and informed improvements to the toolkit. We conclude with a reflection about the knowledge we obtained from applying user-centred design to creative technology, in the context of an innovation project in the creative industries.

KEYWORDS

User-centred Design; Action Research; Interactive Machine Learning; Application Programming Interfaces; Toolkits; Creative Technology.

ARTICLE INFO

Received: 03 April 2018
Accepted: 09 May 2018
Published: 11 July 2018

https://dx.doi.org/10.7559/citarj.v10i2.509

1 | INTRODUCTION

Recent developments in artificial intelligence are generating a surge of interest around making machine learning (ML) more accessible to new groups of people—particularly people who are not ML experts—for problem-solving and practical applications in various domains. However, using ML is still often difficult for those who are not ML experts or data scientists. Nonetheless, as with other computer technologies that have seen mass adoption (e.g., email interfaces, word processing software, web technologies, etc.), broader adoption of ML can be facilitated by improving the usability of tools for employing ML.

Many software developers within the creative industries, as well as many non-professional

developers working with creative technology, can benefit from ML techniques. For instance, ML can facilitate the analysis of audio, visual, and sensor data. ML can support the creation of new systems for embodied interaction and expression, as well as provide new creative workflows for rapid prototyping and product customisation (Hartmann et al., 2007; Fiebrink, Cook, & Trueman, 2011; Katan, Grierson, & Fiebrink, 2015).

This paper describes the development of new ML tools within the context of a European Commission-funded "Innovation Action"—a joint effort between academic institutions and companies aimed at technology transfer and the production of new or improved products or services. This project, called RAPID-MIX ("Realtime Adaptive Prototyping for Industrial Design of Multimodal Interactive eXpressive technology"), has involved a consortium of three European research labs and five small and medium-sized enterprises (SMEs) collaborating in the development of creative technology tools for rapid prototyping and product development. Central to RAPID-MIX's goals is the creation of new machine learning tools targeting developers in the creative industries.

A user-centred approach to design has been critical to ensure that these new tools are usable and useful for creative developers. The needs of creative developers using ML are not well-understood. ML presents unique challenges to developers (Patel et al., 2010), and the needs of people employing technology used in design and other creative practices are often distinct from people engaging in activities with more well-defined outcomes (Cherry & Latulipe, 2014). Further, the design of software application programming interfaces (APIs) presents distinct challenges from other design processes in which user-centred methodologies are often used (e.g., the design of end-user-facing graphical interfaces) (Myers & Stylos, 2016). In RAPID-MIX, we have thus needed to carefully craft user-centred methodologies appropriate to the design of our new ML tools, within the additional constraints of a complex project with multiple academic and industry partners.

The paper is structured as follows. In the following section, we review prior work in user-centred design and interactive machine learning. We then present our main adopted research methodology and the framework we devised to employ user-centred design techniques systematically. We describe two user interventions that involved different user groups, methods and data collection strategies. Finally, we discuss the emerging knowledge about the application of user-centred design in the context of innovation projects and creative technology.

2 | BACKGROUND WORK

2.1 USER-CENTRED DESIGN

User-centred design (UCD), a term coined by Norman and Draper (1986), is a design approach based on understanding users, their tasks and environments. UCD has been characterised as "philosophy and methods, which focus on designing for and involving users in the design of computerised systems" (Abras, Maloney-Krichmar, & Preece, 2004).

When adopting UCD, we look for ways to better understand users, their characteristics, skills and behaviour in specific tasks. Users are at the centre of this process; they are involved at an early stage and throughout the design process. This approach enables the design team to communicate and negotiate a better and shared understanding of the right problem, and to reason about and inform design decisions for the right solution (Monk, 2007). However, Norman notes that in UCD, "even though the ideal can seldom be met in practice, it is always good to aim for the ideal, but to be realistic about the time and budgetary challenges" (2013, p. 239).

According to Ritter, Baxter and Churchill (2014), UCD has a broader focus, greater emphasis on the user, and lesser use of formal methods for requirements gathering and specification than other approaches, such as human factors and ergonomics, or socio-technical systems design. They claim that adopting a user-centred approach can help to address essential design problems and lead to systems that are more useful, usable and satisfying. It can help designers overcome errors or issues that arise from relying on their personal assumptions, experience or intuition. Ritter, Baxter and Churchill argue that, on the one hand, UCD leads to financial savings, as iteratively refining designs can lead to fewer problems in the final product. On the other hand, though, UCD entails its own costs, and UCD does not guarantee the success of a product: while "the lack of usability can

be a sufficient reason for failure", "usability is neither a necessary nor sufficient condition for success" (p. 14). The value of UCD may be assessed by estimating its return on investment (ROI)—the extent to which the time and effort spent doing user research provide worthwhile benefits; ROI is highly valued but also often difficult to measure and to manage (Ritter et al., 2014). Holtzblatt, Wendell and Wood (2004) also refer to the ROI as an important criterion for the adoption of UCD practices in organisational contexts.

To avoid complicated, time-consuming and expensive techniques for ensuring usability of a new product, Nielsen (1994) proposed applying "discount" methods—such as user and task observation, scenarios, simplified thinking aloud and heuristic evaluation—to design problems with a well-defined user population and set of tasks. Monk (2007) similarly proposed lightweight techniques that can be easily picked up and applied effectively (i.e., learned in one day, only taking person-days to apply).

2.2 INTERACTIVE MACHINE LEARNING

Many different approaches have emerged for enabling application of ML by non-experts to various problem domains. One approach involves packaging learning algorithms, evaluation strategies, etc., into high-level, GUI-based tools that can be used without programming; this is the approach used by tools such as Weka (Hall et al., 2009). Other approaches provide additional mechanisms for users to incorporate information about their goals or domain knowledge into their work with ML algorithms. Some of these may employ an interactive machine learning (IML) approach, in which users engage in iterative training, evaluation, and corrective actions (such as modifying the data on which an ML algorithm is trained) (Fails & Olsen, 2003).

IML can be useful for ML problems in which the user's goal is to encode a human-understandable behaviour into the system. In the creative industries, this can include the design of many types of systems that respond to human activity (e.g., new musical instruments or games controlled by human movement) or that map between different domains of multimedia data (e.g., visualisations that respond to real-time characteristics of music). In such applications, users can provide training examples that communicate their intention for the trained model (e.g., showing that certain actions performed with a sensor should result in certain sounds or game commands). Users can iteratively steer the behaviour of the trained model by modifying these training examples.

Previous research on IML has mostly focused on designing graphical user interfaces for end-user interaction with learning systems. This includes interfaces that enable domain expert users (who may have little or no machine learning expertise) to steer models by providing new training examples (Fails & Olsen, 2003; Fiebrink et al., 2011), adjusting misclassification costs (Kapoor et al., 2010), adjusting weightings of component classifiers in an ensemble system (Patel et al., 2010), etc.

Prior work by Bernardo et al. (2017) has proposed that the experience and challenges faced by developers using learning systems should also be considered. Developers are "users" of machine learning when they configure learning algorithms, and when they train, evaluate, and export models. Some research has focused on methods to make these activities more efficient and effective (Amershi et al., 2015; Patel et al., 2010). Developers are also users of ML when they create intelligent systems ultimately intended for use by others (i.e., the "end users"). Developers building intelligent systems use ML through infrastructural software (i.e., middleware) such as software libraries, APIs, online services, etc. These tools inevitably influence developers' working processes and experiences. Some existing research considers usability and user experience factors of programming languages, IDEs, middleware, API documentation and code examples (e.g., Clarke, 2011; Edwards et al., 2003; Myers & Stylos, 2016). However, the additional challenges experienced by developers working with ML middleware are underexplored. New research should focus on these challenges and contribute with additional understanding about the needs and the experience of developers working with ML middleware.

3 | METHODOLOGY

In 2015, members of the RAPID-MIX consortium developed a UCD framework (Bernardo et al., 2015; Bevilacqua et al., 2015) for internal guidance of the consortium (RAPID-MIX researchers and industrial stakeholders), for other actors within the creative industries (individuals, start-ups, academia, etc.) and

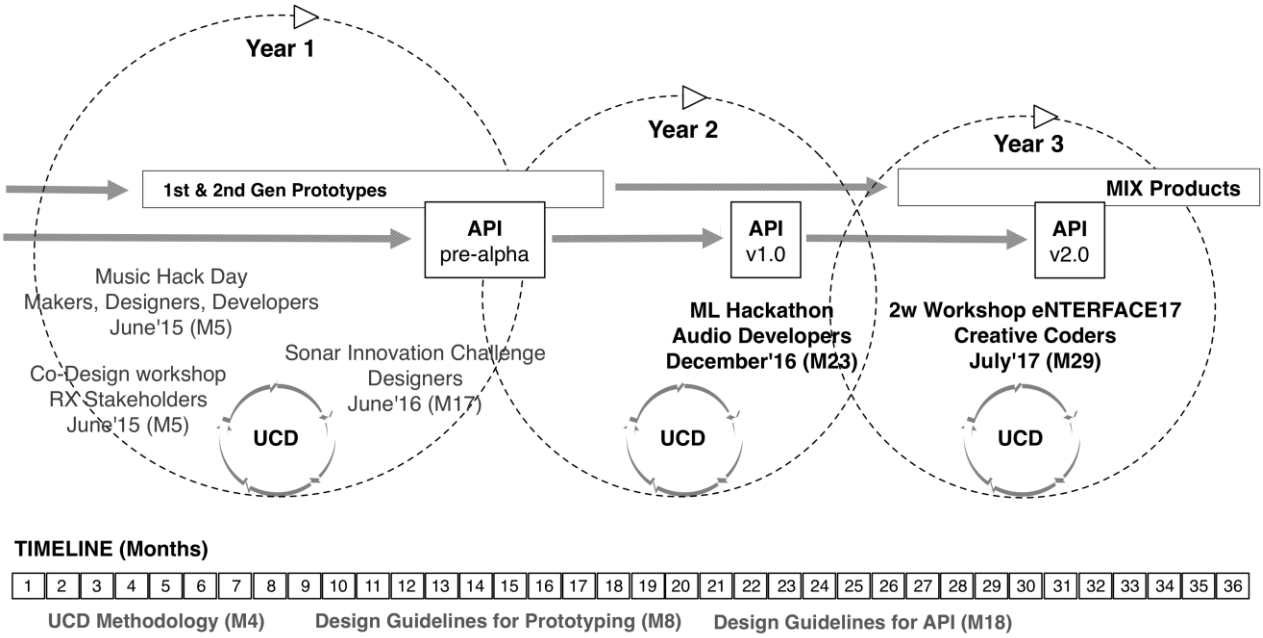

Figure 1 | RAPID-MIX UCD framework with selected UCD actions.

the general public. This framework consists of a methodology of UCD actions—research interventions in which a UCD technique is deployed to answer questions about the design of a new technology. These questions are directed by the goals of understanding the characteristics of potential users and of users' experiences with the RAPID-MIX technologies as they developed.

Early UCD actions within RAPID-MIX included co-design workshops with project stakeholders, hackathons, and public workshops with different user groups, including professional audio developers, creative developers and students (see Figure 1). In the next sections, we describe the two selected UCD actions that are the focus of the remainder of this article. These actions (which took place in months 23 and 29 of the 36-month RAPID-MIX project) investigated the usability and appropriation of two different subsets of the RAPID-MIX API by different types of users. With these activities, we wanted to investigate the following questions:

- What are the needs, goals and values of this user group?
- How do these users use the RAPID-MIX API and what for? What about machine learning or the RAPID-MIX API was confusing for these users? What errors did they make? What was unexpected?
- What other API features might they need, and which would they need the most?

3.1 JUCE MACHINE LEARNING HACKATHON WITH AUDIO DEVELOPERS

The JUCE Machine Learning Hackathon (Figure 2a) was a one-day hackathon in December 2016, organised with ROLI [1], an SME in music technology that participates in the RAPID-MIX consortium. Part of ROLI's product portfolio, JUCE [2] is a popular cross-platform C++ framework with a focus on audio applications, which is widely used in the industry. JUCE's customers and users are audio software engineers, developing audio and music apps for different platforms (standalone applications and plug-ins for desktop, and mobile apps for iOS and Android). The hackathon focused on the JUCE RAPID-MIX Module, a thin wrapper around the RAPID-MIX API that exposed functions for training and evaluating classification and regression models, utility functions for model serialization/de-serialization from JSON (JavaScript Object Notation) files, and data structures with JUCE primitive data types.

3.1.1 METHOD

The JUCE Machine Learning Hackathon was advertised on an online booking site, on the JUCE forum and mailing lists, and on mailing lists of educational institutions. As motivation, a ROLI Lightpad BLOCK [3] was given as an award for each of the winning team's attendees. The hackathon began with an induction introducing participants to supervised machine learning techniques and to the JUCE RAPID-MIX Module. Consent forms were

Figure 2 | a) The JUCE Machine Learning Hackathon and b) the winning 'hack' "Harmeggiator".

distributed along with a short pre-hack questionnaire about participants' skills in software development, programming languages and environments, and machine learning. Participants then had six hours to complete a "hack"—a small project of their choosing that used the JUCE RAPID-MIX Module—after which every team presented their hack to a jury panel of JUCE and RAPID-MIX representatives. Participants worked in groups of no more than three. Teams posted their hack code to GitHub. Hackathon facilitators recorded questions, critiques and feedback that were voiced by the participants. After the awarding ceremony, we conducted structured interviews with participants. We interviewed participants about changes in their design goals throughout the hack, module features that they used, limitations they discovered and strategies they used to overcome them, and suggestions for real-world applications of the module. The interviews were video recorded and subsequently analysed.

3.1.2 RESULTS

Around 20 developers attended the event. The pre-hack questionnaire indicated that most of the participants were proficient in C++, had extensive programming knowledge, and had used JUCE before. Most participants stated they had very limited or no knowledge of machine learning techniques. Most teams had a clear idea of what they wanted to build for their hack after the ML presentation. Some participants indicated that they did not intend to submit a hack; rather, they participated to learn about machine learning. Five teams submitted a hack. We briefly describe each hack as follows:

- "Embedded ML" - This hack ran the JUCE RAPID-MIX module on Beagle Bone Black with Bela (McPherson, 2017), a highly constrained embedded system tailored for ultra-low-latency audio. This system used ML for gesture recognition with a ROLI Lightpad BLOCK. The author interestingly stripped the JUCE wrapper code away and used the RAPID-MIX API directly in a console application.

- "Filter Classification" - used ML as a quick prototyping alternative for digital audio filter design. This system classified filter types (i.e., high-pass, low-pass, band-pass) from a set of coefficients of a Finite Impulse Response (FIR) filter—i.e., feedforward filter type with a finite duration impulse response (Steiglitz, 1996).

- "Harmeggiator" - implemented a MIDI effect VST plugin (Virtual Studio Technology by Steinberg [11]) to arpeggiate chords from gestures performed with the ROLI Lightpad BLOCK. The system provided functionalities for applying the IML workflow for training and mapping arpeggiation parameters (speed, arpeggiation direction, shape) extracted from gestures to a set of intervals extracted from chord note values (Figure 2b).

- "Feature Extractor + RapidMix" - extended this participant's existing audio feature extraction software with IML capabilities. IML was applied to the audio analysis features and used to drive generic parameters of an audiovisual application.

- "FM Synth Patch Generator" - calculated FM synthesis (Chowning, 1973) parameters to match synthesiser output with sampled instruments through similarity analysis of audio—i.e., to make an FM synthesizer to resemble the sound of a recorded

Figure 3 | a) and b) Participants working on their projects and c) presenting a final project.

piano. Technical and conceptual issues prevented the timely delivery of a fully functional hack.

In general, observation and feedback from interviews about the use of JUCE RAPID-MIX module confirmed it was an appropriate tool to achieve the users' proposed hacks and prototypes. There was highly positive feedback about module code quality and clarity, and the fast implementation results it enabled. Documentation and examples were found easy to navigate and use. The diversity in the type of applications submitted showed that the JUCE RAPID-MIX module is useful and usable for a broad range of applications and for a variety of hardware platforms. One participant mentioned that introductory talks delimited the state and capabilities of the library very well and that this influenced the scope of what the participant wanted to do.

There were usability issues identified along with other technical issues, including:

- Participants were confused by one of the higher-level abstractions built into the API that targeted a use case that was not relevant to these hacks (specifically, an abstraction aggregating many ML models into one data structure for use in multiparametric synthesiser mapping).

- Some participants noted that the lack of C++ templated data structures could exclude applications where significant numerical precision was required.

- Some participants noted the lack of asynchronous API calls for progress notification in the ML model training.

- The use of much larger datasets than anticipated unveiled some bugs in the RAPID-MIX API implementation.

Participants expressed a desire for additional and more effective documentation about general machine learning concepts, specific API methods, and more domain-specific code examples (i.e., IML applied to audio). Participants also suggested additional features for the JUCE RAPID-MIX module such as:

- incorporating more types of ML algorithms, particularly for temporal modelling, such as dynamic time warping (DTW).

- providing more granular control over ML algorithms and evaluation methods by exposing more parameters for expert use (e.g., changing the architecture or activation function of neural networks).

- providing more ways to examine the ML models, for example through data visualisation, to aid understanding of model decision boundaries or model behaviour in higher dimensional spaces.

- improving the training speed of ML models.

validation of input data for both training and model evaluation.

3.2 TWO-WEEK SUMMER WORKSHOP WITH CREATIVE DEVELOPERS AT ENTERFACE'17

We ran a two-week workshop at eNTERFACE, a yearly summer workshop organized by the SIMILAR European Network of Excellence. eNTERFACE 2017 was held between 3–15 of July, at Universidade Católica Portuguesa in Porto. This workshop followed the official beta release of the RAPID-MIX API (May 2017).

The beta release included several new features; additional learning algorithms, such as DTW, Gaussian Mixture Models (GMM), Hierarchical Hidden Markov Models (HHMM)—via improved integration with the XMM package (Françoise, Schnell, & Bevilacqua, 2013)—and particle filtering—via integration with Gesture Variation

Follower (Caramiaux, Montecchio, Tanaka, & Bevilacqua, 2014); a new class library with signal processing primitives (e.g., circular buffer, Root Mean Square, Mel Frequency Cepstral Coefficients, first- and second-order derivatives, etc.), and an improved web API for cloud-based multimodal data storage and retrieval.

This UCD action targeted creative coders who had a more diverse set of interests (i.e., not just audio programming), and who used a wider variety of programming languages and tools. It focused on helping participants to gain practical experience with elements of the toolkit, and on simultaneously identifying usability issues, learning obstacles and intended uses.

3.2.1 METHOD

Thirteen participants (2 female, 11 male) with prior background in creative coding and multimedia were recruited in two rounds through research mailing lists, creative communities on Facebook, and personal contact networks. Most participants had master's level degrees, three participants were PhD students, and two participants were professionals in web development and game development, respectively.

In the weeks before the workshop, we surveyed participants about their background and motivations for attending. We then refined the UCD action plan to reflect these. For instance, as participants all had prior experience in JavaScript (JS), we narrowed the scope of evaluation to the JS subset of the RAPID-MIX API. The RAPID-MIX API JS library had been previously integrated into CodeCircle (Zbyszyński, Grierson, Yee-king, & Fedden, 2017), an online live coding environment for beginning coders and computing students. CodeCircle aims to support efficient experimentation and prototyping activities (Parkinson, Zbyszyński & Bernardo, 2017; Zbyszyński, Grierson, & Yee-king, 2017).

Each day of the first workshop week began with a researcher-led induction session. These sessions progressively introduced participants to ML concepts and use cases, and to the relevant components of the RAPID-MIX API. After the induction session, participants spent 2–3 hours engaged in hands-on exploration with the tools (Figures 3a and 3b). Each day concluded with a video-recorded group discussion.

In the second week, the workshop format changed to mentored project work. Participants worked independently on their creative projects, and at the end of each day a group discussion took place in which they reported on their progress and challenges.

Participants also completed questionnaires after the workshop in which they provided information about their experience with the different elements of the RAPID-MIX API and how they benefited from them.

We provided several resources for assistance, reference and learning during the workshop. The RAPID-MIX API code repository and website provided documentation. A Slack channel supported Q&A with remote mentors and participants. We also provided CodeCircle documents exemplifying how to use different functionalities of the JS RAPID-MIX API (e.g., classification, regression, temporal classification, etc.) with different sensors (e.g., mouse, webcam, LeapMotion [4], MYO [5], Gametrak, etc.), and audiovisual outputs (e.g., WebGL [6], Web Audio API [7], P5.js [8], Three.js [9], etc.).

The workshop was structured to allow flexible participation schedules. From the 13 participants who attended most of the first week, only the 5 participants (3 males, 2 females) who had previously enrolled for the full 2-week workshop attended the second week. The participants who opted not to carry on lacked availability due to academic or professional commitments and stated they had fulfilled their initial goal of getting a cursory understanding of ML and the RAPID-MIX API.

3.2.2 RESULTS

At the end of the second week, four participants submitted their projects (as source code in GitLab [10] or in CodeCircle documents) and delivered a final demonstration to the remaining group. They presented the following projects:

- Participant A explored the use of ML to exert more expressive control over the feedback loop and slide transitions of a Kodak carousel projector. She mixed and mashed-up CodeCircle examples until she focused her exploration on using ML regression with the Myo sensor's EMG and motion signals, to control different visual outputs such as colour gradients and animations. She also employed temporal classification with DTW, using an Arduino

microcontroller board to control the projector (https://vimeo.com/225762966).

- Participant B submitted two projects: 1) a web application that implemented the rock-paper-scissors game—single player against computer—for which he used RAPID-MIX API JS and Leap Motion for classification of hand poses, providing both pre-trained pose models and optional customisation features; and 2) a "Gesture server" application that used a server-side component to do gesture recognition with the accelerometer data of a wirelessly connected smartphone (Figure 3b).

- Participant C used RAPID-MIX API regression with Leap Motion hand pose data to train and control 3D mesh deformation, iterating from using a simple regression model to control single-parameter mesh transformation, to employing a more sophisticated solution that used multiple models to control individual vertices (Figure 3c).

- Participant D decided to build his own toolkit with building blocks for visualization of the Myo sensor data and ML processing, wrapping the RAPID-MIX API in a single-page web application. His project evolved from building client-side to server-side ML training and processing. The project involved recent web technologies but unfortunately was not finished before the conclusion of the workshop.

The data collected from the workshop included observation notes, Slack chat logs, pre- and post-workshop questionnaires, video recordings from group discussions and final presentations, and source code of the participants' projects.

In the discussion groups, we asked participants to identify compelling uses of the RAPID-MIX API for creating future technology. Identified uses include: making products for others (e.g., games, physical activity recognisers, interactive music performances, smartphone sensor apps); creating personalised experiences for oneself; enabling social and group interaction; emotion classification; providing corrective user feedback; enabling more natural and expressive interaction and controllers; using it as a teaching material for kids in hands-on workshops; using it for efficient workflows and fast results when working with sensors; and combining different ML algorithms for simultaneous use in real-time applications.

User feedback about API usability was distilled from the group interviews and from the post-workshop questionnaire into the following items:

- Participants were generally enthusiastic about the speed of prototyping with the RAPID-MIX API, particularly using CodeCircle and the RAPID-MIX JS library. They found CodeCircle examples useful for prototyping interface designs with different sensors, and for providing building blocks for quick integration into their own creative projects.

- Participants appreciated the code clarity, simplicity and terseness of the provided examples. However, they perceived the significant boilerplate code and poor-quality code comments negatively, mentioning that these impeded their understanding of the API. Other participants requested that we added better code comments to explain the role of constant values that were not clearly contextualised or explained.

- Participants complained about the examples' focus on audio, and the lack of examples applying ML to visual media. In general, the group was not knowledgeable about digital audio and found audio examples too abstract. Code examples that provided richer audiovisual feedback and control were most highly regarded.

- Participants suggested the provision of complementary high-level documentation that could give them a quick and broad explanation about ML concepts and expected API workflows. They also requested improvements to the structure and visual presentation of documentation, claiming it was not uniform across the whole set and that it was confusing to navigate.

- Some participants revealed difficulties in understanding how to use data with IML. One participant found it difficult to understand the conceptual difference between using data for training and for running ML models; she overcame this difficulty by creating different variables to store the datasets for each functionality and testing the outcomes step-by-step, offline. Another participant was not getting the expected classification results, because he had made a conceptual error of implementing training with raw data, and testing with RMS-smoothed (Root Mean Square) data.

- As in the JUCE Hackathon, some participants noted the lack of asynchronous API calls for progress and termination notification in the ML model training.

- Participants found the thread-hogging behaviour in the ML model training function problematic for browser applications. Two participants opted for a server-side ML design implementation because this limitation made their application unresponsive.

- Several participants expected a community forum, which did not exist at the time of the workshop. They suggested it would have been useful to collect their interactions in the workshop and to support future adopters and users of the RAPID-MIX API.

Based on this feedback, we synthesised a set of recommendations to inform further development of the RAPID-MIX API and its documentation. So that these recommendations could directly influence subsequent development, we created Gitlab Issues [10] within the RAPID-MIX repository for each recommendation.

4 | DISCUSSION

In this section, we discuss how these two UCD actions enabled a better understanding of users and the scope of the design space. We also discuss some of the challenges of applying UCD in RAPID-MIX.

4.1 UNDERSTANDING THE USER

Our UCD actions contributed to a better understanding of the needs, goals and values of the target users of the RAPID-MIX API. The JUCE Machine Learning Hackathon focused on understanding audio developers, users of the JUCE RAPID-MIX module that wraps the RAPID-MIX C++ API. The eNTERFACE17 workshop was useful for understanding a more diverse user group—creative coders with different skill sets—and how they used the JS library. These two actions motivated the definition of two design personas (Cooper, Reimann and Cronin, 2007; Clarke, 2007) that are part of a more comprehensive set that characterises the users of RAPID-MIX:

- Jack, 35, experienced audio software developer, has a computer science degree. Programming means using C++. He owns/works for a start-up that produces VST plugins and mobile music apps. He uses JUCE as his main development tool. He is interested in machine learning but doesn't really know what it is about, beyond data mining in large databases and music information retrieval. As a pragmatic/systematic developer, he has built deep technical understanding and prides himself on developing DSP code with maximum performance, predictability and minimal memory usage.

- Sue, 23, creative computing and media student. She learned basic coding skills in C++, Python and JavaScript. She also made a couple of games in Unity and has experimented with physical computing and biofeedback sensors. She is interested in creating and expressing and is driven by concepts more than by technology. As an opportunistic developer, she writes code in an exploratory fashion and develops the sufficient technical understanding to solve her design problem.

According to Cooper, Reimann and Cronin (2007), design personas can support more natural and effective reasoning about design. In creating personas, we wanted to ground the design process in the most precise user population. Besides the characteristics uniquely conveyed by these personas, we found additional characteristics that are transversal to both groups. Most importantly, both groups share having little or no experience in ML. Further, these groups share other characteristics—e.g., high degree of intrinsic motivation, customisation and development skills, anticipating market needs, building for their personal needs, hobbyism, bricoleurism, etc.—that have been captured by previous research on Lead Users (von Hippel, 1986) and End-User Developers (Lieberman et al., 2006; Blackwell, 2017).

This knowledge about the user has provided a frame to the design process and has had practical implications. For instance, after considering the needs of the audio developer, we exposed lower-level primitives for configuration of the neural networks in the RAPID-MIX API C++ (e.g., hidden layers, activation function, etc); we built templated data types into the library for allowing a finer control over the numerical precision required in many audio applications and embedded hardware. We also added additional ML algorithms such as DTW to the library based on the overall interest in temporal data. To address the needs of the creative coder, we

created new examples with more visual feedback and better code styling. Online documentation was restructured to integrate a set of interactive tutorials with increasing complexity and contrasting features onto the website; these changes aimed for providing a smoother learning curve and an accelerated learning experience to the general audience.

4.2 SCOPING THE DESIGN SPACE

The set of artefacts that the RAPID-MIX API enables contributes to map and delimit its design space, by unveiling concepts, features, and technical integrations, i.e., plausible designs that match users' and stakeholders' goals with the affordances of the RAPID-MIX API.

A variety of project activities helped us to scope the design space. These include earlier UCD actions such as co-design sessions, prototypes created by members of the consortium, and products ultimately created by SMEs. The projects created at JUCE Machine Learning Hackathon and eNTERFACE17 contributed to improve the scope and understanding of the design space at these points in the project. The hacks and prototypes produced are real-world artefacts that can help characterise classes of designs enabled by our toolkit. Participants applied the IML workflow to quickly prototype, customise or personalise expressive control and real-time mappings between multimodal sensor data, application logic and multimedia outputs. We were able to confirm the applicability of RAPID-MIX technologies for rapid prototyping of sensor-based interactive applications and expressive multimodal controllers.

We were also able to deepen our understanding of certain aspects of the design space, such as design factors around the end-user interaction goals and contexts of use; or, how these factors and technical constraints manifested in end-user design decisions. For instance, we observed that certain participants' interaction goals led them to build technologies whose functionalities or interaction models differed significantly from our induction examples and demonstrators: some designs limited the exposure of the IML workflow to the end-user (e.g., in rock-paper-scissors, making the IML workflow optional and relegated to the settings panel); other designs were not interactive (e.g., FIR classifier used a unit test fixture). Some designs used unexpected volumes of data or number of inputs (e.g., FM synth path generator used FFT (Fast Fourier Transform) bins as inputs, gestural 3D mesh modelling used up to 76 models). Some participants changed their goals for ML over time (e.g., moving from 1-hand rock-paper-scissors to ambidextrous support). Some designs introduced new devices as data sources (e.g., ROLI Lightpad), or hosted ML models in new and particularly constrained systems (e.g., Bela on Beaglebone Black, which required float types). Other designs hit critical usability issues related to the inherent performance and memory limitations of the browser environment (i.e., single-thread client-side JS runtime) and the current architecture of the library when working with a high volume of data or number of models (e.g., gestural data controlling one model per vertex of 3D geometry). In terms of end-user constraints, most designs assumed little or no machine learning knowledge, and wrapped up data collection and model training in domain-specific abstractions or UI metaphors which facilitated the IML workflow.

4.3 MANAGING THE UCD PROCESS AND ITS ROI

UCD has most often been applied to the design of physical artefacts and graphical user interfaces. However, the application UCD with technologies such as middleware or ML is not as straightforward. In such cases, it might be challenging to verify success or recognise the benefits of such an approach.

The practical implementation of UCD within the large RAPID-MIX scale had a complex and challenging nature. Our experiences accord with Norman's (2013) remarks about the compromise between the UCD philosophy ideal and its practical implementation problems (e.g., conflicting requirements between different teams, process management difficulties, the explosion of data, limited and overworked personnel, etc.). For instance, some of the UCD actions had additional goals related to dissemination, promotion or pedagogy. Such additional layers can contribute to the complexity of UCD actions by blurring the roles within the team deploying the actions, challenging its coordination and effectiveness, and undermining the actions' end goals.

We observed that the methodology had different degrees of acceptance with different RAPID-MIX stakeholders. Despite the efforts to make sure we selected the right techniques, applied them correctly, deployed the actions effectively, and delivered useful documentation, their overall usefulness was

occasionally questioned. We received contradictory remarks about the relative contribution compared to the amount of time invested in UCD actions; or, about whether this was an adequate methodology to apply to the design of an API. In some situations, there was a lack of interest in engaging with particular techniques recommended in the UCD literature, for instance, collaboratively analysing user data, crafting user personas or doing API walkthroughs. Many of these occurrences could be identified as instances of what Holtzblatt, Wendell and Wood (2004) identified as the organisational backlash; or, as consequences of fragmentation in the overall design process, as suggested by Norman (2013).

However, the application of UCD for lightweight formative evaluation yielded undeniably useful results. As we have shown before, both actions led to useful insights which had an impact on the development of the RAPID-MIX API, with real and incremental enhancements. On the one hand, it validated some of the design assumptions, such as the abstraction level, general usability, usefulness for rapid prototyping, and how it caters well to the opportunistic approach of creative developers who lack ML expertise. On the other hand, it identified usability and technical issues, shortcomings of the documentation and learning materials, and lack of support for working more effectively with data. It also challenged some of the design assumptions—mostly about the distinct support that should be provided for different user groups with different levels of expertise in ML and software development—in terms of the learning content, learning curve, and ceiling of the middleware.

UCD actions, despite being lightweight, still require a great deal of thought and preparation to be engaging and simultaneously useful for participants and organisers. The data organisation and analysis can require significant time and effort, which is difficult to estimate before the action. Furthermore, in the overall UCD cycle, the outcomes of one stage must be clear and of consequence to the next stage (i.e., insights obtained from one UCD action should inform a subsequent design or development stage, and the design stage outcome should be used for inquiry in the next UCD action). This requires sequencing and integration with the overall development cycle, which may or may not be straightforward. Here, knowledge of UCD methodology alone was insufficient; management skills and practical working experience with UCD were invaluable, as was intuition to strike the right balance to between breadth and depth of assessment.

Overall, we found that UCD actions for lightweight evaluation provided gains that compared favourably to their operational costs. Using a lightweight approach, we managed to iteratively assess a broad set of dependent artefacts iteratively in the wild. This approach seems particularly well-suited for a technology with the level of indirection, and dependency on documentation and examples, such as the RAPID-MIX API or other software toolkits.

5 | CONCLUSION

We used the JUCE Machine Learning Hackathon and eNTERFACE'17 workshop for lightweight evaluation of our IML toolkit. These UCD actions were particularly useful for the assessment of a more comprehensive set of elements (C++ and JS APIs, documentation and code examples) with two different user groups. We uncovered conceptual and technical limitations and informed subsequent improvements to the toolkit and its documentation. Both UCD actions allowed us to gain a deeper understanding of the user groups—i.e., professional audio software developers and creative coders. We observed how participants in both actions were able to quickly grasp and successfully use different subsets of the RAPID-MIX API. Participants produced fully working hacks and prototypes that contributed to understand further and refine the scope of the design space of the RAPID-MIX toolkit. Both actions illustrate the effectiveness of our UCD approach by contributing with useful iterations to the overall design process of our toolkit. The practical experience of applying multiple UCD iterations in a multi-stakeholder innovation project comes with a significant caveat: even lightweight UCD actions require careful management of the organisational backlash and significant effort to achieve recognisable effectiveness and ROI.

ACKOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 Innovation Act H2020-ICT-2014-1 Project ID: 644862.

ENDNOTES

[1] https://roli.com

[2] https://juce.com

[3] https://roli.com/products/blocks/lightpad-m

[4] https://www.leapmotion.com/

[5] https://www.myo.com/

[6] https://www.khronos.org/webgl/

[7] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

[8] https://p5js.org

[9] https://threejs.org

[10] http://gitlab.doc.gold.ac.uk/rapid-mix/RAPID-MIX_API/issues?scope=all&state=all&label_name=UCD

[11] https://www.steinberg.net/en/products/vst.html

REFERENCES

Abras, C., Maloney-Krichmar, D., & Preece, J. (2004). User-Centered Design. In W. Bainbridge (Ed.), *Encyclopedia of Human-Computer Interaction* (Vol. 37, pp. 445–56). Thousand Oaks: Sage Publications. http://doi.org/10.1.1.94.381

Amershi, S., Chickering, M., Drucker, S. M., Lee, B., Simard, P., & Suh, J. (2015). ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 337–346. http://doi.org/10.1145/2702123.2702509

Bernardo, F., Tanaka, A., Fiebrink, R., Parkinson, A., Meala, S., & Bevilacqua, F. (2015). *D2.1 User-Centred Design Methodology*. Retrieved from https://rapidmix.goldsmithsdigital.com/wp-content/uploads/2016/02/D2.1-User-Centred-Design-Methodology1.pdf

Bernardo, F., Zbyszyński, M., Fiebrink, R., & Grierson, M. (2017). Interactive Machine Learning for End-User Innovation. In *Proceedings of the Association for Advancement of Artificial Intelligence Symposium Series: Designing the User Experience of Machine Learning Systems* (pp. 369–375).

Bevilacqua, F., Bernardo, F., Mealla, S., & Fiebrink, R. (2015). *D2.2 Design Guidelines for Prototyping*. Retrieved from http://rapidmix.goldsmithsdigital.com/wp-content/uploads/2016/02/D2.2DesignGuidelineforPrototyping.pdf

Blackwell, A. F. (2017). End-User Developers – What Are They Like? In F. Paternò & V. Wulf (Eds.), *New Perspectives in End-User Development* (pp. 121–135). Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-60291-2_6

Caramiaux, B., Montecchio, N., Tanaka, A., & Bevilacqua, F. (2014). Adaptive Gesture Recognition with Variation Estimation for Interactive Systems. *ACM Transactions on Interactive Intelligent Systems*, *4*(4), 1–34. http://doi.org/10.1145/2643204

Cherry, E., & Latulipe, C. (2014). Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Transactions on Computer-Human Interaction*, *21*(4), 1–25. http://doi.org/10.1145/2617588

Chowning, J. M. (1973). The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering Society*, *21*, 1–10. http://doi.org/10.1038/srep01061

Clarke, S. (2011). How Usable Are Your APIs? In A. Oram & G. Wilson (Eds.), *Making Software: What Really Works, and Why We Believe It* (1 edition, pp. 545–565). O'Reilly Media.

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design*. Indianapolis, Indiana: Wiley.

Edwards, W. K., Bellotti, V., Dey, A. K., & Newman, M. W. (2003). Stuck in the Middle : The Challenges of User-Centered Design and Evaluation for Infrastructure. *Chi 2003*, 297–304. http://doi.org/10.1145/642611.642664

Fails, J. A., & Olsen, D. R. (2003). Interactive Machine Learning. In *Proceedings of the 8th international conference on Intelligent user interfaces IUI 03* (pp. 39–45). http://doi.org/10.1145/604045.604056

Fiebrink, R., Cook, P. R., & Trueman, D. (2011). Human Model Evaluation in Interactive Supervised Learning. In *Proceedings of the 2011 annual conference on Human factors in computing systems*

- *CHI '11* (p. 147). http://doi.org/10.1145/1978942.1978965

Françoise, J., Schnell, N., & Bevilacqua, F. (2013). A multimodal probabilistic model for gesture-based control of sound synthesis. *Proceedings of the 21st ACM International Conference on Multimedia - MM '13*, 705–708. http://doi.org/10.1145/2502081.2502184

Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, *11*(1), 10–18. http://doi.org/10.1145/1656274.1656278

Hartmann, B., Abdulla, L., Mittal, M., & Klemmer, S. R. (2007). Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (p. 145). http://doi.org/10.1145/1240624.1240646

Holtzblatt, K., Wendell, J. B., & Wood, S. (2004). *Rapid Contextual Design: A How-To Guide to Key Techniques for User-Centered Design*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Kapoor, A., Lee, B., Tan, D., & Horvitz, E. (2010). *Interactive Optimization for Steering Machine Classification*. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA, NY, USA: ACM Press. http://doi.org/10.1145/1753326.1753529

Katan, S., Grierson, M., & Fiebrink, R. (2015). Using Interactive Machine Learning to Support Interface Development Through Workshops with Disabled People. In *CHI '15: Extended Abstracts on Human Factors in Computing Systems 2015*. http://doi.org/10.1145/2702123.2702474

Lieberman, H., Paternó, F., Klann, M., Paternò, F., & Wulf, V. (2006). End-user development: An emerging paradigm. *End User Development*, *9*, 1–8. http://doi.org/10.1007/1-4020-5386-X

McPherson, A. (2017). Bela: An embedded platform for low-latency feedback control of sound. *The Journal of the Acoustical Society of America*. http://doi.org/10.1121/1.4987761

Monk, A. (2007). Lightweight Techniques to Encourage Innovative User Interface Design. In L. E. Wood (Ed.), *User Interface Design: Bridging the Gap from User Requirements to Design*. CRC Press LLC.

Myers, B. A., & Stylos, J. (2016). Improving API usability. *Communications of the ACM*, *59*(6), 62–69. http://doi.org/10.1145/2896587

Nielsen, J. (1994). *Usability Engineering*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Norman, D. A. (2013). *The Design of Everyday Things*. MIT Press.

Norman, D., & Draper, S. W. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Parkinson, A., Zbyszyński, M., & Bernardo, F. (2017). Demonstrating Interactive Machine Learning Tools for Rapid Prototyping of Gestural Instruments in the Browser. In *Web Audio Conference 2017* (pp. 1–2). Retrieved from https://qmro.qmul.ac.uk/xmlui/handle/123456789/26150

Patel, K., Bancroft, N., Drucker, S. M., Fogarty, J., Ko, A. J., & Landay, J. (2010). Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (pp. 37–46). http://doi.org/10.1145/1866029.1866038

Ritter, F. E., Baxter, G. D., & Churchill, E. F. (2014). *Foundations for Designing User-Centered Systems: What System Designers Need to Know about People*. London: Springer London Heidelberg New York Dordrecht.

Steiglitz, K. (1996). *A Digital Signal Processing Primer: With Applications to Digital Audio and Computer Music*. Addison-Wesley.

von Hippel, E. (1986). Lead Users: An Important Source of Novel Product Concepts. *Management Science*, *32*(7), 791–805. http://doi.org/10.1017/CBO9781107415324.004

Zbyszyński, M., Grierson, M., & Yee-king, M. (2017). Rapid Prototyping of New Instruments with CodeCircle. In *NIME 2017 Proceedings of the*

International Conference on New Interfaces for Musical Expression (pp. 227–230).

Zbyszyński, M., Grierson, M., Yee-king, M., & Fedden, L. (2017). Write once run anywhere revisited: machine learning and audio tools in the browser with C++ and emscripten. In *Web Audio Conference 2017* (pp. 1–5). Centre for Digital Music, Queen Mary University of London.

BIOGRAPHICAL INFORMATION

Francisco Bernardo is a software engineer, an interactive media artist, and a PhD candidate in Computer Science, at Goldsmiths, University of London. His research is focused on human-computer interaction approaches to toolkits that broaden and accelerate user innovation with interactive machine learning. He holds a graduate degree in Computer Science and Systems Engineering and a master's degree in Mobile Systems, both from the University of Minho. He also holds a master's degree in Management in the Creative Industries, with a specialism in Creativity and Innovation in the Music Industry, from the Catholic University of Portugal. Francisco has been working at the interface between industry and academia, mostly on corporate R&D, front-end software engineering, interaction design, greenfield product management, and applied research in publicly-funded projects at the intersection of art and innovation.

Mick Grierson is a Professor in the Department of Computing at Goldsmiths. His research explores new approaches to the creation of sounds, images, video and interactions through signal processing, machine learning and information retrieval techniques. Mick is also the director of the Daphne Oram Collection, and co-founder of the Daphne Oram Trust. Hardware and software based on his research has been widely used by world leading production companies, tech startups and artists including the BBC, Channel 4, Sigur Ros, Christian Marclay, Martin Creed and many others. He is Principal Investigator on the £1million Artificial Intelligence project MIMIC, which is a partnership between Durham University, Sussex University, and Google's Project Magenta.

Dr. Rebecca Fiebrink is a Senior Lecturer in the Department of Computing at Goldsmiths, University of London. Her research focuses on designing new ways for humans to interact with computers in creative practice. Fiebrink is the developer of the Wekinator, open-source software for real-time interactive machine learning, and the creator of a MOOC titled "Machine Learning for Artists and Musicians," which launched in 2016. She was previously an Assistant Professor at Princeton University, where she co-directed the Princeton Laptop Orchestra. She has worked with companies including Microsoft Research, Sun Microsystems Research Labs, Imagine Research, and Smule. She holds a PhD in Computer Science from Princeton University, a Masters in Music Technology from McGill University, and undergraduate degrees in Computer Science & Engineering and Music from The Ohio State University.