# Sketchument: Empowering users to build DMIs through prototyping

FILIPE CALEGARIO*, JERÔNIMO BARBOSA*, GEBER RAMALHO*, GIORDANO CABRAL† and GABRIEL FINCH†

*Centro de Informática, Universidade Federal de Pernambuco, Av. Prof. Luis Freire, s/n, Cidade Universitária. Recife – PE, Brasil. CEP: 50740-540
†Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Rua Dom Manuel de Medeiros, s/n. Dois Irmãos. Recife – PE, Brasil, CEP: 52171-900
E-mails: fcac@cin.ufpe.br; jbcj@cin.ufpe.br; glr@cin.ufpe.br; giordanorec@gmail.com; salsaman@gmail.com

**New interface artefacts are changing the way we interact with machines, and this is particularly important for the musical domain. Conversely, the growing do-it-yourself (DIY) culture is subverting the manufacturer–consumer model. Regarding music, software such as Pure Data and Max/MSP allows users to build their own interactive systems. These factors contribute to the emergence of new digital musical instruments (DMI). However, DMI creation still requires a strong technical background. Based on the importance of prototyping in the process of designing things, this paper presents the Sketchument, an environment devoted to help non-technical users to easily prototype DMIs, using multiple input modes and allowing the integration to other useful technologies. From a lo-fi paper prototype, to functional ones, passing through movies, questionnaires, interviews, Sketchument has been developed following the same prototyping philosophy we intend to propose to its users. The cyclic process of design–implementation-evaluation has produced valuable feedback from potential users, which has been very useful to back design choices and to push modifications.**

## 1. INTRODUCTION

Multitouch screens, accelerometers, 3D depth cameras… The whole way we interact with machines is changing, since gestures, movements and direct graphic manipulation are gradually replacing keys, buttons and pointers (Saffer 2008: 247). New interfaces bring new ideas to products, artefacts and paradigms (McGlynn, Lazzarini, Delap and Chen 2012). In the musical context, these new interface technologies combined open a wide range of possibilities in the creation of digital musical instruments (DMIs), artefacts with inputs (interface controllers) and outputs (sound synthesis modules) that connect according to a mapping strategy (Miranda and Wanderley 2006).

Contrary to acoustic instruments, which impose physical constraints to their design and fabrication concerning input (possible gestures) and output (sound quality), the design of DMIs has more freedom. The mapping strategies, which are the DMI core, are virtually countless (Rovan, Wanderley, Dubnov and Depalle 1997). Paradoxically, this advantage is a problem, since there is no established method or tool to guide the DMI designer or luthier to define how interfaces gestures should be adequately mapped into sound variables (Miranda and Wanderley 2006).

Given the fact that the user is the 'driving force' of innovation (Baldwin and Hippel 2010; Bogers, Afuah and Bastian 2010), why not allow the musicians to define their own mapping strategies, building their own instrument? Why not allow them to conduct experiments in DMI creation in order to find the most adequate mapping according to their preferences, intentions and context of use? (Hartmann 2009; Gray, Brown and Macanufo 2010: 290; Dow 2011; Gerber and Carroll 2011).

Giving power to the user is a strategy that converges to some current trends, such as the 'do-it-yourself' or 'Maker' culture (Dougherty and Frauenfelder 2005), which is spread out over the Internet and is also present in the musical field (Steiner 2005; Kirn 2011: 256). Nowadays users seem to want to be proactive, to participate in the construction of the solution, to adapt it to their interests, instead of being only consumers of finished products (Anderson 2012: 272).

In this context, it is important to remember that prototyping plays a central role in Design (Dow 2011), since it helps to identify flaws in the artefacts, to redirect and adjust them, to have a better understanding about how it works and to generate new ideas (Warfel 2009). Then, we can conclude that giving to the musicians the opportunity to easily create DMI prototypes could be a promising approach to the above-cited problem of DMI mapping.

Some musical systems that could help users to prototype DMIs already exist (e.g. Pure Data, Max/MSP, Chuck, SuperCollider) (McCartney 2002; Puckette 2002; Wang 2008). However, despite being powerful tools, most of them do not provide an adequate level of usability in order to be fully explored by a non-technical audience, such as musicians, designers and artists in general that have poor knowledge of programming and digital technology.

*Organised Sound* **18**(3): 314–327 © Cambridge University Press, 2013. doi:10.1017/S1355771813000290

In this paper, we present Sketchument, an experimentation environment devoted to help the non-technical public to prototype and then create DMIs. In this environment, pre-built modules, representing possible input controls and sound outputs, can be easily connected and reused, generating immediate feedback for the user. The process adopted to create Sketchument, also briefly presented here, is based on user-centred design principles, which include cycles of inspiration, investigation, prototyping, evaluation and change. Sketchument runs initially on the iPad platform, and it was evaluated by potential users several times during our development process, helping us to arrive at a better environment. The results are encouraging, including a favourable evaluation by the users as well as some improvements to be made.

The remainder of the paper is organised as follows. The next section presents some guidelines for the development of DMIs. The following sections present the development process and prototype evaluation by the users and some related projects. The last section presents some final considerations.

## 2. GUIDELINES FOR A DMI EXPERIMENTATION ENVIRONMENT

In order to solve the problem of DMI mapping, our approach concerns giving power to the user, who, by experimenting, will be able to find the suitable mapping for his or her context and intention. For this, we propose an experimentation environment, where the user can conceive, build and experiment with prototypes of DMIs.

Following this approach, a set of guidelines is necessary to help the development of an experimentation environment. We have compiled the following guidelines, as explained below: usability, feedback, diversity, configuration, integration and availability.

It is important to stress that, although these guidelines are related to the DMI experimentation environment, there is a clear connection between the environment and the created DMI. It mainly happens because, as part of experimentation process, the environment should provide ways to run the DMI. This means that, during performance, the DMI becomes a subset of the environment. Consequently, some guidelines of the environment may also be applicable to the built DMIs. However, in this work, we will focus only on the development of the environment.

### 2.1. Usability

Usability allows us to create software ‘more intuitive and effective for a person trying to accomplish tasks at a hand’ (Isbister and Schaffer 2008). It is related to how easily and satisfyingly an interface can be learned, used and memorised (Nielsen 1993: 362).

Besides the traditional human–computer interaction domain, it has been successfully used in particular contexts such as games (Isbister and Schaffer 2008), webpages (Nielsen 1993), and others – including DMIs themselves (Orio, Schnell and Wanderley 2001).

Thus, if we want users to quickly experiment with different setups aiming to find the most suitable one for his or her needs, we believe usability must be considered.

### 2.2. Feedback

Amongst the aspects that constitute usability, one of them seems to be particularly sensitive regarding a musical context: feedback. It is related to providing a clear, perceptible and real-time response to actions and modifications performed by the users and it can come from different sources: haptic (tactile and kinetic), sound and visuals (lights, images, etc.).

As shown in previous works, feedback can greatly increase the intuitiveness of an interactive music system (Jordà 2003) and can also improve the general playing accuracy of DMIs (O’Modhrain and Chafe 2000). Therefore, being so critical in a musical context, we believe feedback must be considered as an independent guideline for the environment.

### 2.3. Diversity

According to its own nature (Miranda and Wanderley 2006), every DMI is composed of a combination of input modules, which allow users to interact with the instrument, and output modules, which are responsible for the sound result. Defining the way those modules are combined, there is a set of mapping strategies.

The diversity guideline deals with the space the user will have in order to explore these possible combinations. It would ideally be expressive (large amount of modules), although not too complex or it would be impossible to be understood.

On the other hand, experimentation presupposes exploration and combination, and it is recommended to have a reasonable number of options that could enable this situation (Thomke 2003). However, in the context of this project a balance must be struck between allowing experimentation and not confusing understanding.

### 2.4. Configuration

As we aim to build an environment for creating DMIs, it is important to assume that we have no prior knowledge about the specific user context or intention. For that reason, the user should be able to customise settings in order to adapt the system for his or her individual needs. Therefore, we included configuration as one of our guidelines.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

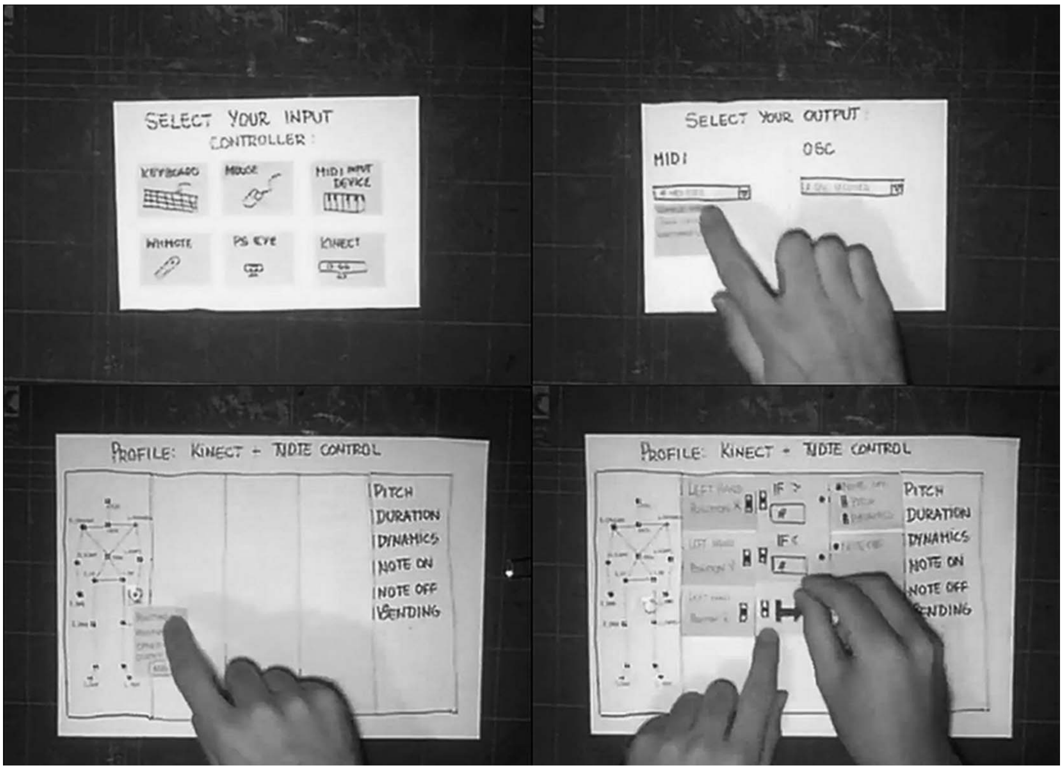

**Figure 1.** Lo-fi paper prototype.

### 2.5. Integration

As already noted in previous works (Chagas 1992; Schmeder and Freed 2008), communication protocols specially designed for a musical context (such as MIDI and OSC) are very powerful tools often used in DMI design.

A common explanation for this popularisation derives from their simplicity (ease of use) and 'the promise of interoperability with a diverse array of applications', which results in a great versatility and integration capability (Schmeder and Freed 2008).

Thus, if we want to allow the environment to be easily integrated with previous or legacy systems, benefiting from their functionalities, we believe integration must be considered as one of the guidelines.

### 2.6. Availability

According to Jordà (2005), a very common problem was the lack of availability in those systems. This means that, although intuitive to use during performance, those DMIs required a technical expert for setting the system up (including installation and calibration) before it was properly usable – a basic problem that could make the popularisation of a DMI impossible.

As our environment is intended for non-technical users, it should be available in an easy way – easily acquired and easily installed. Thus, availability was adopted as one of our guidelines.

## 3. DEVELOPMENT PROCESS AND EVALUATION

The Sketchument design process follows the same philosophy we intend to offer to the users – namely, using prototypes to learn and build a system with more mature design decisions anchored in potential users' judgement. This process is based on cycles of conception, prototyping and evaluation; steps synthesised from the study of available literature on user-centred design and innovation processes (Thomke 2003; Neves, Campos, Campello, Castillo, Barros and Aragão 2008; Warfel 2009; Dow 2011; IDEO 2011: 200; Maurya 2012: 240). So far, we have performed three full iterations.

Initially, two low-fidelity prototypes and a functional one were evaluated through informal interviews. This first feedback plus a competitor analysis resulted in a second prototype that was evaluated through questionnaires and interviews applied to potential users. Based on previous feedback, we prototyped a third version that was evaluated with real users through beta testing. This process is described in detail in the following sections.

### 3.1. Prototyping: first phase

The first concept of the experimentation environment was built as a lo-fi paper prototype, shown in Figure 1, and a stop-motion movie produced to collect feedback from potential users through informal interviews.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

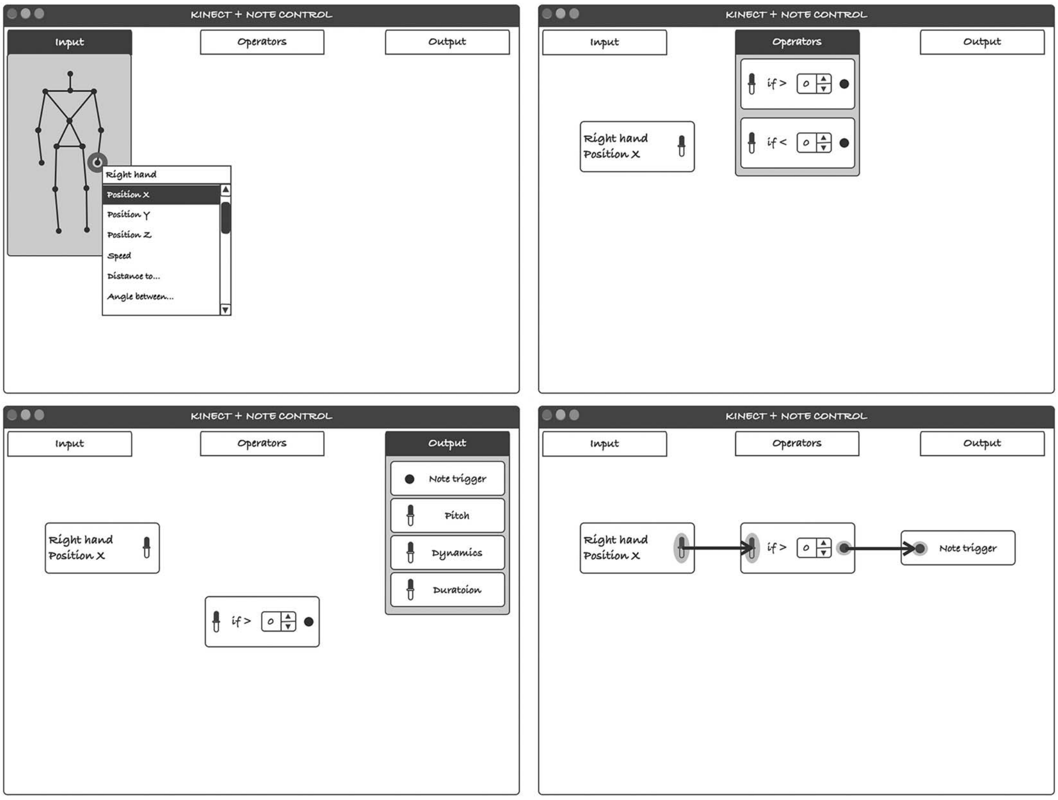

**Figure 2.** Lo-fi interactive presentation prototype.

After this, another lo-fi prototype was built using slides and links in Keynote (Apple's presentation software) (Apple Inc. n.d.), shown in Figure 2, with which potential users were able to test a mock-up of how the system would work.

In both prototypes, aspects such as note duration and pitch were presented as separate variables causing problems regarding system comprehension by the user. The informal evaluations demonstrated the importance of grouping those variables as individual modules, so the user could really use them as building blocks for DMIs.

Aiming to collect more complete and concrete results about real-time interaction, a functional prototype was developed using Kinect$^{TM}$ technology (Microsoft Corporation 2012) as input control, as shown in Figure 3. During informal evaluations, users complained about the lack of responsiveness due to excessive latency presented in the interface. As latency and delay contradict the feedback guideline, we have made a project decision to develop the system using the iPad platform, due to its low latency, good precision, large multitouch screen, availability and popularity (especially for musical applications).

### 3.2. Competitor analysis

Focusing on iPad, we ran a competitor analysis (Neves et al. 2008), whose objective was to identify interesting ideas, concepts or mechanisms that could be absorbed in the system development, and to understand gaps that would be used as design opportunities for Sketchument.

This method was based around two steps: first, the best ranked apps related to musical expression (such as instruments, controllers etc.) were selected from the App Store; second, this group was filtered into apps that shared a similar approach to Sketchument's: to have a set of editable musical blocks, which, when combined, could be used for musical experimentation. Finally, from this set we selected the direct competitors:

- Audulus, a modular synthesiser with a minimalistic interface, where the modules can be connected through drag and drop gestures over the connectors.
- Beatsurfing, a customisable controller where objects can trigger MIDI events with simple gestures.
- Lemur, a programmable controller where the user can build her or his own interface and trigger MIDI or OSC events.
- Reactable mobile, a simpler version of the Reactable (Jordà, Kaltenbrunner, Geiger and Bencina 2005), where objects can be combined in order to control musical variables in an interactive environment.
- TC-11, a programmable modular synthesiser with no buttons or faders, where the controllers are

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

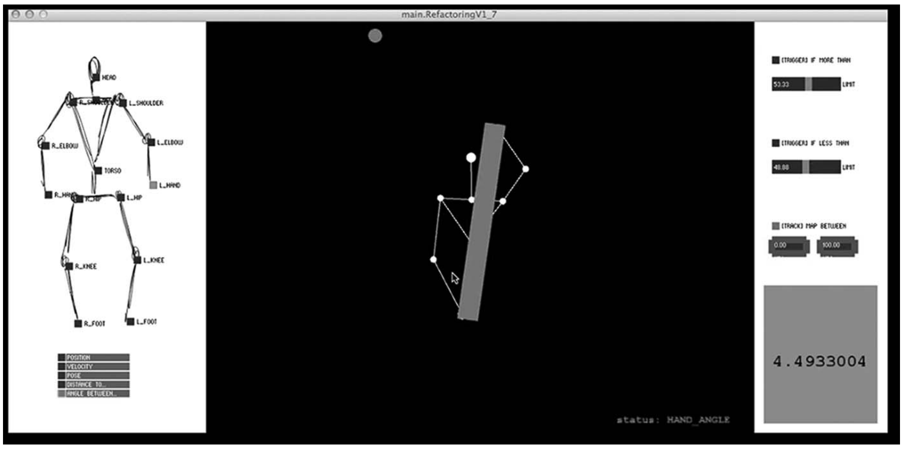

**Figure 3.** Functional prototype developed using Microsoft Kinect$^{TM}$ sensor.

based on finger position and relative data about them (distance, angle etc.).

- Tabletop, a modular environment where devices, similar to real audio devices from recording studios, can be combined and mixed.

This competitors group was analysed following the project guidelines with the addition of a further guideline, which is intrinsic to the iPad multitouch interface, which is presented in literature as an important feature for musical applications on this kind of platform (McGlynn et al. 2012): the use of gestures to control musical variables.

Concerning usability overall, the apps present organised interfaces. For example, the ones where the connections between modules are explicit are easier to understand. The modules, in general, are associated with self-explanatory icons, which help the user to memorise their function. All the competitors have introductory examples in order to show to the users how to use the system, and the majority have quick help guides inside the app or a link to external documentation.

With regard to the feedback guideline, we could see that animations are commonly used to show the user the connections working in real-time. In all the tests, latency was low or unperceivable. Besides this, the apps that produce their own audio allowed more agile experimentation, basically because the user is not dependent on another app or device to provide the audio response.

Diversity was objectively measured through the quantity of the modules provided by the system. Concerning the integration guideline, we looked for app features that could enable integration with other systems. The configuration guideline was related to the possible ways the module settings could be modified by the user – for instance, changing colour, position and size.

Due to App Store's online distribution and its ease of access and installation, all the apps on this online platform can be considered to have a high availability value. Thus, we chose price as the availability criterion to compare different apps.

Table 1 summarises the objective comparison between competitors. Analysing it, we see that the majority of the apps have an expressive number of modules (a dozen or more) and all the competitors have at least one means to integrate with other systems, MIDI being the most common (and only one deals with OSC connections). All the apps allow access to the module settings, but few deal with appearance customisation (colour, size etc.).

Concerning availability, all of the analysed prices are high in comparison to the common App Store prices, which are from US$0.99 to US$1.99.

Regarding gestural control, only three competitors use gestures, mostly used in a simple or limited way. Basically, the gestures are used in configuration mode and not during play time to control musical variables.

### 3.3. Prototyping: second phase

Based on feedback received from the initial prototypes and the knowledge acquired in competitor analysis, we developed a functional prototype in the iPad (demo shown on an online video).[1] It is a modular environment with thirteen modules, where the explicit connection (or mapping) can be easily made by dragging input to output connectors. Below, the prototype is described in detail.

As shown in Figure 4, the floating menu presents the set of input control objects and sound output objects. In 'edit mode', both can be dragged into the central area and the mapping is established simply by

[1] http://vimeo.com/filipecalegario/sketchumentprototype.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

**Table 1.** Objective comparison between competitors

| Competitor apps | Usability | Feedback | Diversity | Integration | Configuration | Availability | Gestures |
|---|---|---|---|---|---|---|---|
| Tabletop | • Examples<br>• User guide<br>• Graphical connection<br>• Representative icons | • Sound<br>• Reactive GUI widgets | 28 | • MIDI in<br>• Music Library access<br>• SoundCloud<br>• WAV export<br>• AudioCopy<br>• WIST | • Modules specific settings | Free + US$13.00 optional modules | • No gesture, only interaction with GUI widgets |
| Audulus | • Examples<br>• User guide<br>• Graphical connection<br>• Representative icons | • Sound<br>• Connection animations | 29 | • Audiobus<br>• JACK<br>• MIDI in<br>• Samples import | • Modules specific settings | US$14.00 | • Drag to move<br>• Drag to connect |
| Reactable | • Examples<br>• User guide<br>• Graphical connection<br>• Representative icons | • Sound<br>• Connection animations | 14 | • Samples import<br>• WAV export<br>• Download from web | • Modules specific settings | US$9.99 | • Swipe to activate<br>• Drag to move |
| Beatsurfing | • Examples<br>• No graphical connection<br>• Abstract icons | • No sound<br>• Reactive GUI widgets | 4 | • MIDI out | • Object size and colours<br>• Modules specific settings | US$11.99 | • Drag to play |
| TC-11 | • Examples<br>• User guide<br>• No graphical connection<br>• No icons | • Sound<br>• Reactive GUI widgets | 76 | • Audiobus<br>• AudioCopy | • Color theme<br>• Display style<br>• Modules specific settings | US$29.99 | • No gestures, only raw touch position data |
| Lemur | • Examples<br>• User guide<br>• No graphical connection<br>• Icons | • No sound<br>• Reactive GUI widgets | 17 | • MIDI in<br>• MIDI out<br>• OSC in<br>• OSC out | • Object size and colours<br>• Modules trigger events | US$49.99 | • No gesture, only interaction with GUI widgets |

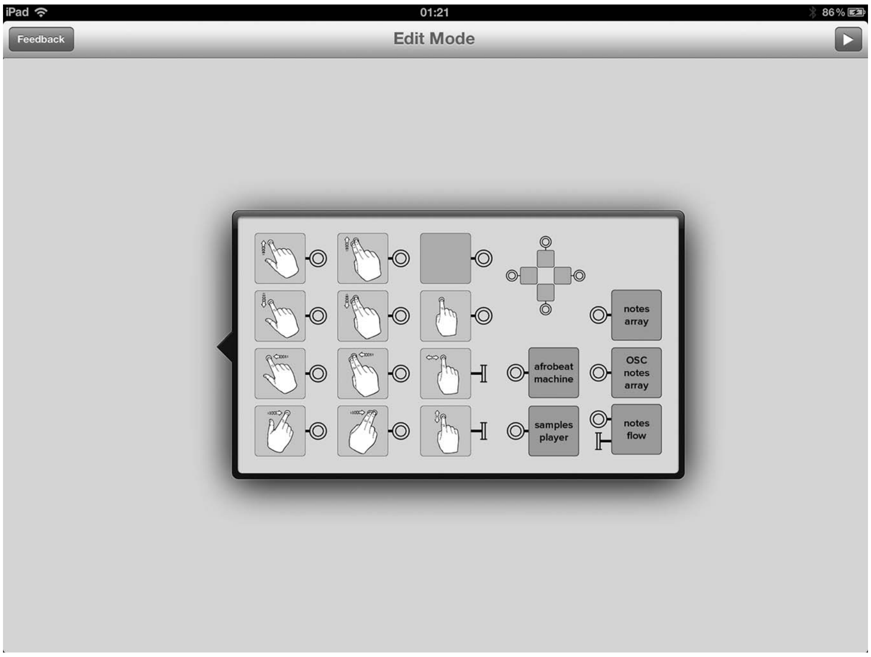

**Figure 4.** Sketchument Prototype version 2: floating menu and the available modules for use.

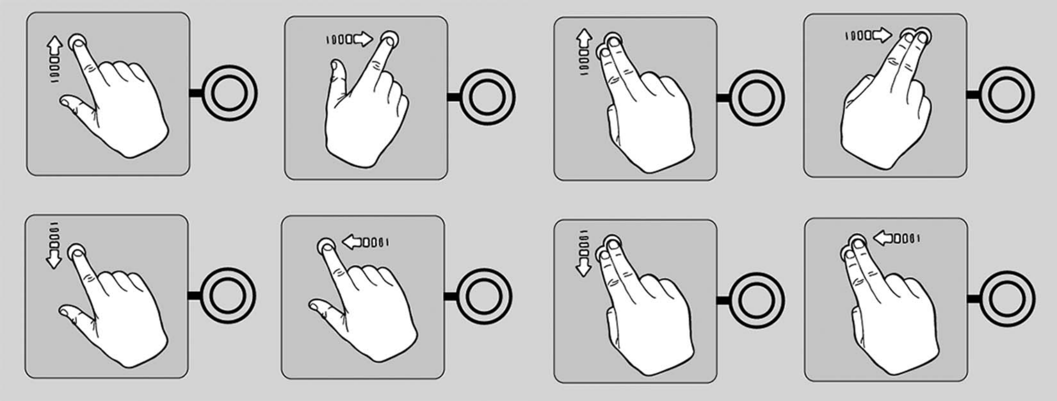

**Figure 5.** Gestural input modules.

dragging their connectors. The system highlights the possible connections to help the user.

Switching to 'play mode', the user can use the gestures and input control to play the DMI, listening to the sound results in real-time. In short, in a simple and agile way, users can make their ideas concrete, following a trial-and-error approach.

### *3.3.1. Input objects*

Following the interaction guidelines proposed to multitouch-based artefacts (McGlynn et al. 2012), eight gestural objects were implemented in this prototype, as can be seen in Figure 5 (one-touch swipes and two-touch swipes in four directions).

Among the input possibilities, there is also a resizable touchable object, which works as a customisable button that can be used as an event trigger in play mode. The event is triggered by touching or by passing a finger through the button.

Besides these objects, aiming to provide haptic feedback, there is an ensemble of touchable objects that can be used together with a joystick for capacitive screens (Figure 6). The idea of tangible objects is related to the lack of haptic feedback of multitouch surfaces – described by musicians as one of the strongest characteristics of acoustic instruments (Magnusson and Mendieta 2007).

### *3.3.2. Output objects*

Output objects are responsible for the sound synthesis of the environment. According to Schloss (1990), users of musical systems could control musical processes in

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

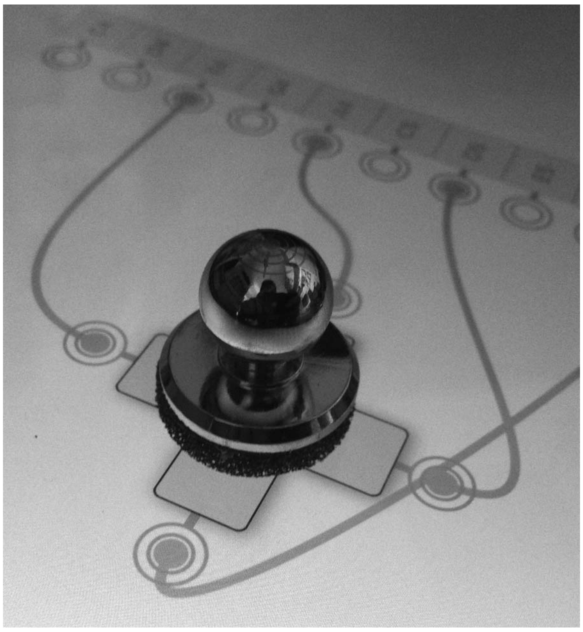

**Figure 6.** Joystick for capacitive screens.

three different levels: timbral level (a 'microscopic' control synthesis parameters), note level (the most familiar), and musical-process level (a 'macroscopic' control of meta-musical processes, like an improvisation).

In the Sketchument concept, there are no technical or conceptual constraints concerning output objects. All these three categories could be available in the system. For instance, concerning the timbral level, there could be an output module with which the user could activate and adjust in real-time the number of frequencies of an additive synthesiser, or another module that could control parameters of an audio effect applied to an input sound source. However, due to scope reasons, we decided to initially focus on note and musical process control.

In order to implement these controls, we have developed two mechanisms: sound trigger objects and integration objects. Both can be, in principle, used to control notes or processes by simply associating them, through a graphical connection, with any input object.

Regarding the sound trigger objects, which control sound events (e.g notes, audio samples, sound effects) when an action input occurs, we implemented the following.

- The note array: composed by a set of musical notes that can be triggered by input module objects. Each note has a small connector that allows its value to be changed. So far, its tone is not customisable and sounds like a piano.
- The sample trigger: allows audio samples to be triggered when input actions occur. It is so far composed by six different boxes, each one containing a small connector (that allows it to be triggered) and an unique pandeiro sample (sounds produced on the Brazilian tambourine by player's thumb, fingertips, heel, palm of the hand, etc.).

Concerning the integration objects, which group objects that are responsible for allowing users to integrate Sketchuments with other environments or modules, we implemented the following.

- The OSC object: responsible for connecting the Sketchument with other software and devices that implement Open Sound Control (OSC) protocol (Wright 2005). To exemplify this possibility, we have encoded a simple protocol using hard-coded messages that are sent via network connection and interpreted by OSCulator software, installed in the computer, which will be responsible for controlling an external sound synthesiser.
- The Pure-Data-based objects: represent the capacity Sketchument has to create new modules based on previously existent Pure Data patches. With this functionality, we aim to explore possible contributions by technical users that could easily integrate their own patches to our prototype. This easy integration was achieved due to the libPd library (Brinkmann 2012: 124).

As an example developed in the current Sketchument version, we have integrated a Pure Data patch known as Afrobeat Machine (Jácome 2009) – a musical supportive patch that allows performers to turn tracks on and off, and to change the tempo and the song played. Except for some minor adjustments to the original patch, no modifications were made in its kernel during the porting process from Desktop to iOS. The Afrobeat Machine output object has seven sub-modules: the first four are related to choosing the current song and they might be triggered by separated input actions; the sub-modules 'bpmUp' and 'bpmDown' allow tempo adjustment of the current music; and the 'TrackSel' allows for sequentially activating (or deactivating) the song tracks.

### *3.3.3. Mapping*

With the approach of experimentation as a solution for the huge number of possible combinations for mappings in a DMI, the environment should provide rapid ways of connecting, disconnecting and reconnecting modules. As seen in competitor analysis, explicit graphical connections can provide this agility in mapping, and that is why it was implemented in this prototype.

In edit mode, input objects can be connected to one or more output objects by simply dragging the connectors between them. It is a graphic and natural operation that saves time for making more combinations.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

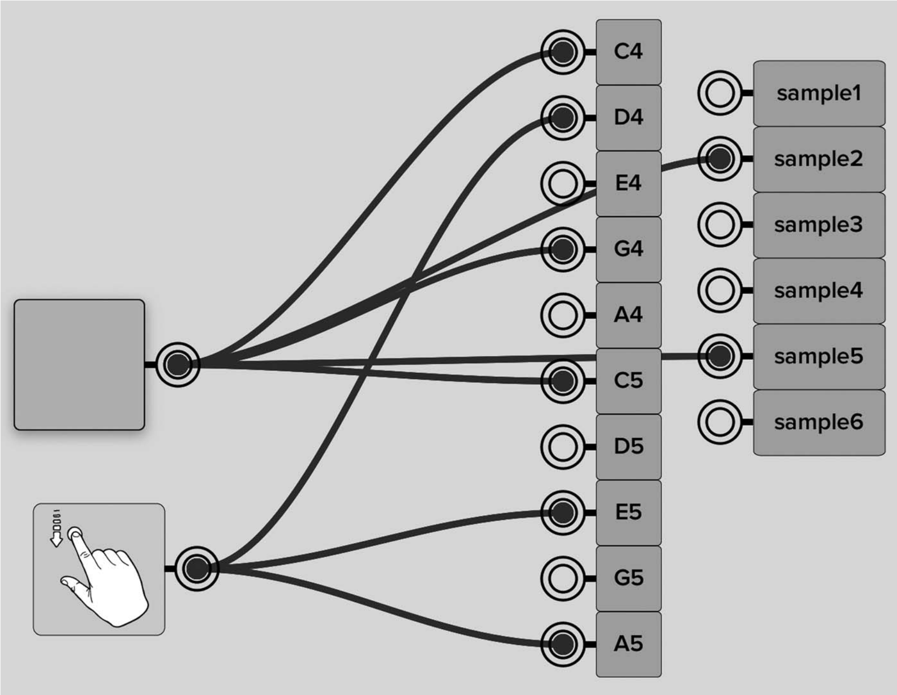

**Figure 7.** Multiple connections using different modules in Sketchument.

In play mode, triggering an input object will result in specific output musical results. It is important to highlight that different kinds of output could be used for the same input, providing more flexibility to experiment. Thus, for example, it is possible to use touchable buttons or gestures to trigger chords and samples, as shown in Figure 7.

The mapping strategy provided by Sketchument can be authored in this simple way because both input and output objects have a high level of abstraction. This means that some details are transparent for the non-technical user, making the interaction easier and more agile, focusing the user on dealing with the musical subject instead of technical ones.

### 3.4. User evaluation

Aiming to receive richer and more structured feedback from users, we decided to submit our prototype to a deeper evaluation method, which consisted of questionnaires and semi-structured interviews.

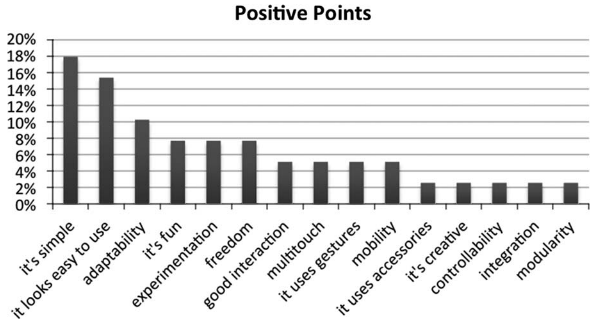

**Figure 8.** Positive points raised during evaluation.

### 3.5. Questionnaires

A group of potential users were asked to watch an online video introducing the prototype and then asked to answer an online questionnaire containing open questions related to it. The first part was to fill in personal information concerning age and relationship with technology and music. Then, there was a set of questions about the Sketchument. The questions were open, allowing participants to write their impressions about the interface, to list good and bad points about the system and to suggest improvements. With this approach, we focused on extracting emerging opinions from potential users, not constraining them with multiple-choice forms.

After two weeks, 87 people had taken part in the experiment. From these, 31 were selected regarding their profile: they had high familiarity and previous experience with tablets. Based on the selected answers, a thematic analysis (text extracts grouped by similar themes) (Tanaka, Parkinson, Settel and Tahiroglu 2012) was applied and then common themes were highlighted, so that they could be presented in percentage form.

Some users said that the system was not providing enough visual feedback for a clear understanding

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

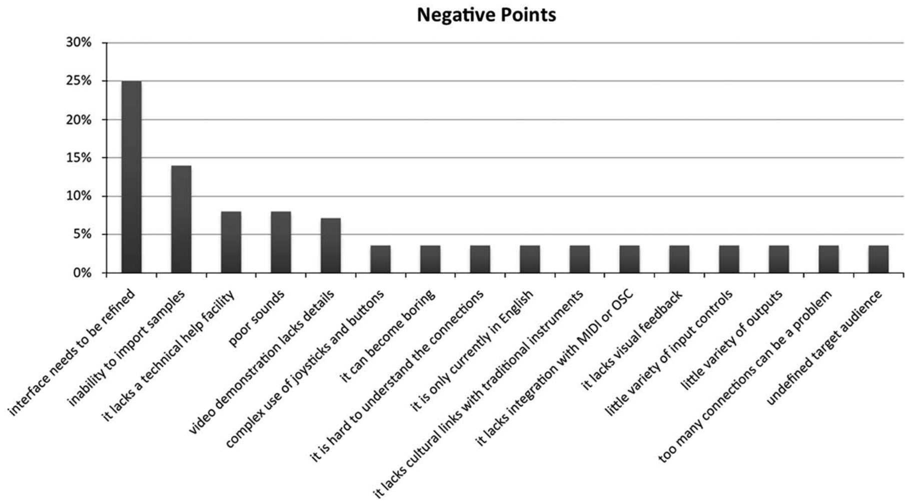

**Figure 9.** Negative points raised during evaluation.

about how the system works. As illustrated in Figure 8, the main positive aspects mentioned in the questionnaires are related to simplicity, ease of use and adaptability. This reinforces the idea that usability, stated as the most important guideline, is also one of the strongest aspects of our prototype.

Regarding the negative points, as shown in Figure 9, the enhancement of interface aesthetics is the most mentioned topic at 25 per cent and it is an important point to be considered for future versions.

It is possible to verify that most of the problems were related to input/output (I/O) diversity and customisation aspects ('inability to import samples', 'little variety of outputs', 'little variety of input controls' and 'it can become boring'). Besides, feedback aspect ('it lacks visual feedback') was once again mentioned as a critical point in the system.

### *3.5.1. Interviews*

In this step, seven potential users were invited to use of the system and then to answer a semi-structured interview (Preece, Rogers and Sharp 2002: 68).

The joystick for the capacitive screen, proposed as haptic feedback, was not well received by the participants. One of the interviewees stated that he was afraid to use the joystick because it could damage the iPad screen. There was also a criticism about the low-quality tracking of the joystick due to problems with physical contact between the lower part of the joystick and the screen.

During the app evaluation, users were confused about how to start, because they first had to touch the screen for the floating menu to appear. Some users took a long time to understand this first action and were surprised when the floating menu was shown. Another detected problem was related to how an object could be deleted. A double tap was used for this purpose, but it was not well evaluated by the users, who several times deleted the objects without intending to.

One suggestion for improvement was to have a sample import function, also mentioned in the questionnaires and related to the customisation and diversity guidelines. An interviewee stated that the most innovative point of the Sketchument is to be a customisable, modular, gestural controller, which corroborates one the guidelines we were trying to follow. Therefore, the focus should be on the gestural input and MIDI and OSC output modules, presenting a new kind of controller that does not yet exist on the market.

## 3.6. Prototyping: third phase

Based on the feedback received from potential users during evaluation, we performed some enhancements to the previous prototype and developed the third prototype, as shown in Figure 10. According to user acceptance, the tangible object was removed from the set of input objects. Due to confusion reported by the users during the object placement task, we focused on improving the interface by changing the way input and output modules were presented. Two tabs were implemented: on the left one were placed the input objects; on the right one, the output objects. Besides this, because of delete problems in the main edit canvas, a recycle bin was implemented and placed in the middle of the canvas, and only activated by proximity.

In order to help the user, a quick guide, showing the main functionalities of the system, was added and, preparing for beta test, an embedded user feedback form was placed in the interface.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

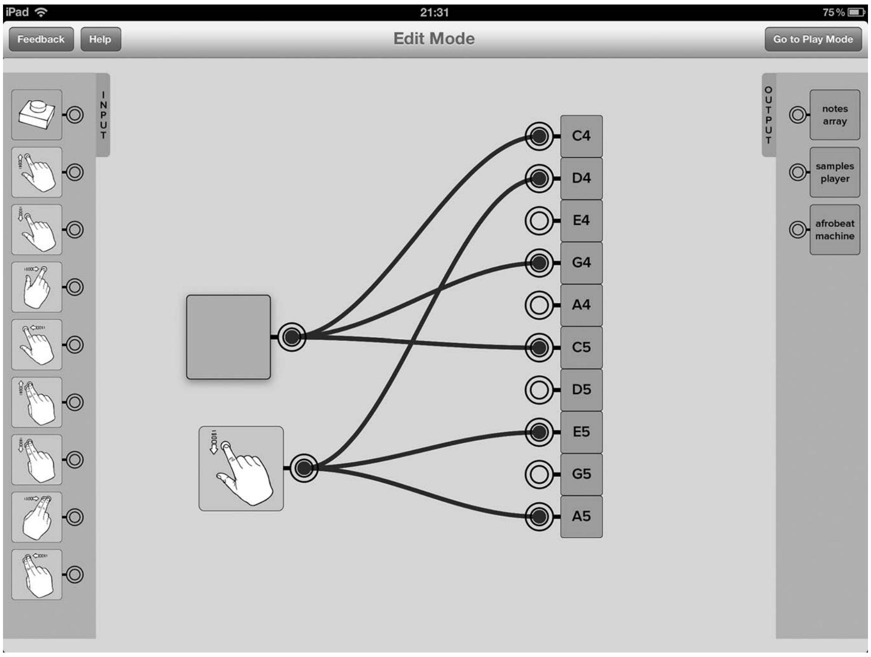

**Figure 10.** Sketchument Prototype version 3: interface overview.

**Table 2.** Beta testers' use data

| User | Duration (minutes) | Number of sessions | Duration average (minutes) | Checkpoints |
|---|---|---|---|---|
| User 1 | 32:56 | 5 | 06:35 | 1014 |
| User 2 | 05:15 | 2 | 02:38 | 225 |
| User 3 | 01:45 | 1 | 01:45 | 52 |
| User 4 | 02:13 | 1 | 02:13 | 118 |
| User 5 | 07:48 | 1 | 07:48 | 447 |
| User 6 | 00:56 | 1 | 00:56 | 56 |
| User 7 | 10:37 | 1 | 10:37 | 297 |
| User 8 | 18:44 | 16 | 01:10 | 2240 |
| User 9 | 08:36 | 4 | 02:09 | 322 |

## 3.7. Beta test

For the beta test, the application could be installed on the user's own device through an ad-hoc distribution using the online platform TestFlight.[2] Quantitative evaluations were provided by nine users via an open form, available inside the app.

### *3.7.1. Use data*

The use data collected comprises:

- Number of critical crashes: how many times did the system suddenly stop working due to an error?
- Number of use sessions: how many times did a user open the system?
- Average duration of sessions: how much time did users spend using the prototype?
- Number of checkpoints reached: how many functionalities were used by a specific user?

Results are exhibited in Table 2.

Although User 1 and User 8 seemed to be engaged with the system, the majority of participants spent less than 10 minutes using it. A possible explanation for this is the low diversity of objects available so far, which could have caused apathy in users to experiment with more combinations. No critical crashes were detected during the sessions.

### *3.7.2. Qualitative feedback*

The users, using the online distribution platform, had the opportunity to test the application in their own time and environment, without being aware of participating in a laboratory experiment. We believe this intimacy brings deeper results concerning the interface.

Although the system had positive feedback from users (such as: 'I would certainly use the app, … the

[2] http://testflightapp.com.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

mere fact that I can arrange the positions of the buttons as I want, and create the instrument of my choosing makes it worthwhile for me'), as is common in beta tests, most of the feedback was focused on flaws and improvement suggestions. The principal ones are described as follows.

One suggestion was that the output objects could give some kind of audible feedback in edit mode in order to choose the best output form for a particular input. Talking particularly about the Sample Player, one user wrote: 'otherwise you would have to add a button on each one (of the outputs), go into Play Mode, listen to each one, then return to Edit Mode and remove the buttons which were not wanted'. Actually in the way in which it is implemented, this functionality breaks the principle of immediate results.

Some users commented that it was difficult to visually distinguish between edit mode and play mode. This caused further confusion when the users thought the app was in play mode, tried to press a button, and, instead, ended up changing the object configuration in edit mode. This type of confusion was common, and it stresses the need to present the modes in visually different ways.

As a bug, the testers reported the lack of simultaneity when more than three buttons were pressed at the same time: 'it is noticeable that the buttons do not react simultaneously'. There was also a conflict between buttons and gestures mentioned by the users. When the user made a gesture at the same time as pressing a button, inconsistent results were produced.

Another limitation of the prototype was the inability to press a button and have a signal sent continuously to the output to which it is attached. There was a suggestion to include an object that would function like a toggle button in Pure Data or Max/MSP, whereby one touch makes it permanently activated and a second touch deactivates it.

From the statement: 'I know that it could be the fashionable part to use multitouch gestures for interaction, or to slide a finger, but I never felt the urge to do so. I only used the buttons', we can perceive that the use of gestures is not considered important by some users. Some possible reasons for this could be the way the selection of gestures is presented and how they are associated with actions. In an attempt to give more options to gesture use, one possible functionality for the next version is to divide the screen into zones and allow mapping of gestures by zone, which would allow a greater variety of combinations and more diversity (of expression) for the users.

Further information about the system and the process can be found online.[3]

## 4. RELATED PROJECTS

Some existing musical tools, which have not been originally created to serve as a prototyping environment for designing DMIs, can be, however, used as one. Although they present fast feedback, I/O diversity, customisation and possible integration with other systems and devices, the majority present usability issues with respect to non-technical users.

Among the modular musical systems, applications that provide means for interconnecting modules to obtain a musical result, there are flowchart-oriented systems (e.g. Max/MSP and Pure Data: Puckette 2002); musical programming languages (e.g. SuperCollider, CSound and Chuck: McCartney 2002; Wang 2008); and tangible or physical interfaces (e.g. Molecular Synth and Reactable: Jordà, Geiger, Alonso and Kaltenbrunner 2007; Feldman 2012).

Since musical programming languages are totally based on text, their usability is compromised due to the absence of visual and structural cues. On the other hand, flowchart-oriented systems are user-friendly, yet powerful, programming tools that can be used to build real time interactive musical applications. They are more usable than textual programming, but a certain level of expertise is still required to obtain non-trivial results with them. Some tangible and physical interfaces are really intuitive, providing good usability for non-technical users. Unfortunately, they are difficult to acquire, because of either cost or availability.

Another category of systems that can serve as DMI prototyping environment is Mapping Systems, whose mapping strategies between input and output messages is generally based on OSC or MIDI protocols (Steiner 2005) (e.g. OSCulator, junXion and libmapper: Rudraraju 2011). Those systems make mapping a simple task, saving a lot of time and effort during a DMI prototyping process. However, they are just responsible for one part of the DMI design. With the mapping system alone the whole DMI cannot be completed, as it is dependent on the control interface and the sound production module. Besides this, OSCulator and juXion are based on tabular interfaces, which makes it difficult to visualise one-to-many and many-to-one mappings.

## 5. CONCLUSIONS

Technology often simplifies crafts, allowing non-technical users to conduct experiments and to discover new ways to express themselves. For instance, photography gave people who did not paint the ability to register reality and/or produce artworks.

The underlying principle of Sketchument is to give to the non-technical audience an adequate and attractive prototyping environment, so they can work as if they were luthiers. As our preliminary, but

[3] See http://www.cin.ufpe.br/ ~ fcac/sketchument.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

encouraging results suggest, this combination of 'do it yourself'/'maker' culture along with some general design principles is a good pathway to be explored in developing DMIs, particularly in tackling the central issue of I/O mapping.

The current version of Sketchument still only presents a few choices for providing a more expressive experimentation environment. Nevertheless, the user-centred prototype-based process of developing Sketchument has proven to be promising, helping us to make good design decisions and allowing incorporation of users' suggestions and criticisms that enable evolution of the system.

Sketchument is still growing and, as its development process is cyclic, it is not supposed to stop being updated. We prioritised growth with knowledge over a simple technical implementation, which is why it is not completely developed, but it has a lot of aggregated knowledge for further implementation.

Regarding the problems identified during the evaluation cycles (mostly related to feedback, diversity and configuration aspects), future improvements include: to allow importing custom samples; to allow sound recording; to have more outputs in general; to use the iPad accelerometer as input; to use the iPad magnetic sensor as input; and to have more inputs in general. Primarily, we intend to adapt Sketchment software architecture in order to easily incorporate (via plugins for instance) input and output objects developed by the computer music community. Sketchument keeps its vocation as a DMI prototyping environment, but it may benefit more from third-party contributions.

## REFERENCES

Anderson, C. 2012. *Makers: The New Industrial Revolution*. New York: Random House Business.

Apple Inc. n.d. Apple Developer. http://developer.apple.com.

Baldwin, C. and Hippel, E.von. 2010. Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation. *Harvard Business School Finance* (November 2009): 1–37.

Bogers, M., Afuah, A. and Bastian, B. 2010. Users as Innovators: A Review, Critique, and Future Research Directions. *Journal of Management* **36**(4): 857–75.

Brinkmann, P. 2012. *Making Musical Apps: Real-Time Audio Synthesis on Android and iOS*. Sebastopol, CA: O'Reilly Media.

Chagas, P. 1992. Le MIDI et la musique électronique: Quelques remarques esthétiques et techniques. *Revue informatique et statistique dans les sciences humaines* **28**: 15.

Dougherty, D. and Frauenfelder, M. 2005. The Making of Make. *MAKE Magazine* **1**: 7.

Dow, S. 2011. How Prototyping Practices Affect Design Results. *Interactions* **18**(3): 54.

Feldman, T. 2012. Molecular Synth. http://www.moleculesynth.com

Gerber, E. and Carroll, M. 2011. The Psychological Experience of Prototyping. *Design Studies* **33**(1): 64–84.

Gray, D., Brown, S. and Macanufo, J. 2010. *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers*. Sebastopol, CA: O'Reilly Media.

Hartmann, B. 2009. *Gaining Design Insight Through Interaction Prototyping Tools*. PhD dissertation, Stanford University.

IDEO. 2011. *Human-Centered Design Toolkit: An Open-Source Toolkit to Inspire New Solutions in the Developing World*. n.p.: Authorhouse.

Isbister, K. and Schaffer, N. 2008. *Game Usability: Advice from the Experts for Advancing the Player Experience*. Burlington, MA: CRC Press.

Jácome, J. 2009. Afrobeat Machine. http://jarbasjacome.wordpress.com/2009/10/15/fela-kuti-pure-data-fela-day-recife (accessed on 29 August 2013).

Jordà, S. 2003. Interactive Music Systems For Everyone: Exploring Visual Feedback as a Way for Creating More Intuitive, Efficient and Learnable Instruments. In *Proceedings of the Stockholm Music Acoustics*, 1–6.

Jordà, S. 2005. *Digital Lutherie: Crafting Musical Computers for New Musics' Performance and Improvisation*. PhD thesis, Universitat Pompeu Fabra.

Jordà, S., Kaltenbrunner, M., Geiger, G. and Bencina, R. 2005. The Reactable. *Proceedings of the International Computer Music Conference*, 2–5.

Jordà, S., Geiger, G., Alonso, M. and Kaltenbrunner, M. 2007. The ReacTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, 139–46.

Kirn, P. 2011. *Keyboard Presents the Evolution of Electronic Dance Music*. Milwaukee, WI: Backbeat Books.

McCartney, J. 2002. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal* **26**(4): 61–8.

McGlynn, P., Lazzarini, V., Delap, G. and Chen, X. 2012. Recontextualizing the Multi-Touch Surface. *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*.

Magnusson, T. and Mendieta, E. 2007. The Acoustic, the Digital and the Body: A Survey on Musical Instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 94–9.

Maurya, A. 2012. *Running Lean: Iterate from Plan A to a Plan That Works*. Sebastopol, CA: O'Reilly Media.

Microsoft Corporation. 2012. *Human Interface Guidelines: Kinect for Windows v1.7.0*. http://go.microsoft.com/fwlink/?LinkID=247735 (accessed on 31 July 2013).

Miranda, E.R. and Wanderley, M.M. 2006. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Middleton, WI: A-R Editions.

Neves, A., Campos, F., Campello, S., Castillo, L., Barros, S. and Aragão, I. 2008. XDM: Métodos Extensíveis de Design. In *Anais Do 8º Congresso Brasileiro De Pesquisa e Desenvolvimento Em Design*, 249–59.

Nielsen, J. 1993. *Usability Engineering*. Cambridge, MA: AP Professional.

O'Modhrain, S. and Chafe, C. 2000. Incorporating Haptic Feedback Into Interfaces For Music Applications. In

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press

*Proceedings of the International Symposium on Robotics with Applications, World Automation Conference*.

Orio, N., Schnell, N. and Wanderley, M.M. 2001. Input Devices for Musical Expression: Borrowing Tools from HCI. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (*NIME*), 1–4.

Preece, J., Rogers, Y. and Sharp, H. 2002. *Interaction Design: Beyond Human-Computer Interaction*. Chichester: Wiley.

Puckette, M. 2002. Max at Seventeen. *Computer Music Journal* **26**(4): 31–43.

Rovan, J.B., Wanderley, M.M., Dubnov, S. and Depalle, P. 1997. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *Proceedings of Kansei: The Technology of Emotion Workshop*. Genoa: Associatzione di Infomatica Musicale Italiana: 26–73.

Rudraraju, V. 2011. *A Tool for Configuring Mappings for Musical Systems using Wireless Sensor Networks*. MA thesis, McGill University.

Saffer, D. 2008. *Designing Gestural Interfaces*. Sebastopol, CA: O'Reilly Media.

Schloss, W. 1990. Recent Advances in the Coupling of the Language Max with the Mathews/Boie Radio Drum. In *Proceedings of the International Computer Music Conference*.

Schmeder, A. and Freed, A. 2008. uOSC: The Open Sound Control Reference Platform for Embedded Devices. In *Proceedings of the 8th International Conference on New Interfaces for Musical Expression (NIME)*.

Steiner, H. 2005. Building your Own Instrument with Pd. In *Proceedings of the 1st International Pd Conference*.

Tanaka, A., Parkinson, A., Settel, Z. and Tahiroglu, K. 2012. A Survey and Thematic Analysis Approach as Input to the Design of Mobile Music GUIs. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (*NIME*).

Thomke, S. 2003. *Experimentation Matters: Unlocking the Potential of New Technologies for Innovation*. Cambridge, MA: Harvard Business School Press.

Wang, G. 2008. *The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environmentality*. PhD thesis, Princeton University.

Warfel, T.Z. 2009. *Prototyping: A Practitioner's Guide*. New York: Rosenfeld Media.

Wright, M. 2005. Open Sound Control: An Enabling Technology for Musical Networking. *Organised Sound* **10**(3): 193–200.

https://doi.org/10.1017/S1355771813000290 Published online by Cambridge University Press