The Modular Surface

Paweł Boniecki
KTH
boniecki@kth.se

Daniel Molin
KTH
dmol@kth.se

John Turesson
KTH
johntu@kth.se

Johannes Törnqvist
KTH
jtornqv@kth.se

ABSTRACT
Interaction with a computer typically involves a keyboard and a mouse, where the mouse is a very general tool used for many different kinds of functions. The interaction is constrained by pointing and clicking with the mouse, an abstraction that often lead to unintuitive interactions. Many UI's are also influenced by the physical world and its interactions, showing that there is a growing gap between what is represented on the screen and how it is interacted with. We present a concept, the Modular Surface, which gives the user a set of general physical controls that is put on the surface and used to interact with a common computer GUI. The research shows that this kind of WYSIWYG interaction is needed in the field and the current technology suggest that realizing the Modular Surface concept could be feasible in a time frame of ten years from now.

Author Keywords
Tangible interfaces; skeuomorphism; touch-screens; music production; DAW.

INTRODUCTION
Software originally designed for WIMP interfaces (Windows, Icons, Mouse, Pointer) are being ported to touch-screen devices as these are becoming ever more popular. Current visions of the future commonly feature large touch-screens or gesture-based interfaces (see for example the movies *Minority Report*, *Iron Man* and *Star Trek Into Darkness*). The most striking features of these examples are simply the visuals, displaying a tremendous amount of information at once, shifting rapidly from one context to another with the flick of a hand and projecting detailed 3D holograms anywhere in the room. The interaction itself is fairly basic, however, made out of mostly point-and-click (or point-and-select, tapping, etc.) actions.

During the 50's, the first analog sound mixers were introduced to consumers around the world. Since then their interface and the way we interact with the mixer and other audio equipment have stayed almost the same. When we today have our mixers represented digitally on our computer screens the interaction need to be mapped through some other kind of interaction. Being able to simulate any physical interaction while providing functionality otherwise impossible in the physical realm makes the computer an incredible, general tool. But without a corresponding interface the interaction will not feel natural. It may look like you are painting with a brush or turning a knob on the screen, while you are actually just moving a mouse on the table next to the screen or moving your hand across a glass surface. In this paper, we will describe our vision of how tangible interfaces will come to mirror the generalizability and versatility of digital environments, extending the interaction in ways that correspond to the intended input to improve usability, accuracy of input and the user experience. We will specifically look at these interfaces in the context of music production and creation.

a) An amateur studio setup

b) A screenshot from *Ardour*

Figure 1. An example of a small-scale, amateur studio setup with a mixer and a DAW (a) and Ardour, an example of a DAW with some plug-ins open (b).

Background
As with film, music and music production has inevitably become mostly digital. With increasing computational power comes the ability to use as many tools and components in a digital environment as can be fitted in an

average sized music studio (Figure 1). Furthermore, since there really is no objective answer to the question if analog is "better" than digital, many professionals within the audio and music production industry are including digital tools in their work process to utilize their convenience.

Almost all of the digital replicas used today are trying to look as "physical" as possible. This is achieved often by using a skeuomorphic interface (an interface that replicates the original design of the product). In this way it makes it easier for those who are familiar with the original product to use the digital representation. This only solves the graphical gap between the analog and digital worlds though, not the physical interaction aspect. When using a skeuomorphic interface it would be good to be able to interact with it as one would with the original device. When using an analog mixer for example you turn a knob by grabbing it and rotate your hand. To make the same action with the digital representation of the same mixer you click the knob with your mouse and drag it up or down to change the value.

The computer mouse can be used for most tasks on a computer, such as clicking on buttons, scrolling a page and making selections from a menu. Fitzmaurice and Buxton (1997) use the terms *space-multiplex* and *time-multiplex* to differ between different kinds of input. A space-multiplex input device would have controls where each control maps to one specific function, whereas a time-multiplex input device would have controls that serve different functions for different actions. [2] The keyboard is an example of a space-multiplex input device where in a text writing context each key creates one character. The computer mouse on the other hand is a time-multiplex input device because of its wide range of different functions.

Furthermore, Fitzmaurice and Buxton propose the usage of *Graspable User Interfaces* in contrast to regular UI's (regarding the usual computer interaction with a computer, screen, keyboard and mouse). A Graspable UI is not built from typical devices but rather consists of graspable functions, where each such function is attached to a virtual object on the computer. While the sound mixer table takes up a lot of space relative to its functions and the computer mouse cover very many different functions, the Graspable UI is put forward as a response to these interfaces and is argued to be better suited for users who work in a specific domain with a defined set of work tasks.

Previous work

Ullmer, Ishii and Jacob (2005) have conducted research about the usage of tokens with constraints. [6] The concept of tokens with constraints is about having regions or slots on a surface where specific tokens can be placed giving constraints to the movement of the tokens. Each token have certain regions where it can be put and mapped to a function on a computer, creating a physical interaction with a graphical user interface (GUI). When interacting with the tokens, the user is going through two phases: the first one being association, that is placing a token in the right region, and the second phase being manipulation, where tokens are used/manipulated in the constrained regions.

We have already touched upon the fact that computers can be used to visualize almost anything. Unlike in physical objects, all properties and states of a computer program are of the same type (namely digital bits) and equally accessible, meaning that any state of the program can, potentially, be visualized at any time. "Visibility of system status" is the first point in Nielsen's list of heuristics for usability evaluation [5] and contributes indeed to the user's ability to understand what a system is doing. Physical devices are limited in this regard since some properties are very hard to extract, but as we have pointed out that impediment does not exist in software. Combining the advantages of tangible user interfaces (TUI's) with the ability to display more information should therefore make for a more usable system.

A product trying to realize this combination is the reacTable, described by Jordá et. al in [3], a table-top musical instrument similar to analog synthesizers, controlled by a combined tangible and touch-based interface, and designed primarily for collaborative performances. Building on Jordá's previous work on visual feedback in digital music, the authors describe some important guidelines for tabletop musical instruments. For example, one goal was that "any shape, form, line or animation drawn by the visual synthesizer is strictly relevant and informational." They also wanted "to avoid any type of textual or numerical information, while banishing at the same time any decorative display"; basically subscribing in this case to the idea that less is more.

We think the reacTable is a successful attempt at incorporating the strengths of different types of interfaces in one system. Tangible bits are used for arranging parts of the interface and making precision adjustments in ways that the users are used to (e.g. knobs for turning), while the screen and touch is used for more complex input (displaying menus and selecting from these). The authors write that musical performances "often require the combination of intimate and sensitive control, with a more macro-structural and higher level control which is intermittently shared, transferred and recovered between the performer(s) and the machine". We feel the same is true to some extent for music production and similar contexts, and that the reacTable shows good way to make this combination.

THE MODULAR SURFACE

From our research and a number of reflective brainstorming sessions we have developed a concept describing a modular controller with the aim of supplementing the conventional interaction with the common computer interface. By having a set of physical controls the user can control specific

Figure 2. A CG concept illustration of the Modular Surface.

functions on the computer, thus creating an own modular controller. This *Modular Surface* would take the form of a flat screen placed on a table between you and the computer screen (Figure 2). On top of the surface the user can place the physical controls, which would then be given a physical constraint related to the interaction, i.e. having the controls stay in place and only allow certain movements. The physical controls should be as general as possible so that the user should not have to focus on choosing the correct controller, but rather focus on the interaction itself.

The concept was further discussed and evaluated in a music production context to get a better picture of how the workflow would look like[1]. Also, as we have seen, music production and creation involves many different interactions (though mostly turning knobs and moving sliders) while at the same time the workflow involves a more creative process rather than solving problems. Two main types of controllers are necessary for this context: a *rotary knob controller* and a *slider controller*. The knob controller's constraints would be formed by the need for it to stay in place while also allowing for the knob to be turned. Though a knob controller could also be used for controlling a slider, given the right constraints, we thought it more intuitive to have a dedicated controller for the sliders since the constraints can be made more obvious.

Regarding the size and appearance of the surface there is not much constraining the format or how big it should be. The smallest screen size could be argued to be around 9”, as big as the standard tablet, since the user should be able to put two hands on the screen and interact with two controls without problems. Also, knobs and sliders cannot be too small but to some extent need to mimic the physical controls seen today.

Interaction and workflow

The current workflow in Digital Audio Workstation (DAW) environments is limited by the amount of information that can be shown on a computer screen at one time. In general not many windows can be viewed simultaneously, resulting in windows quickly being hidden by other windows (Figure 1). Often when producing music in a digital environment you install different plug-ins to your audio software that replicates the analog tools that you would like to use, for example different kind of mixers and equalizers. During a production the user often uses a lot of different plug-ins and therefore needs to switch between these with the help of the mouse and keyboard to find the right one.

A way we thought of solving this issue of constantly switching between different windows is by having a layout where the screen is divided into a number of adjacent regions/windows of fixed size. Another area where this is used frequently is within 3D modeling software. In these software the user can switch between different layouts to be able to see the 3D model from up to four different angles at the same time. In our case the user will be able to choose a suitable layout to be able to fit the desirable plug-ins on the same screen. The surface can in other words be divided into a suitable grid where the users can select on which part of the screen a certain plug-in will appear (Figure 2).

This would make it easier to change plug-ins and solve the problem of superfluous interface components. From our interviews with amateur music producers, we found that they only use a fraction of all available buttons and knobs for any given plug-in, while all available setting controllers are visible (due often to their skeuomorphic nature and to give more control to the user). This is a great waste of screen space, meaning not all currently used plug-ins can be directly accessed at once, forcing the user to switch back and forth between them. By allowing the user to choose precisely which parts of the plug-ins to show on the modular surface he or she can change the focus from one to another more efficiently, without having to memorize controller bindings - a problem inherent in many MIDI controllers, where each knob or button can be mapped to an arbitrary function in the DAW, but the mappings are not made directly visible.

Prototypes

To further investigate our vision and to get a literal feel for how a TUI affects interaction, we developed some quick-and-dirty prototypes and a simple software. We made a slider out of LEGO and a touch-screen stylus, as well as two knobs by basically wrapping cylindrical objects in aluminum foil (Figure 3). The rotation of the knobs was captured by having two points under the knob touching the screen and calculating the angle of the line that ran through the points.

While the slider was a bit too primitive to be of any use, since it required using both hands for a successful interaction, the knobs could be used to rotate objects on a touch-screen as one would with a traditional knob. We evaluated them informally by using them on a tablet, comparing that to performing the same type of actions using only our fingers, and trying to extract some specific knowledge from the experience.

[1] For a concept video see http://vimeo.com/89420374.

a) Our knob prototype b) Our slider prototype

Figure 3. Our low fidelity prototypes.

DISCUSSION

A noticeable weakness with our prototypes was that they were not properly held in position. They tended to move around on the low-friction glass, putting extra strain on the user. That they stay in place is important to provide a pleasant user experience, but we still found that they were easier to use and more accurate for their specific purpose than a standard touch interface. After surveying some popular software, we found that the interaction method for turning virtual knobs in touch-based interfaces is traditionally either sliding the finger up and down (or right and left) on the knob, as if it was a slider, or "grabbing" the surface with two fingers (i.e. a pinching movement) and rotating the hand. The former dissonates with the affordances of a knob, and the latter requires an unnatural, unergonomic movement.

We also found that an important aspect of well-designed physical input modalities is that they constrain the user to only producing the relevant form of input. On a touch-screen, the fingers can and do move freely across the surface regardless of which action was the initially intended one. If you start turning a virtual knob with two fingers, they will probably slide away a bit from the knob on the screen, and the distance between them will change; when moving a slider, the finger will not move in a perfectly straight line. On the other hand, if the user's hand is locked onto a physical device the user will be constrained in space and functionality to that device - there will be less need for the user to wonder if the knob is still being turned or care about where the hand is currently at. We are confident though that the prototypes gave a fair depiction of the interaction and that the low fidelity prototypes were enough to represent the concept.

The conclusions drawn from our short evaluation, that TUI's are superior for certain types of interaction, is supported by other research [4] and indicates a practical use of independent, movable input devices to interact with skeuomorphic components in the intended way. However, the interaction might include both the use of common finger movements as well as physical objects and not one exclusively. The tools used for interaction in virtual environments need to be limited in their functions and so there is a tradeoff between a tool's generality and its constraints [1]. The Modular Surface could therefore benefit from letting the users create their own controllers as 3D printing is becoming more advanced and available to the ordinary user.

CONCLUSION

In 2013, one third of Americans owned a tablet [7]. The current trends indicate that they will become even more popular, perhaps as ubiquitous as laptops. They will become more integrated into common work tasks and a natural element in people's everyday life. Some argue that the intuitiveness of interacting with them is responsible for part of their success; touch-interfaces indeed bring users closer to a natural interaction with skeuomorphic interfaces, but there is still an unbridged gap. As the novelty wears off (and it has started to) people will realize that they are not holding perfection in their hands. There will be a demand for even more sophisticated and natural interaction, and the vision of the Modular Surface might be an answer to just that. We have shown, based on our own testing and previous research, that tangible interfaces provide extended usability to some applications and that it would be both reasonable and technologically possible to see them become reality within the next ten years.

ACKNOWLEDGEMENTS

We wish to thank Anders Lundström for sharing his thoughts about the current state of music studios and his vision of the future within the area.

REFERENCES

1. Doug A. Bowman, L.F.H. User Interface Constraints for Immersive Virtual Environment Applications. .
2. Fitzmaurice, G.W. and Buxton, W. An empirical evaluation of graspable user interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97*, ACM Press (1997), 43–50.
3. Jordà, S., Geiger, G., Alonso, M., and Kaltenbrunner, M. The reacTable. *Proceedings of the 1st international conference on Tangible and embedded interaction - TEI '07*, ACM Press (2007), 139.
4. Lucchi, A., Jermann, P., Zufferey, G., and Dillenbourg, P. An empirical evaluation of touch and tangible interfaces for tabletop displays. *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction - TEI '10*, ACM Press (2010), 177.
5. Nielsen, J. Ten usability heuristics. (2005).
6. Ullmer, B., Ishii, H., and Jacob, R.J.K. Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction 12*, 1 (2005), 81–118.
7. Zickuhr, K. Tablet Ownership 2013. Pew Research Center's Internet & American Life Project. http://www.pewinternet.org/2013/06/10/tablet-ownership-2013/#fn-63-1.