Towards a Humane Graphical User Interface for Live Electronic Music

Jamie Bullock
Birmingham Conservatoire
Birmingham City University
UK
james.bullock@bcu.ac.uk

Lamberto Coccioli
Birmingham Conservatoire
Birmingham City University
UK
lamberto.coccioli@bcu.ac.uk

Abstract

In this paper we describe findings related to user interface requirements for live electronic music arising from research conducted as part of the first three-year phase of the EU-funded Integra project. A survey of existing graphical tools for live electronics is presented along with observations about current usage patterns and cultural trends. From these data a set of requirements are gathered, and resulting design questions are discussed. A number of graphical user interface (GUI) prototypes developed during the Integra project initial phase are described and conclusions drawn about their design and implementation. Finally a proposal is made for a new GUI that takes into account the findings of our research.

Keywords: Integra, User Interface, Usability, Design, Live Electronics, Music Technology

1. Introduction

According to Raskin[1] in order to be humane, interfaces should meet the following criteria:

- **modeless**: user actions should have the same effect regardless of the application's state.
- **monotonous**: there should only be one way to accomplish any given task in the UI.
- **visible**: the 'right' features of an application should be visible at any given time and users shouldn't be forced to memorise that a feature exists.
- **affordance**: UI functions and operation should be obvious to most people in the culture for which it was intended and make use of already learned human skills.

A humane interface is a *usable* interface designed to be sympathetic to the way humans instinctively interact with computers, and not necessarily designed around the structure of computer hardware and operating systems. Interface usability is closely coupled with the notion of the humane interface and is often defined in the broader context of system acceptability. Figure 1 shows a diagrammatic representation of an early acceptability graph devised by Nielsen[4]. Consideration of complete system acceptability is beyond the scope of this paper, instead, we will focus on the usability branch. Usability is defined by Nielsen as having the following five attributes:

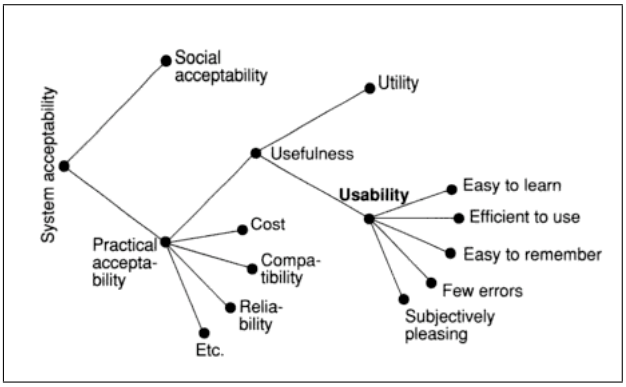

Figure 1. Graph of qualities for user interface acceptability.[4]

- **Learnability**: The system should be easy to learn so that the user can rapidly start getting some work done with the system.
- **Efficiency**: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.
- **Memorability**: The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.
- **Errors**: The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.
- **Satisfaction**: The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

Nielsen argues that usability isn't just a broad, multi-faceted concept, but something that can be quantified and qualitatively measured. In this paper we will evaluate software currently used for live electronic music in the context of humane and usable interface considerations. In particular we will discuss the usability evaluation processes employed as part of the Integra project prototyping phase.

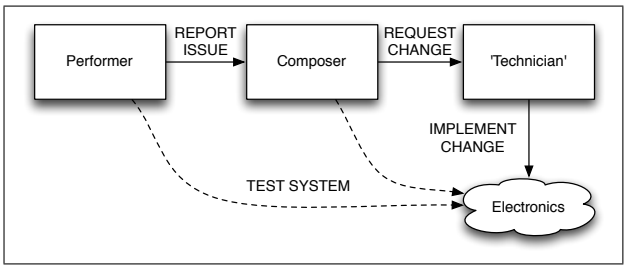

Figure 2. Diagram illustrating observed dependence between performer, composer and 'technician'.

2. Research process

Integra[1] is a project led by Birmingham Conservatoire in the UK and supported by the Culture programme of the European Union. Integra brings together new music ensembles and research centres to collaborate on a wide range of artistic and scientific activities centred around live electronic music. The initial 3-year phase of the project is now complete. During this period eleven European composers were commissioned to write works for five professional ensembles, all partners in the project. As part of the composition process the composers were each paired with one of the eight research centres in order to develop the 'live electronics' of their compositions.

The process worked very well and resulted in successful performances of all works involved, including premieres at the first Integra Festival held at Birmingham Conservatoire (UK) in June 2008. However it also highlighted some important issues relating to the Integra project's scientific objectives, namely the development of a new environment for live electronics. The following observations are pertinent:

- The tool chosen by the majority of composers for the composition and performance of the live electronics was Max/MSP by Cycling74[2].
- The relationship between composers and research centres was primarily collaborative, but in many cases composers were highly dependent on the research centres for technical assistance.

We observed in some cases, that the traditional relationships of composer and musical assistant emerged, with the musical assistants interpreting, facilitating or simply executing the composer's artistic wishes on their behalf. Composers often employed other tools, for example Protools, Csound, VST plugins as auxiliary applications in the creative process. These were used for preparatory work such as pre-processing source material prior to incorporating into a 'live' context.

Since most of the compositions commissioned by Integra involved interaction between (acoustic instrumental) performers and the electronics, we were also able to observe the

[1] www.integralive.org
[2] www.cycling74.com

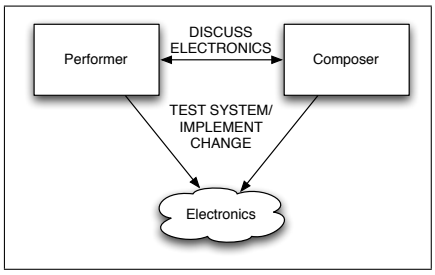

Figure 3. Diagram illustrating ideal scenario where performer and composer interact with the software on equal terms without technical assistance.

reactions of the performers to the live electronics systems devised by the composers and researchers. Reactions were mostly positive, with performers showing a sense of discovery and excitement in 'playing' with the electronics. However, in some cases this was accompanied by a feeling of uncertainty, especially when the electronics seemed not to be working as expected. We also noticed a secondary level of dependence for technical assistance if changes were required in the electronics for musical reasons. This 'distancing' of the performers from the electronics is shown in Figure 2. We believe that if the software systems used for the live electronics were sufficiently humane and usable then the performer and composer would become empowered, and less dependent on the 'technician'. This would lead to the relationship shown in Figure 3.

2.1. Problems identified

In the course of our research we have identified the following key problems:

1. Musicians are discouraged from working with live electronics due to difficulties using the software.
2. Musicians will often collaborate with a technical expert if they intend to use live electronics in their work.
3. There are no purpose-built applications for live electronic music; general purpose tools and programming languages are used instead.

By surveying a range of software available to musicians, we established that item 3 in the above list is one of the causes of items 1 and 2. However, despite the lack of purpose-built systems for live electronics, there are a number of projects in progress, which aim to address usability issues in general purpose 'multimedia processing' software. Some of these will be discussed in the following section.

3. Existing systems

This review is not intended to be exhaustive, but rather to give a range of contrasting approaches to the problem of user interface design in this field.

3.1. Jamoma

Jamoma consists of a number of interoperable modules built in the Max/MSP environment, and a set of guidelines for module construction[2]. Jamoma provides a number of usability improvements compared to Max/MSP including individual 'pre-built' module GUIs, human-readable parameter addressing, HTML documentation and a flexible mechanism for control over composite parameters[3]. The primary advantage of using Jamoma over Max/MSP is the nature of the modules provided, which are generally 'higher level' from a systems architecture perspective, and encapsulate functionality that is specifically tailored to the needs of interactive music and video. The Jamoma modules essentially provide an artist's toolkit without presenting the user with too many choices or requiring a significant amount of specialist knowledge.

3.2. ixi

ixi works on a different principle to Jamoma, treating the graphical user interface not as a tool, but as a musical instrument[5]. Each ixi GUI provides a unique user experience, resembling a graphical toy or game to be played by the performer, artist or casual experimenter. From a usability perspective ixi GUIs invite an exploratory approach to interaction. They are usually graphically simple and focus on one idea such as the interconnection of certain lines and shapes for a given musical task. Such tasks include granular synthesis, algorithmic note generation or beat slicing. The strength of ixi lies in the expressive but often non-obvious relationship between the graphical interaction and the resulting change in the sound output. ixiQuarks is a recent development of the ixi concept providing a complete system of interoperable tools for audio processing and synthesis[6]. Some of the ixiQuarks components have deterministic user interfaces where the UI controls and DSP are closely coupled, whereas tools in the 'instruments' category have an abstract game-like quality akin to the early ixi releases.

3.3. Bidule

Similar in emphasis to Jamoma, Bidule is a commercial software package consisting of modules (called bidules), which can be connected up in a patcher environment. Like Jamoma, each module has its own UI. As a modular patching environment Bidule has many usability advantages over Max/MSP:

- Adding modules to the patcher is drag n drop (no typing).
- Textual labels are used instead of (non-obvious) icons.
- Modeless interface (no 'edit' or 'performance' mode).
- New patches contain sensible default objects (inputs, mixer, output) to get the user started.
- All parameters are OSC (Open Sound Control) addressable by default.

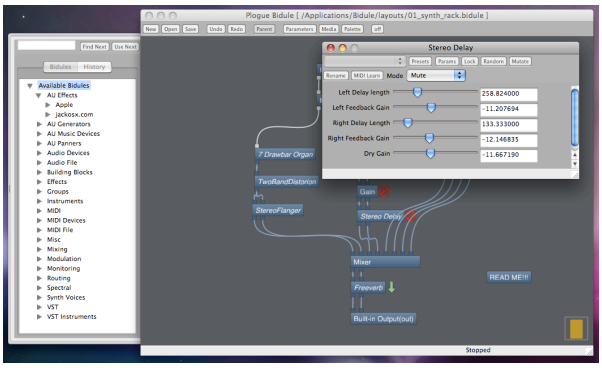

Figure 4. Bidule with individual Delay module GUI shown.

- Colour-coded connections for audio, MIDI and spectral data types.
- Massively simpler preferences dialogue.

Bidule comes with a collection of more than one hundred musically usable modules and can run as a VST plugin. A screenshot is shown in Figure 4.

4. Towards a GUI for live electronics

The Integra project seeks to develop a new graphical user interface drawing on best practices from the software discussed in section 3, whilst taking account of the usability criteria highlighted in section 1. In the following sections we outline our proposal for the specification of a new GUI for this purpose.

4.1. User interface model

We started our research and design process by identifying the possible personas, scenarios and prototypes for the interface using a model proposed by Tidwell[7]. This is shown in Figure 5. The diagram illustrates the large number of stakeholders for a specialised interface for live electronics, and highlights the inadequacy of existing environments regarding this task. For example the behavioural prototypes, 'composition', 'rehearsal', 'performance', and 'development' are mostly not represented in the interfaces explored in section 3. Max/MSP is the exception, having a 'performance' mode that allows the visual layout of certain components to be independent from the logical layout of the patch. Since composition, rehearsal and performance all require different modes of thought at different stages of completion of the musical work, it seems logical that the UI should reflect this.

4.2. Interface prototypes

A number of user interface prototypes have been developed reflecting this model. We started with a small number of GUI mockups showing an initial layout and workflow. These mockups and supporting documentation are available on the

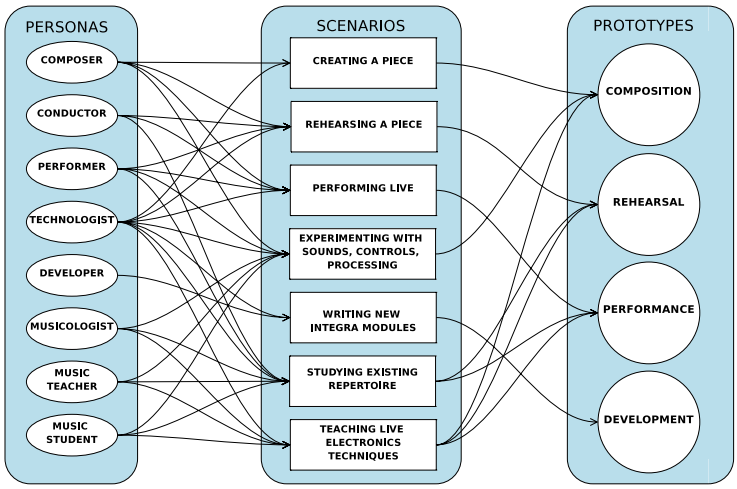

Figure 5. Possible user interface scenarios in software for live electronics.

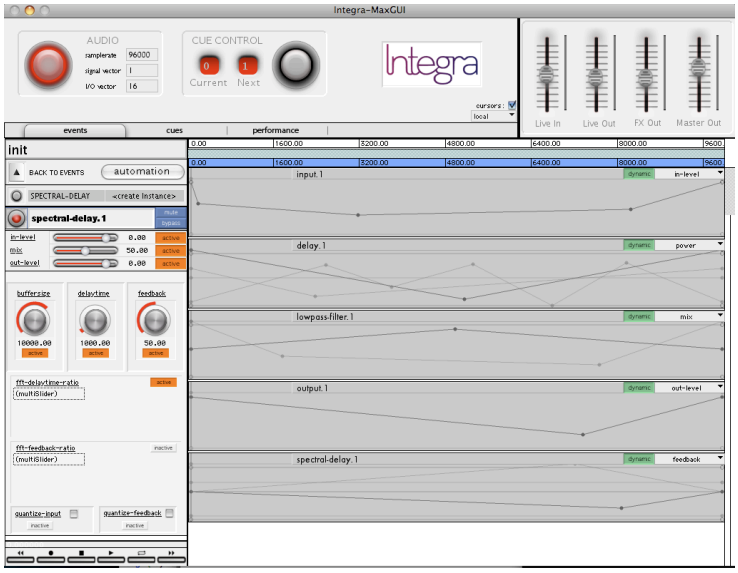

Figure 6. Integra prototype GUI developed in Max/MSP by CIRMMT, McGill University, Montreal.

project wiki[3]. UI prototypes have subsequently been developed in Java, Javascript/XUL, and Max/MSP/Javascript. Multi-platform development has been chosen in the early stages of the project in order to simultaneously develop UI ideas and evaluate development environments for the final product. The most evolved prototype was developed by Integra project partner CIRMMT and is shown in Figure 6.

This GUI is functionally compatible with libIntegra version 0.3.1[8] and is capable of loading, connecting and sending control data to Integra modules hosted in a supported DSP environment.

5. Future Work

Now that we have several GUI prototypes and a body of research in performer/composer HCI to inform our work, our next task is to implement the user interface in a more robust and full-featured manner taking into account usability deficiencies found in the GUI prototypes. We have obtained funding to achieve this and plan to make a public release in 2010. We intend to capitalise on our position as Conservatoire-based researchers to employ user testing early in the development process, through to release.

6. Conclusions

According to Raskin "An interface is humane if it is responsive to human needs and considerate of human frailties"[1]. The Integra project seeks to create a new graphical user interface specifically for use in live electronic music, which meets these demands. We have discussed the findings from our work during the initial 3-year period of the Integra project, including evidence from HCI studies in the composition and performance processes resulting from eleven commissions of new works. We have identified a number of existing user interfaces, which represent the state-of-the-art in design for live audio and video processing in an electronic music context and conclusions have been drawn about the most salient features of this software in relation to new interface design. Finally we have described a new conceptual model for a live electronics interface leading to a number of prototype systems.

References

[1] J. Raskin, *The Humane Interface: New Directions for Designing Interactive Systems*, Addison-Wesley Professional, 2000.

[2] T. Place and T. Lossius "Jamoma: A modular standard for structuring patches in Max." in *Proceedings of the International Computer Music Conference (ICMC)*, New Orleans, 2006.

[3] T. Place, T. Lossius, A. R. Jensenius and N. Peters " Flexible control of composite parameters in Max/MSP" in *Proceedings of the International Computer Music Conference (ICMC)*, Belfast, 2008.

[4] J. Nielsen, *Usability Engineering*, Academic Press, 1993.

[5] T. Magnusson. "ixi software: The Interface as Instrument" in *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, Vancouver, 2005.

[6] T. Magnusson. "The ixiQuarks: Merging Code and GUI in One Creative Space" in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, 2007.

[7] J. Tidwell. *Designing Interfaces*, O'Reilly, 2005.

[8] J. Bullock and H. Frisk "LibIntegra: A System for Software-Independent Multimedia Module Description and Storage" *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, 2007.

[3] http://wiki.integralive.org/integra2_design