EXPLORING THE DESIGN SPACE: PROTOTYPING “THE THROAT V3” FOR THE ELEPHANT MAN OPERA

Ludvig Elblaus
KTH Royal Institute of Technology
elblaus@kth.se

Kjetil Falkenberg Hansen
KTH Royal Institute of Technology
kjetil@kth.se

Carl Unander-Scharin
University College of Opera
carl.unander-scharin@telia.com

ABSTRACT

Developing new technology for artistic practice requires other methods than classical problem solving. Some of the challenges involved in the development of new musical instruments have affinities to the realm of *wicked problems*. Wicked problems are hard to define and have many different solutions that are good or bad (not true or false). The body of possible solutions to a wicked problem can be called a *design space* and exploring that space must be the objective of a design process.

In this paper we present effective methods of iterative design and participatory design that we have used in a project developed in collaboration between the Royal Institute of Technology (KTH) and the University College of Opera, both in Stockholm. The methods are outlined, and examples are given of how they have been applied in specific situations.

The focus lies on prototyping and evaluation with user participation. By creating and acting out scenarios with the user, and thus asking the questions through a prototype and receiving the answers through practice and exploration, we removed the bottleneck represented by language and allowed communication beyond verbalizing. Doing this, even so-called tacit knowledge could be activated and brought into the development process.

1. INTRODUCTION

It is common practice when working with new musical instruments, new media art or other artistic practices that rely heavily on new technology, to work interdisciplinary. Engineers, technicians or instrument makers work together with artists, composers or musicians towards a common goal. We will here denote these groups as *developer* and *artist* respectively, well aware that this is an oversimplification.

The developers often know the technology well but have less insight into the intended context and usage than the initializing artist, while the artist may have less knowledge of the technology or other priorities. Many projects are therefore close collaborations between these professionals (and commonly, the cross-fertilization is strong, thus smearing out the borders between the roles), as have been reported in for instance the Sound and Music Conference [1] and the New Interfaces for Musical Expression conference proceedings [2].

Figure 1. Portrait of Joseph Merrick (1862–1890), also known as *The Elephant Man*.

Communication during the development process is very important. The more of the artist’s relevant information that can be available to the developer, the higher the probability is of the project being successful. This knowledge is however not always easy to communicate. It can be embedded in practice, so-called tacit knowledge that can be hard to verbalize. The communication can also be hindered by questions that the developer fails to ask and details the artist fails to recognize as being important [3].

When dealing with technology intended to create or enhance experiences both from a performer and an audience perspective other practices than traditional hardware–software development models can be used to open the necessary channels of communication. During recent work with prototype development for a new musical instrument, ideas from the fields of interaction design and participatory design were used to bridge the artistic and technological divide.

In this article these ideas will be briefly outlined as they are theorized within their respective fields and their application described in three specific cases. Furthermore, the divide between the artistic and the technological is bridged by a hands-on approach using prototyping and involvement of both artist and developer in a team. Instead of only using the artist's expertise as a starting point and to evaluate the result of the process, the artist has been immersed in the development process so all available resources could be exploited in the project.

1.1 The project

In this project, Elblaus was approached by Unander-Scharin to develop a gesture controlled signal processing device for stage use. The project, called "The Throat v3" is still ongoing by the time of writing. The Throat v3 is to be used in an opera entitled "The Elephant Man", currently being composed by Unander-Scharin. Although the roles as artist and developer were conditioned from the start, the developer has documented artistic experiences, and the artist has documented experiences in development. Therefore, interchange of experiences was well supported and the use of participatory design patterns was a natural direction in the development.

Frederick Treves notes in his autobiography [4] regarding Joseph Merrick, popularly known as *The Elephant Man*, that "the fact that his face was incapable of expression", and "his attitude that of one whose mind was void of all emotions" (see Figure 1). When conceptualizing The Throat v3, these aspects were thoroughly considered. A microphone was used to capture the singers' portrayal of the limited vocal sounds that were possibly produceable by Joseph Merrick, due to his severe physical disability. The smaller components of speech and singing, which are normally inaudible in applied operatic (italianate) technique, could be utilized to create soundscapes and accompaniment for arias.

A suggested term for this practice could be "deformed vocal technique"—as opposed to "extended vocal technique". Keywords suggested by the artist to the developer, when designing the sound processing modules were: mucus, inflammation, coughing etc.

The prototype uses a computer with an audio interface, a microphone, an Arduino microcontroller [5], and pressure and flex sensors. The sensors are varistors so some simple voltage divider circuitry is needed to let the microcontroller read the varying resistance corresponding to the measured pressure or bending of the sensors.

The software, written in the SuperCollider language [6], is a modular environment that offers a wide assortment of processing types that can be modified, combined and collected into scenes which in turn can be arranged into sequences. This way longer structures of scenes can be prepared for a performance so that a performer can focus on stepping through the structure and modify parameters expressively in each scene.

A system of morph groups is available, where one sensor can be mapped to any number of signal processing parameters on a per scene basis, allowing both *one-to-one* and *one-to-many* mappings to be constructed [7]. Thus, scenes can both sound very differently and also offer individual types of interaction.

2. THEORY

2.1 Wicked problems

Horst Rittel and Melvin Webber [8] describe what they call *wicked problems* as problems that defy the standard problem solving method. Rittel and Webber were concerned with complex large scale problems such as public policy, but the reasoning behind wicked problems also motivates why some tasks in general benefit more from a design methodology than from problem solving.

Rittel and Webber formulated a set of distinguishing properties that showed how a wicked problem differs from what they call *tame* or *benign* ones. The theory is quite extensive, but in short, there are some properties explaining why design is not problem solving. In general wicked problems

- have no definitive formulation,
- have no stopping rule,
- have good-or-bad (not true-or-false) solutions,
- have no ultimate tests for solutions,
- have no finite number of potential solutions, and no defined set of permissible operations.

The last distinguishing property is very important for the process of dealing with wicked problems. Firstly, it states that the set of potential solutions is not known explicitly, which means that we can never try all the possible solutions to find the best one. Secondly, it states that each solution might contain any operation or element, and that we can never go through all the combinations of a defined set of operations since that set is not known.

2.2 The Design Space

The design space is the sum of all possible solutions to a design problem. The question is how to approach this space and how the design process should be structured with this in mind to produce good results.

The design space will never be fully known or fully understood but at the same time knowledge about the design space is needed to evaluate the solutions that are discovered during the design process. Therefore the goal of the design process can never be to fully define the design space but to get as much knowledge of it as possible so as to find the best possible solution given the constraints of the process itself, e.g. budget and time.

Accepting the process implied by the many different solutions in the design space means that the design process will be more of a gradually narrowing search than a journey to a predefined goal. It is clear that several intermediary solutions must be explored before the process is finished and that these solutions must be evaluated in some way. Every solution that the development process produce will chart a small subset of the design space so it is by putting forward solution suggestions and testing them that information about the design space is discovered.

Notice that the goal from the beginning is to find *several* solutions to the problem and not to first try figure out the optimal solution. For such work, prototypes will be used to explore and map the design space, and each prototype will generate knowledge about the design space. By making prototypes and mock-ups of the proposed solutions, they can be evaluated and discussed from an interaction perspective and not just a theoretical one.

3. METHOD

3.1 Participatory Design

Knowledge of the product's intended user and the context can be acquired in many different ways. Combining methods that include observational field studies, interviews with users or other stakeholders, statistics, and surveys, lead to a better understanding of user needs and practices [9]. These methods can be problematic as they rely on intermediary observers or mediums like language and statistics, necessitating interpretation. Personal background, preformed views, and prejudices color the observers interpretation [10]. There can also be a discrepancy between what the respondents think they do, and what they actually do [11]. Many experiences are also very hard to accurately describe across modalities. For the subtle experiences of musical instruments, this is very much true.

Another approach is to invite the user to partake in the development process to directly access user response and feed that back into the development. Directly involving the user in the process can be challenging, but it will provide an abundance of information that is relevant since the information springs from the interaction between user and process [12].

With user participation, the information one gets is not filtered by the questions one ask, i.e. answers to questions that are never verbalized or thought to be relevant by the developers can still be given. Similarly, answers that can not be verbalized, so-called tacit knowledge, can still be used in the development process by letting the user participate and show practices, act out scenarios or by other means communicate what is hard or impossible to reduce to words [3].

Leman [13] uses the term *embodied cognition* to describe how the body is intimately engaged both musical performance and perception. Getting participants involved in the design process in a physical way is therefore very important to get an understanding of all the mechanisms involved in a musical context. These activities must have a solid support in the design philosophy used.

3.2 Iterative Design

Iterative design is a prototype-driven way of structuring the development process that is well suited to the needs of participatory design, as well as exploration of the design space.

It is a well established development method, see for instance Gould and Lewis [14] who focus on the designer's perspective or Nielsen [15] who discusses usability from a practical economic perspective.

Iterative design uses a cyclical work flow that for each cycle further refines the design. This model is a good match

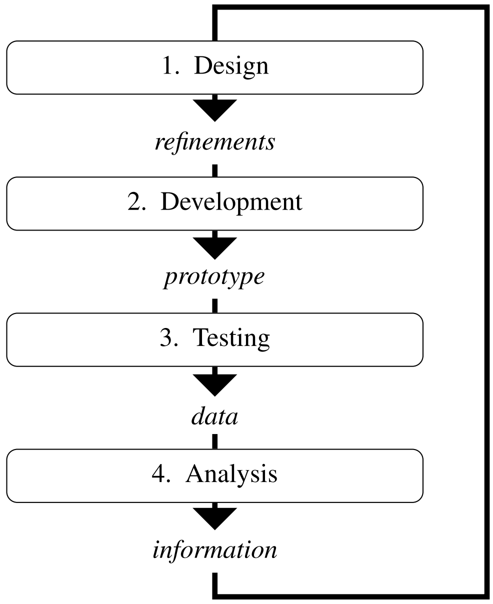

Figure 2. The cyclical flow of the iterative design process.

for participatory design, where maximizing the user involvement and quick reactions to user input are crucial. As soon as a first sketch or idea is formed the cyclical iteration can begin.

At its simplest, a design cycle contains four stages, as shown in Figure 2. First, a design stage where refinements to the design, based on available information, are proposed.

Second is a development stage where the design changes are carried out, which is the only stage where user involvement can be unnecessary. If hi-fi prototypes are to be constructed or if the product is nearing completion, user may have very little to contribute.

Third is a testing stage where it is vital to involve the users, to learn from the prototype. All cycles involve testing, a central requirement of participatory design.

Fourth is an analysis or evaluation stage where the data collected during testing is processed and refined into usable information. This is an important step as the experiences from the testing stage can be ambiguous and difficult to interpret. The information produced in this stage is used as input to the next cycle, as basis for the decisions taken in the next cycle's design stage.

3.3 Prototyping

When working with technology that is used in an artistic context developers are often in need of information on diffuse, hard to measure qualities regarding user experience. Whether something works or not in a technical sense is easy to determine but how the user experiences the interaction is a very different matter [16].

Language might be a bottleneck in communicating these kinds of questions and their corresponding answers. Simply asking someone how they would act in a certain situation might yield very different answers compared to actually observing them experience that situation and act in it [11]. This is why asking the question through a prototype and receiving the answer through practice and exploration can be so valuable to a design process.

A prototype from a user interaction perspective is anything that can provide information on an interaction scenario. As Westerlund has shown [3], the prototype does not have to be an advanced piece of technology, but it must be able to successfully put the user in the desired scenario. A prototype does not have to be functioning on its own, it does not even have to be built with the same material as the intended product of the design process, as long as it answers a question or provides information through real or simulated interaction (for instance, a Wizard of Oz experiment).

Another aspect to consider when prototyping is to what extent the prototype is to evolve into the following generation and to what extent it will be thrown away. This is a dichotomy known as Evolutionary / Throwaway prototyping. When creating artworks, it is common practice to let the artifact evolve from prototype to a finished work of art. However, in iterative design, the use of a prototype that will be abandoned is common.

4. USE IN DEVELOPMENT OF MUSICAL INSTRUMENTS

Interaction design has emerged to fill a function where earlier methods have been less effective. The measurable and quantifiable aspects of design and development such as ergonomics and efficiency are all captured in more traditional development processes [17].

The field of new musical instruments and other artistic uses of technology are completely saturated with these very types of problems and questions of soft values and sometimes even subliminal experiences [18]. How does the performer experience her instrument? How is the instrument perceived by the audience? Is the instrument experienced as expressive and if not why?

Using the well researched tools of interaction design and other similar models can not only help to find these questions but also provide a method to learn from them and work with them. The idea of the design space is a good metaphor for projects that are not searching for an ideal solution to a well defined problem, but rather exploring a myriad of possible designs that can only be evaluated aesthetically.

4.1 Limits of user participation

With technology intended for an artistic context, one might assume that getting the technology to work and to make it a work of art is two separate tasks. This leads to the assumption that the artist primarily has artistic concerns, and the developer mainly technical. In reality, the roles are rarely so explicit.

A definition of purely technical aspects is to view them as black boxes. The input and output from the black box might have aesthetic ramifications but the inside workings are obscured from the user and indeed the system as a whole. Given that the same input leads to the same output, the mechanics inside are bereft of artistic relevance for the system.

Case 1: The Experimental Environment

Participation	Incorporation of artistic vision
Prototyping	Non-verbal communication in the form of concrete sound exploration
Software-modularity	Reduction of developer bias

Case 2: Lo-fi sensor workshop

Participation	Activation of tacit knowledge of practice
Low-fidelity prototyping	Unhindered exploration
Workshop	Open form that supports experimentation

Case 3: Concert test

Wizard of Oz-prototyping	Testing without fully functional prototype
Context exploration	Simulation of the intended final context
Participation	Use of the composer's stage experience

Case 4: External artist

Studio test with external artist	Testing fully working prototype
Simplified context	Elimination of audience interaction
Externalized cognition	To explore the recontextualization of The Throat

Table 1. Methods and goals for each of the case studies.

Elements that fit that description can be handled with regular problem solving, user participation is not necessary and aesthetics can be disregarded. These black boxes are the only elements that the developer can design alone, leaving out the artist. It can be mutually beneficial to conserve responsibility for the technical details to the most knowledgeable party, especially when for instance the artist could direct efforts where it is more needed.

5. THE THROAT V3

For *The Throat v3* participatory design was used extensively. This was helped by the fact that the developer in this case had an artistic background and the artist had already developed early versions of The Throat.

However, the developer had however no previous experience of working with opera or singing voice, while the artist was unfamiliar with SuperCollider and could therefore not partake in the source code. Thus, the situation re-

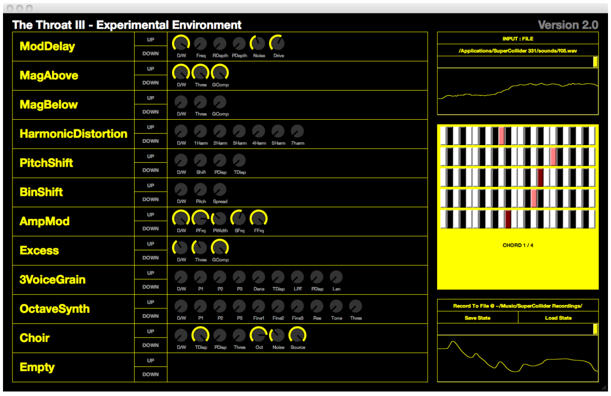

Figure 3. The Throat Experimental Environment is a modular signal processor written in the SuperCollider programming language. It offers a wide array of effects that can be used in real time.

sembled that of a traditional divide between project roles, and although in practice the participants were experienced in both roles, many details still needed to be shared and communicated. The design methods used needed to supply the tools for this communication to take place.

In the following, we outline how applied design methodology corresponding to stages in the project exemplifies the use of participatory design method, transmission of tacit knowledge through workshops, and partially working prototypes used with a Wizard of Oz approach. A summary of the methods and goals used in each case can be seen in Table 1.

5.1 Case 1: The Experimental Environment

When the foundations of the signal processing in the prototype were to be laid out, an approach was needed to ensure that no design decisions with artistic ramifications were taken without the artist's involvement. For this purpose an experimental environment was constructed. It contained modular building blocks of signal processing. The processing performed could be controlled by a set of parameters unique to each module. The signal chain could use any selection of modules in any order. The graphical user interface is show in Figure 3.

While it fell on the developer to code the actual software, the preliminary selection of processing types could be agreed on through discussions with the artist. Here, language was no bottleneck since well-known unambiguous terms exist for operations in signal processing (such as *ring modulation* or *low pass filter*).

Different kinds of timbres achievable through these techniques lack a well defined terminology. Even within a specific vocal tradition, terminology that transcends the technical aspects of vocal pedagogy can be ambiguous [19]. Precisely describing the desired timbres of singing through complex, layered signal processing would have led to unusable specifications. Instead, to give some guidance, a scenario was written by the artist to deepen the developers' understanding of the intended use of the prototype.

In working with the environment all or a subset of the available modules were connected in series. An audio signal was sent through this chain of modules and the result could be listened to in realtime. The incoming audio signal could be read from an audio file or taken from an audio interface.

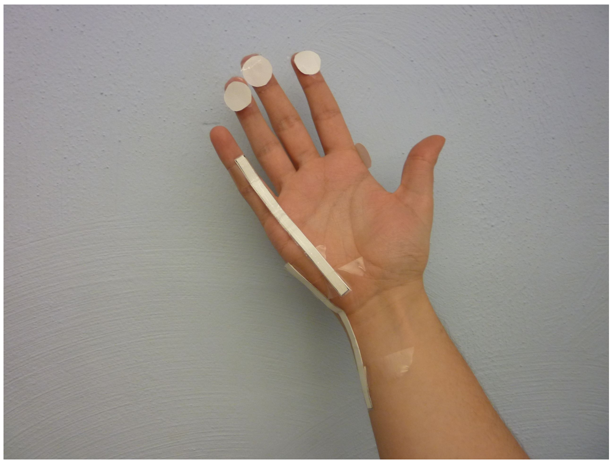

Figure 4. Paper replicas of sensors used in a workshop. The positions of the replicas were found through experimentation, unhindered by risk of damage to the actual sensors.

The parameters of the signal processing of each module were available for manipulation. The mapping of the parameters and the choice of which parameters to offer were done to offer too much rather than too little. This way, the developer imposed as little artistic influence as possible, while leaving the artist room for creativity.

The artist explored this environment and saved noteworthy configurations, whether they were aesthetically pleasing or simply undesirable. It was not required to test the finer details of the signal processing together at that stage. The saved configurations were also practical for expressing aesthetic preference without the limitations and interpretations of language. It was simply easier to show than to tell.

The modular, open-ended concept worked so well that it was carried over into the later prototypes. It was a request from the artist to keep all parameters available from the experimental environment—to be able to return to all possible combinations of processing that had been used during the development.

5.2 Case 2: Lo-fi sensor workshop

One of the most important design decisions was how to position, attach and interpret the gesture-reading sensors placed on the hand of the performer. After discussions on different placements the need for testing the ideas in practice arose. Using the actual sensors would have been restricting and potentially expensive, as care would have to be taken not to damage or stress the sensors in any way.

To be able to freely explore the possibilities, paper replicas of the sensors were constructed. The replicas were made to have the same weight and bending characteristics as the actual sensors. The replicas attached to the hand can be seen in Figure 4.

Soon, a wealth of the artist's tacit knowledge of stage work was uncovered, catalyzed by the experience of the different sensor placements. Different placements affected the hand's stiffness in different ways, having a significant effect on the artist's ability to act on stage, that had not been anticipated.

In the conceptual work with the sensor placements focus had been on function and having as much control as possible. Considering that the performer in the opera would wear an elaborate mask to resemble the elephant man, including an abnormally large arm that would hide the sensors, the strain of the sensors seemed small in comparison. This mistake should possibly, in hindsight, have been noticed and avoided, but in the context surrounding the discussions there were too many other factors that received attention.

Here prototyping worked as a safeguard directing focus to an aspect that might otherwise have been overlooked. It turned out to be relevant to the development process and it surfaced naturally through a hands-on workshop with lo-fi prototypes. Simple discussions had not been able to come to the same important conclusions, and neither would a test with the developer using paper replicas have.

When preparing for the workshop, the developer had devised a set of hand gestures that could be used with the sensors. It became clear however that some hand positions that were trivial to the developer were uncomfortable for the artist, and vice versa. This is naturally especially important if the design is addressing more than one performer.

5.3 Case 3: Concert test

At one point in the development process an opportunity to perform an informal concert was presented. While the prototype at that point was not fully functional there were still many things that could be learned by testing it live, so the work focused momentarily on preparing a performance-ready version.

An attempt was made to integrate all the parts of the system, but the result, while functional, did not seem stable enough to meet the demands of a performance however informal. The concern was the connection between hardware and software that failed at one point and the time that was left before the concert did not allow for debugging to find the problem. Not knowing the cause of the failure, if it would happen again and if so, how often and under what circumstances, the risk of the prototype failing mid-performance could not be assessed, and so a fully-functional prototype was unattainable.

Next, a decision was made to only test a part of the system and to simulate the rest of the functionality in a Wizard of Oz approach. The software was used to process the artist's voice but the hardware control was substituted by an additional person on stage. The artist's hand gestures were not represented by sensor readings, but simulated by the person sitting directly by the computer running the software, watching for predefined hand cues. For this purpose a simple interface for manual control was added to the software and used for the performance.

Figure 5. A singer performing an aria using gesturally controlled signal processing in the final test in the prototyping process of designing the Throat V3.

Much was learned from that test and canceling the performance because of prototype failure or instability would have been a missed opportunity.

5.4 Case 4: Test with external artist

The development process was concluded with a test where a singer, external to the process, tested the prototype in a studio environment. Material from the upcoming opera was used, and the test therefore provided an understanding of how the prototype would function in the artistic process in which it was to be used. The artist could explore the process of preparing signal processing suited to the set material, testing both the usability of the software as well as the possible artistic expressions that could be achieved.

The singer had little problem moving while wearing the prototype and immediately began incorporating the control-gestures in larger gestures, masking their true function and making them a part of a larger stage presence. The singer, using the prototype can be seen in Figure 5. This provided important information about the prototypes ability for interaction and that the interaction itself could be integrated into operatic stage practice.

The singer performing with the prototype can be seen in Figure 5. The test was also filmed and an excerpt is available online. [1]

6. CONCLUSIONS

Learning from the process of developing The Throat v3, it is clear that other tools than problem solving are needed to work with the complexities of constructing prototypes of musical instruments. Methods for design and development exist that are specialized in dealing with the unquantifiable values of subjective user experience. These are commonplace in many fields of commercial design, and development and should be so also in the practice of development

[1] http://www.electronic-opera.com/node/774

of new musical instruments and other similar academic and artistic projects.

Involving the users in the design and development process is beneficial and by creating a context where user participation is maximized and users are empowered by short development cycles and prototype-driven test based design, the users can become a great resource to the process. In a project with artistic goals, only that which can be considered as a technological black box can be developed without an artistic perspective.

The users provide not only what they can verbalize. Involving the users with hands-on, open exploration brings tacit knowledge and practice to light. Exploring situations and scenarios to learn together with the users can be much more revealing than asking specific questions: Sometimes the important questions are answered without ever being asked.

Acknowledgments

A special thanks to the acclaimed opera tenor Håkan Starkenberg for a great performance with the prototype in the final test.

7. REFERENCES

[1] Sound and Music Computing Conference, http://smc2010.smcnetwork.org/.

[2] The International Conference on New Interfaces for Musical Expression, http://www.nime.org/.

[3] B. Westerlund, "Design space exploration - cooperative creation of proposals for desired interactions with future artifacts," Ph.D. dissertation, Royal Institute of Technology (KTH), 2009.

[4] F. Treves, *The Elephant Man and Other Reminiscences*, 1923.

[5] Arduino, http://www.arduino.cc.

[6] SuperCollider, http://supercollider.sourceforge.net.

[7] A. Hunt and M. Wanderley, "Mapping performer parameters to synthesis engines," *Org. Sound*, vol. 7, pp. 97–108, August 2002.

[8] H. Rittel and M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, no. 4, 1973.

[9] C. Bornand, A. Camurri, G. Castellano, S. Catheline, A. Crevoisier, E. Roesch, K. Scherer, and G. Volpe, "Usability evaluation and comparison of prototypes of tangible acoustic interfaces," in *Proceedings of ENACTIVE05*, 2005.

[10] S. Lindquist, "The researchers role at stake the meeting between the objective researcher and the subjective individual," *CID-307 Technical report CID/KTH*, 2005.

[11] J. Singer, T. Lethbridge, N. Vinson, and N. Anquetil, "An examination of software engineering work practices," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, ser. CASCON '97. IBM Press, 1997, pp. 21–.

[12] S. Bødker and Y. Sundblad, "Usability and interaction design - new challenges for the scandinavian tradition," in *Behaviour and Information Technology, Volume 27, Number 4*, 2008, pp. 293–300.

[13] M. Leman, *Embodied Music Cognition and Mediation Technology*. The MIT Press, 2007.

[14] J. Gould and C. Lewis, "Designing for usability: key principles and what designers think," *Commun. ACM*, vol. 28, pp. 300–311, March 1985. [Online]. Available: http://doi.acm.org/10.1145/3166.3170

[15] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[16] E. McNutt, "Performing electroacoustic music: a wider view of interactivity," *Organised Sound*, vol. 8, 2003.

[17] S. Jordà, "Instruments and Players: Some Thoughts on Digital Lutherie," *Journal of New Music Research*, vol. 33, no. 3, pp. 321–341, 2004.

[18] D. Overholt, "The musical interface technology design space," *Organised Sound*, vol. 14, 2009.

[19] D. Prem and R. Parncutt, "The timbre vocabulary of professional female jazz vocalists," in *International Symposium on Performance Science*, 2007, pp. 347–352.