Modular Musical Objects Towards Embodied Control Of Digital Music

Nicolas Rasamimanana, Frederic Bevilacqua, Norbert Schnell, Fabrice Guedy, Emmanuel Flety, Come Maestracci, Bruno Zamborlin
IRCAM - CNRS STMS
Real Time Musical Interactions
1 place Igor Stavinsky, 75004 Paris France
Nicolas.Rasamimanana@ircam.fr

Jean-Louis Frechin, Uros Petrevski
NoDesign
12, passage Turquetil, 75011 Paris France
studio@nodesign.net

ABSTRACT
We present an ensemble of tangible objects and software modules designed for musical interaction and performance. The tangible interfaces form an ensemble of connected objects communicating wirelessly. A central concept is to let users determine the final musical function of the objects, favoring customization, assembling, repurposing. This might imply assembling the wireless interfaces with existing everyday objects or musical instruments. Moreover, gesture analysis and recognition modules allow the users to define their own action/motion for the control of sound parameters. Various sound engines and interaction scenarios were built and experimented. Some examples that were developed in a music pedagogy context are described.

Author Keywords
Music, Gesture, Interface, Gesture Recognition, Audio Processing, Design, Interaction

ACM Classification Keywords
H.5.2 Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Algorithms, Design, Experimentation, Performance

INTRODUCTION
The expressive control of digital sound and audio processing has been the focus of important research over the last decade. In this perspective, gestural and tangible interfaces actually are promising approaches to control abstract, disembodied digital sounds. Such interfaces indeed afford the use of physical movements over digital sounds, and can therefore be used to create an embodied music experience. This approach has been recognized to recreate essential links akin to a musician with his/her instrument [9, 21, 6]. Thanks to the increasing availability of gesture sensing systems, innovative digital musical instruments emerged recently from the computer music community [14], opening several novel approaches for musical expression using digital sounds [7].

Using gesture input and tangible interfaces brings up important design and usage issues. In [22], authors posited that gestural controllers were in most cases designed to fit idiosyncratic needs, and as a consequence were inextricably bound to their creators. As a matter of fact, musical interfaces are often considered as part of an artistic endeavor, and not always meant to be embraced by a large community of users. Even if some counterexamples can be found [22, 10], no clear methodologies or approaches to design and evaluate musical interfaces has clearly emerged yet, which generally makes this research community not integrated in the mainstream of the HCI field.

As a design approach, low-level hardware components such as Arduino [23] have been largely embraced by the computer community, showing clearly that subcomponents, if generic enough, can be widely adopted as a means to build complex systems. This can be paralleled to software environments such as Max/MSP [11] or Pd [17], that are based on building complex applications from relatively simple modular components. In this paper, we describe a project on tangible music interfaces, based on a similar bottom-up approach. Our goal is to provide users with both hardware and software components, considered as building blocks to create various types of embodied music experiences. We aim at giving users opportunities to incorporate objects, gestures and sound materials of their choice in the interaction. We are particularly interested in scenarios where objects and gestures are taken or inspired from other (non-musical) applications, in order to provide some grounds to facilitate gesture control. From a general perspective, this can be compared to the use of natural user interfaces discussed recently by [1].

The paper is structured as follows. We first present our general design approach. Second, we describe hardware and software components. Finally, we describe some examples that were experimented in a music school.

TEI'11, January 22–26, 2011, Funchal, Portugal.

DESIGN STRATEGY

One can oppose two different strategies in designing complex systems: *top-down* and *bottom-up* approaches. The first one consists in creating objects[1] that are fully completed and developed with specific functionalities and uses. As such, objects can hardly be modified and users need to "adapt" to the preconceived functionnalities through learning [12]. On the contrary, in a *bottom-up* design strategy, users can actively create the final working system using simple elements. As such, this approach favors customization, assembling, repurposing. Typically, this approach encompasses do-it-yourself (DIY) such as proposed by Arduino.

As illustrated in Figure 1, we propose a bottom-up approach (points 2, 3, and 4 in Figure 1) we call *user-completed systems* [16], opposed to fully finished interfaces and applications with fixed, predefined usages. Major possibilities include:

- assemble existing basic modules
- "parasite/hack" existing interfaces or musical instruments
- invent new interfaces/usages

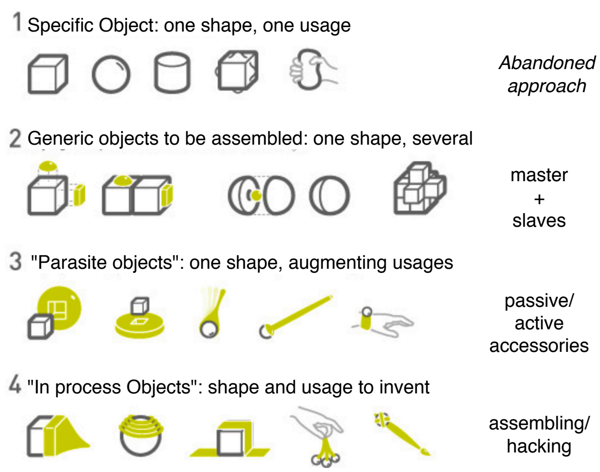

Figure 1. Design approaches

In this project, our goal is to develop hardware and software modules that are intermediate between low-level DIY systems like Arduino or Max/MSP and high level musical system or instruments. Therefore we propose basic elements that require no electronic or programming expertise (without precluding expert users to add new elements). We also propose scenarios or "recipes" to combine elements to create musical interfaces. Similar intermediate level and modular approaches can be found in systems such as the ReacTable [8], BlockJam [15], or the Siftables [13]. However, unlike these examples, we encourage the possibility of hacking and repurposing other objects. Users can hence borrow performing concepts from other disciplines, in visual art (e.g. painting), game (e.g board games) and even sports (e.g ball games). Moreover, using dedicated gesture analysis and recognition software modules, users can propose and incorporate particular motion and actions in the interaction (see the section on software modules).

[1]here the term object must be understood in a broad sense as anything that is produced, a physical object or a software

Hardware Modules

We developed a modular set of objects, called MO, that can be assembled to create tangible sound interfaces. The platform is based on a central wireless module shown in Figure 2. This module, that can be easily held in one hand, contains 6 DOF inertial sensors (accelerometers and gyroscopes), two buttons and LEDs displays and a rechargeable battery. The wireless is based on the IEEE 802.15.4, generally called "Zigbee". We use a Jennic OEM JN-5139.

Figure 2. MO principal module

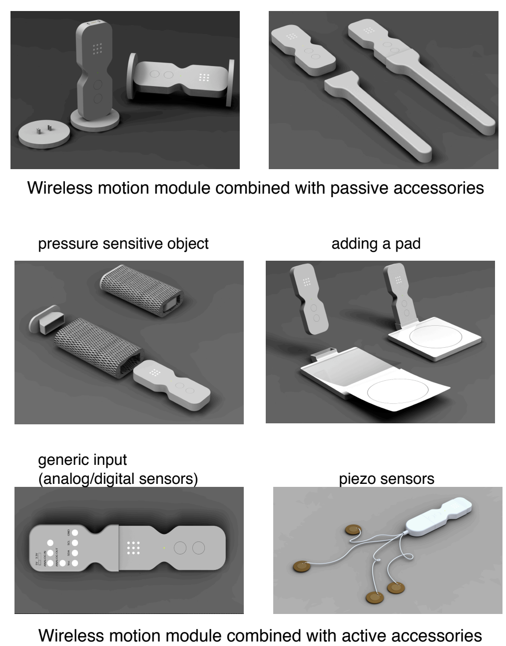

Figure 3. MO accessories (simulation)

Alone, this module allows for motion interactions. Hand gesture can directly be used. But, as already mentioned, we want to stress that its usages can be varied using either passive or active accessories. In particular, active accessories can be directly assembled at each extremity of the modules, using a I2C bus. A first set of accessories is in development (see Figure 3):

- Piezos sensors, e.g. to create a touch sensitive surface
- A pressure sensitive cover, e.g. to make a motion sensible graspable object.
- Generic analog and digital inputs to add any other sensors
- An additional 6 DOF inertial sensors, connected through a cable, e.g. to get relative motion between the module and the accessories

We plan to progressively add new components. Thanks to the ”generic input” accessory, any new sensors can already easily be incorporated.

Software Modules

Similarly to the hardware developments, we conceived modular software components for both gesture/motion analysis and audio processing/synthesis. The elements are programmed in Max/MSP, which facilitates rapid prototyping.

One of the key components of the gesture analysis consists in a gesture recognition and synchronization engine called ”gf” (standing for gesture follower) [2]. This component allows the user to simply record gestures that are then recognized by the system. The main algorithm is based on Hidden Markov Models, but using a non-classical implementation to take into accounts the two following specific constraints. First, a single example can be used for the training, which allows users to easily choose or adapt gesture to their capabilities. Second, the system can perform early-recognition of gestures and as a consequence can continuously synchronize gestures to audio processing. Specifically, the recognition system is designed to accommodate interaction paradigms using both discrete triggering and continuous control.

For audio processing, we integrate a set of synthesis and sound transformation modules [18, 20, 19] that enables to interactively work on recorded sounds. For example, we use various granular and phase vocoder techniques to specifically alter sound characteristics while preserving others (temporally stretching sounds without changing pitch or vice-versa).

EXAMPLE USES

In this section, we present different works integrating the presented hardware and software components. These works were made and used by music teachers and students with the help of our research team. The interaction systems were elaborated in a context of music creation and music pedagogy throughout the year, and were presented during the end-of-year student concert. These experiments were designed within a larger pedagogical context, which description is beyond the scope of this paper.

MO Ball in Collaborative Mode

In this case, the music interaction was based on a ball game, augmented with our wireless module (see Figure 4). It consisted in a percussive sound being played each times the ball is thrown and caught. We experimented with string pizzicati triggered one by one, respecting the score of an actual musical piece. The music performance was thus bound to the continuous and regular ball passing. This collaborative musical game could be played in pairs or with large groups. While everybody can easily play such a game, yet, it requires interesting group interaction, eye-contact, and collective concentration to perform the music. As a matter of fact, ballistic movements (as found when throwing ball) have always been a salient metaphor in music [4]. This indeed encompasses important musical notions such as up-beats, down-beats, phrasing and pulsation. Note that the use of balls as musical interfaces were previously experimented by other authors [5, 3].

Figure 4. MO Ball setup

MO Ball with Continous Control

Contrary to the previous mode where discrete events were triggered, this mode of MO Ball focuses on the control of continuous digital sounds. Users hold the ball and can define their own gestures, which are then used to directly act on the pacing on the sound: sounds are stretched when slowing down and sounds are shortened when speeding up. This mode enables the use of any continuous gestures to lead music, using metaphors ranging from conducting to dance. Users can re-perform music piece using their own gestures and interfaces.

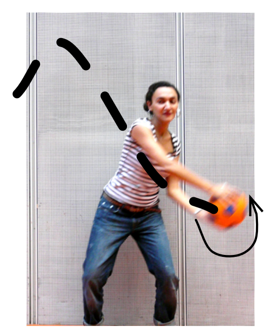

Figure 5. MO ball with continuous control. First, choose one or several sound files and record a set of different gestures. Then, the recognition system along with the sound engine allows the user to gesturally select and control continuously parameters of the sound. For example, sound can be stretched according to the pace of the gestures

MO Chess

In a particular music context, the teacher and students proposed to use a chess game as an interaction metaphor. The game was found to resonate with musical concepts about opposition and dialogue (e.g. canons or duets). A chess table was then augmented with a piezo sensor to register when a piece was put on the table. Players has also attached a MO module to their wrists to track the hand motion above the table. All the sensors were used to control the tempo of a musical piece. Using this setup, performance of a music duet was possible, where each player controls a different music line.

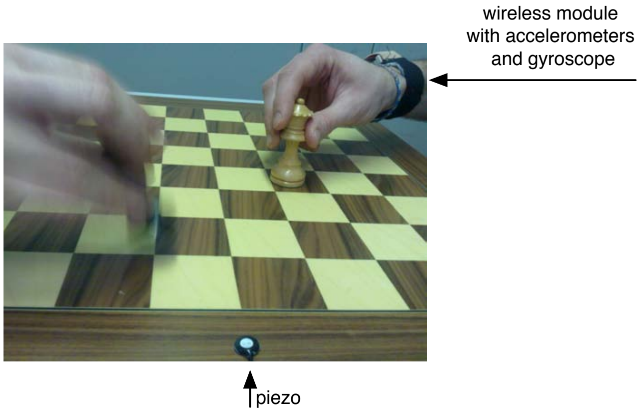

Figure 6. Sound control using a chess board game

CONCLUSION

We presented a modular set of tangible interfaces for sound control, emphasizing gesture and action. Our design strategy was to propose a bottom-up approach: assembling modules, augmenting existing objects and hacking in order to let emerge novel usages with basic interfaces. Moreover, the tangible interfaces are used in conjunction with gesture recognition modules, which invite users to define their own control gestures. Several sound modules were also implemented, based on recorded sound that remained opened to the users. Finally, we presented preliminary works achieved in a music school, which demonstrated promising potential of this approach for music creation and pedagogy. We are currently pursuing the work in this context, developing several other scenarios and applications.

ACKNOWLEDGMENTS

We acknowledge support of ANR and CapDigital (Interlude project ANR-08-CORD-010), and la Région Ile de France. We also thank the whole Interlude consortium and students of l'Atelier des Feuillantines. The second generation of MO prototypes were finalized by DaFact (www.dafact.com).

REFERENCES

1. B. Buxton. Interview on Channel 9. CES 2010. http://channel9.msdn.com/posts/larrylarsen/ces-2010-nui-with-bill-buxton/.
2. F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guedy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In *LNCS*. Springer Verlag, 2009.
3. T. Blaine and T. Perkis. The jam-o-drum interactive music system: A study in interaction design. In *DIS*, 2000.
4. E. Dalcroze. *Rhythm, music and education*. Library Reprints (January 2001), 1921.
5. EZ3kiel. Musical ball. http://www.dailymotion.com/video/x4rub7_ez3kiel-ballon-musical-live-la-ciga_music.
6. R. Godøy and M. Leman, editors. *Musical Gestures: Sound, Movement and Meaning*. Routledge, 2009.
7. International Conference on New Interfaces for Musical Expressions. http://www.nime.org.
8. S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunnera. The reactable: Exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI*, 2007.
9. M. Leman. *Embodied music cognition and mediation technology*. The MIT Press, Cambridge, Massassuchetts, 2007.
10. J. Malloch, D. Birnbaum, E. Sinyor, and M. Wanderley. Towards a new conceptual framework for digital musical instruments. In *DAFx*, 2006.
11. Max/MSP. http://www.cycling74.com.
12. J. McGrenere, R. Baecker, and K. Booth. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of CHI*, 2002.
13. D. Merrill, J. Kalanithi, and P. Maes. Siftables: Towards sensor network user interfaces. In *TEI*, 2007.
14. E. Miranda and M. Wanderley. *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*. A-R, 2006.
15. H. Newton-Dunn, H. Nakano, and J. Gibson. Block jam: A tangible interface for interactive music. *Journal of New Music Research*, 32(4):383–393, 2003.
16. NoDesign. http://www.nodesignlab.net/.
17. Pure Data. http://puredata.info/.
18. N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Muller. Ftm - complex data structures for max. In *ICMC*, 2005.
19. N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi. Mubu and friends: Assembling tools for content based real-time interactive audio processing in max/msp. In *ICMC*, Montreal, Août 2009.
20. N. Schnell and D. Schwarz. Gabor, multi-representation real-time analysis/synthesis. In *DAFx*, 2005.
21. M. Wanderley and M. Battier, editors. *Trends in Gestural Control of Music*. Ircam, 2000.
22. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3):62–76, 2002.
23. N. Zambetti. http://www.arduino.cc/.