A Flexible Tool for the Visualization of Musical Mapping Networks

Master's Thesis Proposal; Music Technology
Aaron Krajeski

Introduction

In a digital musical instrument (DMI) the physical control surface is separate from the sound synthesizer [1], greatly differentiating it from its acoustical counterparts. Thus the mapping of control surface outputs to synthesizer inputs becomes a critical factor in the implementation of DMIs. It is often necessary for performers and composers with little programming expertise to quickly customize mappings for specific performances and pieces. Libmapper [2], an Open Sound Control (OSC) [3] based application programming interface (API), has been developed at the Input Devices for Music Interaction Laboratory (IDMIL) to help accomplish this task.

In the current graphical user interface (GUI) for libmapper, mappings are created by patching[1] together control and synthesis parameters from two lists. These lists are searchable and filterable to aid performances with many instruments. However, this approach is not scalable if many devices exist, as it gives no impression of the network's higher-order structure.

The goal of this research is to design and develop a flexible visual display for libmapper. The final interface will be capable of demonstrating the network's overall structure while also providing information about individual connections. The parameters of devices and signals can be freely customized to the visual display's attributes. In many ways this project is designed to be a visual mapper for the mapping interface. The tool will allow the user to select which features of the network shall be associated with which dimensions of the visual display in order to best communicate the state of the system.

Previous Work

The tremendous expansion of data set sizes in our information era [4] has begotten a similar theoretical expansion for displaying said information. Grounded in Tukey's [5] assertion that we must be "approximately right, rather than exactly wrong," the works of Tufte [4, 6] expound upon the best practices for line diagrams, data labels, colors and layouts in evidence displays.

For pure visualization, Braun[2] gives users basic displays of OSC data flows. The Allosphere [7] at The University of California is a building-sized, spherical display built for the navigation of large sets of data using auditory and visual cues.

On the interface side, many prior connection-type interfaces rely on a patching metaphor [8][3], while [9] describes mathematical ways of "bundling" connections to reveal a network's higher order structure. Simple list interfaces are also common [10][4], wherein users select control surface outputs and associated synthesizer inputs from drop down menus. Building upon the patching metaphor, [11, 12] locate the network's inputs and outputs in space, and nest parameters within certain structures, such as instruments. The EaganMatrix[5] uses a distinctive connection metaphor built around a matrix of input and output parameters. Users make connections between parameters by placing a "pin" at their intersection on the matrix.

Working towards a standard for networked gestural communication in music, [13] details a standardized vocabulary and syntax for describing signals. Standards of OSC networking, both lexical and visual, are

---

[1] Clicking and dragging between parameters, creating a visual connection between them. Analogous to the patch bays in old synthesizers.

[2] J. Bullock, "Braun." http://lists.create.ucsb.edu/pipermail/osc_dev/2008-March/001327.html

[3] David Robillard, "Patchage." http://drobilla.net/software/patchage/

[4] Wildora, "Osculator." http://www.osculator.net/

[5] http://www.hakenaudio.com/Continuum/eaganmatrixoverv.html

presented in [14, 15]. [16] describes a system for visualizing information sent over a musical network.

Proposed Research/Methodology

This project will be structured in four major parts: (i) a review of prior visualized mapping interfaces, (ii) the updating and integration of presently available GUIs for libmapper, (iii) extension of interface features, and (iv) collection of feedback from longterm users of libmapper.

I will begin part one by reviewing previous work in visualized mapping interfaces including [11, 12, 10], with special attention paid to visual features displaying the state of the system. Other connection-based interfaces [8, 14][3] will also be reviewed for effective visual features.

Presently, three GUIs exist for libmapper: mapperGUI, Vizmapper[11] and webmapper. MapperGUI is built using the MaxMSP language. It is currently the most up-to-date and feature-rich of the GUIs, but has the disadvantage of not being cross-compatible and relies heavily on third party software. Webmapper is an Internet browser-based extension for the libmapper library, using a list-and-connection metaphor similar to mapperGUI. The front-end of the application operates upon JavaScript and HTML5, while the back-end communicates with a custom libmapper monitor using Python. Though it is not presently as visually refined as the mapperGUI, the portable, cross-platform nature of webmapper is a more natural fit for a standard GUI for libmapper. This project is essentially a continuation of Vizmapper, an enhanced visual extension to webmapper that is unfortunately no longer functional due to recent updates to the libmapper APIs.

In the second part, Vizmapper will be updated so that it functions with current libmapper API, and integrated as an option alongside the current webmapper GUI. Webmapper will then be updated to include visual features from mapper GUI.

During the third portion, extensions to the overall interface will be designed in JavaScript using the d3 visualization library[6]. They will include options for patching matrices, devices grouped by location on screen and heirarchical edge bundling [9]. Connected block diagrams as in [8] and force diagramswill be explored. The current input/output list model will also be maintained. No single overall structure is to be forced upon users, as flexibility is key. What may be a good arrangement for certain networks may be overcomplicated, obscure or confusing for others. The power of configuration will be given to the user.

Within these visualization schemes, signal and device attributes, such as spatial position, update rate and device type, can be correlated to visual parameters like size, color, line-weight and position of objects. The goal is to create a sort of "meta-mapper," where users are free to connect the devices and signal features with visual properties that are best suited to their network and creative style.

Part four will provide the GUI to long-term users of the libmapper library and will receive feedback. Though this is concentrated at the end of the project, a user-centered design [17] approach will be utilized throughout parts one through three. Users feedback will be a constant part of the iterative design process.

Contributions

Mapping is an essential feature of DMIs, and libmapper is a great open-source solution for performers and composers who wish to experiment with their mappings. An intuitive, flexible interface for libmapper is a necessity if more users are to adopt it. Not only will a visual tool help expert operators, it will also make libmapper more accessible for novice users of computer instruments.

This research will provide a review of data visualization literature with an emphasis on musical and mapping application displays. Such information will be useful to the designers of musical software as these tools continue to progress into a solidly multi-modal realm. As mentioned above, several different models for visualizing mapping software already exist in present applications. Because it is a flexible interface, this thesis may be insightful as to which metaphors are more useful in specific contexts.

---

[6]A JavaScript library for manipulating documents based on data. http://d3js.org/

References

[1] A. Hunt, M. M. Wanderley, and R. Kirk, "Towards a model for instrumental mapping in expert musical interaction," (Berlin, Germany), 2000.

[2] J. Malloch, S. Sinclair, and M. M. Wanderley, "A network-based framework for collaborative development and performance of digital musical instruments.," *R. Kronland-Martinet, S. Ystad, and K Jensen. (Eds.): CMMR 2007, - Proc. of Computer Music Modeling and Retrieval 2007, Conference, LNCS 4969. Berlin Heidelberg: Springer-Verlag*, pp. 401–425, 2008.

[3] M. Wright and A. Freed, "Open soundcontrol: A new protocol for communicating with sound synthesizers," (Thessaloniki, Hellas), pp. 101–104, International Computer Music Conference, 1997.

[4] E. R. Tufte, *Beautiful Evidence*. Graphics Press, 2006.

[5] J. W. Tuckey, "The technical tools of statistics," *The American Statistician*, vol. 19, pp. 23–28, April 1965.

[6] E. R. Tufte, *Envisioning Information*. Graphics Press, 1990.

[7] T. Höllerer, J. Kuchera-Morin, and X. Amatriain, "The allosphere: A large-scale immersive surround-view instrument," (New York, NY, USA), p. 3, ACM Press, 2007.

[8] J. Bullock, D. Beattie, and J. Turner, "Integra live: a new graphical user interface for live electronic music," (Oslo, Norway), Proceedings of the International Conference on New Interfaces for Musical Expression, 30 May - June 1 2011.

[9] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Transactions on Visualization and Computer Graphiscs*, vol. 12, September/October 2006.

[10] STEIM, "Products of interest," *Computer Music Journal*, vol. 28, pp. 105–107, Summer 2004.

[11] V. Rudraraju, "A tool for configuring mappings for musical systems using wireless sensor networks," Master's thesis, McGill University, Montreal, Canada, December 2011.

[12] G. Booth, "Inclusive interconnections: Towards open-ended parameter-sharing for laptop ensemble," Master's thesis, University of Huddersfield, Huddersfield, England, 2010.

[13] A. R. Jensenius, T. Kvifte, and R. I. Godøy, "Towards a gesture description interchange format," in *Proceedings of the 2006 International Conference on New Interfaces for Musical Expression (NIME06)*, (Paris, France), 2006.

[14] T. Place and T. Lossius, "Jamoma: A modular standard for structuring patches in max," (New Orleans, LA, USA), International Computer Music Conference, 2006.

[15] M. Baalman, "Sense world data network." http://sensestage.hexagram.ca, November 29, 2011.

[16] N. Wolek, "The mpg carepackage: coordinating collective improvisation in max/msp." Extended Abstract - Submission for SEAMUS 2010, Stetson University - DeLand, FL, USA.

[17] R. Kling, "The organizational context of user-centered software designs," *MIS Quarterly*, vol. 1, pp. 41–52, December 1977.