INFORMATION SHARING IN NETWORKED MUSIC APPLICATION

Norikazu Mitani
Arts and Creativity Lab, IDMI
National University of Singapore

Lonce Wyse
Arts and Creativity Lab, IDMI
Communications and New Media Programme
National University of Singapore

ABSTRACT

Two key issues for facilitating networked ensemble music are information sharing and message routing. The Bridge application has been developed for networked music composition and performance which implements these functions. The Bridge is designed to support networked music composition and live ensemble performance, supporting dynamic connectivity between applications or physical devices and executing message transfer between applications and devices.

1. INTRODUCTION

Electroacoustic music is often part of rich media environments that may include multiple channels of display and control through graphics and physical devices. Creating a work involving the coordination and communication across many media components is a significant challenge typically involving lots of technical arcana. In particular, there may be many components (instruments, devices, graphical objects) designed by composers or designers with different interface capabilities for real time interaction. If these components are distributed across a network, the challenge of managing the addressing, message protocols, and communications between all the components can be daunting. The problem calls for automated support so that composers can focus on the musical issues of interaction rather than the technical issues.

In the context of networked music, Weinberg [1] proposed a theoretical framework for music network architectures. In terms of that framework, we have designed the Bridge application which supports synchronous dynamically reconfigurable decentralized or centralized networks. We consider "message routing" between applications and devices to be one of the most important processes for both composition and live performance. A second related functionality, that of "information sharing", is necessary for effective message routing, particularly in a dynamic environment. When multiple applications and devices are gathered in a network, the list of component inputs and outputs for designing and manipulating message routes between applications and devices must be shared and readily accessible to all.

2. RELATED WORKS

The NRCI[2] PD tool suite has several similarities to the Bridge is spirit and intent. It is designed for exchanging music control data and communication between networked musicians and uses OSC (Open Sound Control)[3] as its underlying protocol.

NRCI includes three important protocols. The request protocol provides the capability to initiate specific control data broadcasting from another networked musician. The command protocol supports sending command names and values to a target networked musician. Finally, a chat protocol supports the exchange of text messages between all musicians in one network.

NRCI PD tool suite users can request types of control data: pitch, amplitude, duration and onset, but does not provide support for advertising the availability of other streams named by the users. This limitation is mitigated somewhat by the ability to broadcast data with arbitrary names (e.g. modulation index), but effective use of this data requires negotiation between sender and receiver and/or preplanning making it difficult to use in an improvisatory context. These issues are addressed in the design of the Bridge.

3. BRIDGE APPLICATION OUTLINE

The Bridge embodies our approach for networked music design and performance in a small local network [4] with

- a connection interface for music applications, video applications and physical devices,
- a flexible I/O parameter design for application/device designers,
- sharing application I/O parameter information,
- message routing between applications and devices.

The Bridge works as a messaging hub/router of OSC datagrams. This application supplies an interface for creating connections between audio/video applications or physical devices. Currently, applications and devices that use OSC (Max/MSP, PD, Super Collider, Processing, etc.) and MIDI interfaces are supported.

Bridges communicate with each other be sending either configuration information (which changes dynamically) or music control data via "sender" and "receiver" parameter units. Each application or device with sender

parameter registered with the Bridge can send music control data to other application/device with registered receiver parameter (Figure 1). The Bridge is implemented in Sun Java SE 6 for platform interoperability. It has been tested on systems running Microsoft Windows and Mac OS/X.

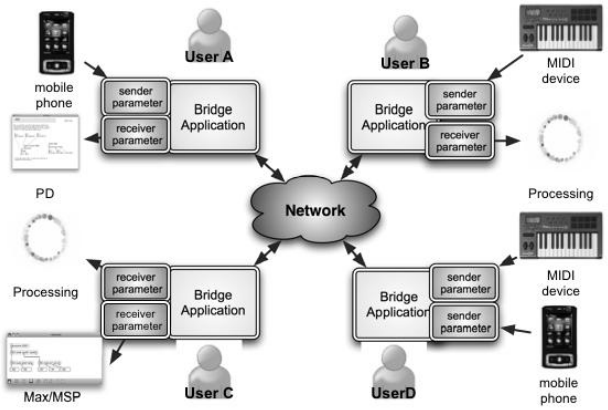

Figure 1. Bridge overview. Each user may have multiple devices or applications that register performance data sender and/or receiver objects with the Bridge.

4. INFORMATION SHARING

The Bridge is responsible for communicating information about the available senders and receivers among all performers.

When a new Bridge instance wants to join the network, it communicates with the other Bridges with a UDP broadcast message containing

- Bridge information (its name, IP address and peer-to-peer IP port),
- I/O parameter information (sender/receiver parameters),
- patch information (control data routing from sender to receiver).

Bridges all maintain the same information at all times. Bridge users can dynamically modify sender/receiver parameters during music performance. Bridge users can also add and delete message routes between any sender and receiver at any time. Modified parameter and routing information will immediately be shared with all other Bridges. These three pieces of information listed above are visualized in control panel of the Bridge which also serves as a graphical user interface for changing the network configuration.

When Bridges receive music control data from sender parameters locally or from other Bridges, the configuration information is locally updated so that each Bridge knows how to map and route future data (Figure 2), and so that the Bridge can provide visualization and a graphical interface for manipulating the current architecture. This is how the Bridge routing functionality is supported by information sharing.

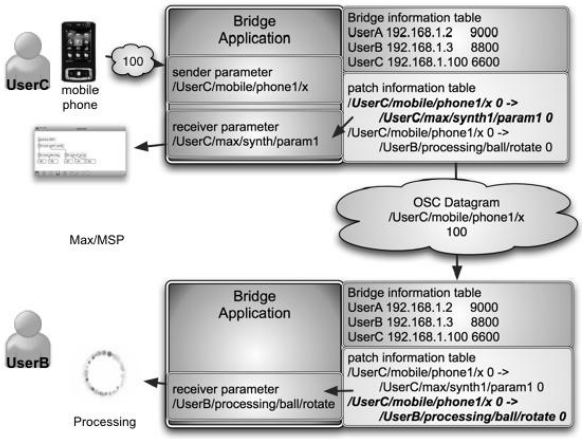

Figure 1. Information sharing with the Bridge. The Bridge determines control data destinations according to the Bridge information table and patch information table.One patch contains both sender and receiver information.

5. EVALUATION AND FUTURE WORK

In this paper we have described the information sharing aspects of the Bridge architecture for networked music. We have successfully used the input/output parameter list sharing and message routing by the Bridge in several demonstrations. In the demo, each Bridge users can append/delete parameters and change the patch at any time. Our next step is to get the Bridge system in to the hands of composers and media creators for real-world testing and user feedback. Increasing the range of device and application support is another important topic also. One of our targets is a TCP socket interface so that the Bridge support Action Script 3 for Flash and Flex applications. The ultimate goal of the system is hide the technology and support research in musical communications.

REFERENCES

[1] Weinberg, G. “Interconnected Musical Networks: Toward a Theoretical Framework" *Computer Music Journal* 29/2 2005.

[2] Burns, C., and Surges, G. "NRCI: Software Tools for Laptop Ensemble." *Proceedings, International Computer Music Conference* Belfast, Northern Ireland, 2008.

[3] Wright, M., and Freed, A. “Open Sound Control: a new protocol for communicating with sound synthesizers”, *Proceedings of the International Computer Music Conference*, Thessaloniki, Greece, 1997, pp 101-4.

[4] Wyse, L. and Norikazu, M. “Bridges for networked musical ensembles”, *Proceedings of the International Computer Music Conference*, Montreal, Canada, 2009.