MaxLink: a New Tool for Networked Performance

Jesse Kriss
IBM Research
jesse.kriss@us.ibm.com
*work done while at Carnegie Mellon University

Abstract
Despite the prevalence of communication protocols in interactive music systems, combining disparate audio and visual systems is a difficult task. This paper introduces MaxLink, a set of Max objects and Java classes that establish a communication channel between Max/MSP and Java.

1 Introduction
Of the numerous systems for creating interactive media works, Max/MSP is a well-known tool for creating interactive audio works. Processing (Reas, Fry 2003) is an innovative tool for creating interactive graphics built as an extension of Java. While Max and Processing each have networking capabilities, establishing useful communication between the two is quite complicated. MaxLink is a simple and easy to use link between the two environments, allowing for quick and seamless communication.

2 Design
The MaxLink system is comprised of a set of Max objects and Java classes connected using local UDP multicasting. Typecasting is handled automatically through Java introspection. While designed specifically for use with Processing, MaxLink works with any Java program.

2.1 Max objects
The jk.link Max object is the core Max element; it acts as a proxy for the Java application. The object is initialized with the network name of the Java program and, optionally, the number of Max inlets and outlets.

2.2 Java classes
In the Java program, inlets on the Max object are mapped to public variables or methods. Output is sent to Max using the MaxLink.output() method.

The translation between Max and Java is simplified by the use of introspection. MaxLink inspects variable types and method signatures present in the Java code, and uses this information to automatically handle type casting.

Figure 1. A Max patch and Java code snippet illustrating MaxLink communication. Changing the value in the number box will alter the value of foo.

3 Advanced capabilities
Because MaxLink operates over multicast UDP, many instances of MaxLink (“network members”) can coexist simultaneously.

In addition to the basic proxy-style communication link, the MaxLink package also includes an object, jk.scout, that will listen for new network members on the local network. When a new member is detected, jk.scout automatically creates a corresponding jk.link object. This allows the network to expand seamlessly as new members join, creating the potential for an easily configurable, flexible interconnected music network, of the kind described by Gil Weinberg (2002).

4 Acknowledgements
Thanks to Brady Lay, and to Golan Levin for his guidance and insight during the development of MaxLink.

References


See also: http://jklabs.net/maxlink