

The KnittingFactory: An Infrastructure for Distributed Web Applications

Distributed Systems Laboratory at New York University
www.cs.nyu.edu/milan/knittingfactory

While Java creates a new opportunity for collaborative and distributed applications on the World Wide Web, some problems remain unsolved. Among these are: (1) how can Java applets find other members of the collaboration session, (2) how to deal with the restrictions imposed by the Java security model, and (3) how to overcome the inability of applets to communicate directly. *KnittingFactory* addresses these problems by providing a set of well-integrated solutions.

Many distributed Java applications are realized as a single stand-alone Java application, the *initiator*, and a set of applets distributed over the Web. This is the setting for KnittingFactory.

Finding Partners: To assist an application trying to find other participants, KnittingFactory provides a registry service. A number of registries are installed on HTTP servers, forming a directed graph. An application can register with one or more registries, thereby publicizing its request for partners. A user wanting to join a collaboration starts a search at any registry (see Fig. 1). This is executed entirely within the user's browser (implemented in Java Script) and the graph is traversed using a breadth-first search. Once a suitable application is found, the browser starts downloading the corresponding applet.

Arbitrary Origin of Applets: In a typical collaborative application, applets are downloaded from an HTTP server and establish a connection to another participant. Under Java's host-of-origin policy, an applet can connect only to the host it was downloaded from, which means that an initiator application must be running on the same host as the HTTP



Figure 1: Main page of a KnittingFactory registry

server. KnittingFactory allows the initiator application to run on any host on the Web. This is achieved by embedding a set of minimal HTTP server functionalities in the application itself—specifically, the capability to serve Java applets.

Applet-to-Applet Communication: The host-of-origin policy prevents applets from directly communicating among themselves. Applets that belong to the same distributed computation can exchange arbitrary information by relaying it through the initiator application. Therefore, there is no inherent reason why applets should not be allowed to communicate directly. KnittingFactory allows direct applet to applet communication by taking advantage of Java's Remote Method Invocation capabilities.

At startup each applet registers a remote object

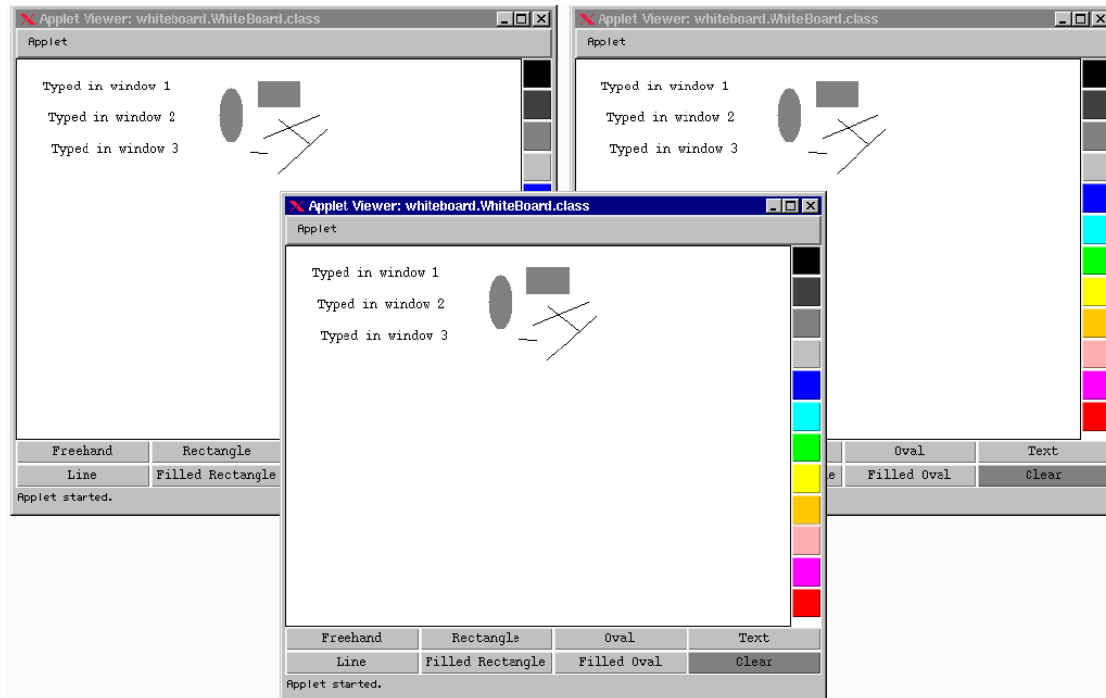


Figure 2: The shared white board demo application

with the initiator application, which distributes it to other registered applets. Applets can then invoke other applets' methods directly and pass along information as parameters. Other communications interfaces such as I/O stream can be realized on top of this mechanism. Apart from relieving the originating host from the burden of relaying messages, this also improves the fault tolerance—after the applets have formed a collaborative group, the initiator application is no longer needed for communication purposes.

As an example of KnittingFactory, Fig. 2 shows three browsers running applets which participate in a shared white board. Note that there is direct applet-to-applet communication.

KnittingFactory provides a flexible infrastructure making it a candidate not only for common distributed Web applications such as shared white boards, editors, calendars, etc., but also for parallel computing environments such as *Charlotte* and *Javelin* [2], and collaborative systems such as the

Java Collaborator Toolset [3].

For detailed information on KnittingFactory, see www.cs.nyu.edu/milan/knittingfactory and [1].

References

- [1] A. Baratloo, M. Karaul, H. Karl, and Z. Kedem. KnittingFactory: An infrastructure for distributed Web applications. TR1997-748, New York University, 1997.
- [2] P. Cappello, B. Christiansen, M. Ionescu, M. Neary, K. Schauer, and D. Wu. Javelin: Internet-based parallel computing using Java. In *Proc. ACM Workshop on Java for Science and Engineering Computation*, 1997.
- [3] B. Kvande. The Java Collaborator Toolset: A collaborator platform for the Java environment. Master's Thesis, Dept. of Computer Science, Old Dominion University, 1996.