



Department of Electrical Engineering
and Computer Sciences
University of California
Berkeley, California 94720

Dynamic Deployment Of Peer-to-peer Networked Applications To Existing World-wide Web Browsers

Wan-teh Chang

David G. Messerschmitt

An abstract selected for publication in the
*Proceedings of the Telecommunications Information Network
Architecture (TINA) '96 Conference,*
Heidelberg, Germany, September 3-5, 1996.

1.0 Introduction

Networked (as opposed to *stand-alone*) applications execute across a network, provide functionality to human users, and can be divided into two classes: *user-to-user* and *user-to-information server*. There are two architectures for realizing them [3][4]:

- *Peer-to-peer*, in which two or more users interact with *peer computers or terminals*, which in turn communicate over the network with one another to provide some user-to-user functionality. Examples include telecommunications (e.g., voice telephony and video conferencing) and collaborative computing (e.g., shared whiteboards and shared editors).
- *Client-server*, in which a user interacts with a *client computer or terminal*, which in turn communicates over the network with a *server computer* for information access (e.g., file transfer, World Wide Web (WWW) browsing, and video on demand). User-to-user applications can also be implemented in this architecture (e.g., Internet Relay Chat (IRC) and CU-SeeMe).

While user-to-user applications have been very successful, they are far less diverse than user-to-information server applications. We believe this is primarily due to the greater deployment hurdle faced by new applications of this class. When an information server is established, any client (or rather its associated user) immediately derives full benefit. On the other hand, the utility of a user-to-user application to a user is strongly related to the number of other users who have the

application. Success depends on a critical mass of users, and early adopters derive little benefit. Economists call this the problem of *network externality*.

2.0 The dynamic deployment approach

Applications, including real-time applications like voice and video, are increasingly implemented in software on programmable terminals and can be distributed over the network. This fast distribution mechanism has dramatically increased the innovation rate of client-server Internet applications (e.g., WWW browsers, document viewers, and audio/video players). Our observation is that network distribution has the potential to make a *much bigger impact* on user-to-user applications, since getting application descriptions to many users is the key to overcoming network externality. However, a user currently has to anticipate the need and execute the relatively sophisticated and time-consuming network file transfer and installation, presenting a continuing externality and rendering unlikely opportunistic participation in such applications. Other obstacles are multiple instruction sets and operating systems and the security problems associated with downloading binary executables from untrusted sources.

Our proposed solution is to transfer executable application descriptions, written in a platform-independent, secure language, over a high-speed network from a repository server, or from one of the participating peer terminals, to all the participating terminals automatically and transparently during session establishment and/or dynamically during a session. We call this *dynamic deployment*. Previously, we have prototyped dynamic deployment of user-to-user applications using Ptolemy and Tcl [2].

3.0 Dynamic deployment based on Java and WWW

The power of dynamic deployment has recently been recognized in user-to-information server applications (e.g., Java [1] programs called *applets* running inside Java-enabled WWW browsers). The present WWW browsers use a “hyperlink following, file pulling” browsing mechanism, which can establish Java applets as stand-alone applications or clients in client-server applications but cannot establish a peer-to-peer session defined by a pair of Java applets running in Java-enabled browsers. We have prototyped an extension of dynamic deployment to user-to-user applications in a peer-to-peer architecture using a WWW environment, called “Java-to-Share.” We use Java as our user-to-user application description language, and a Java-enabled WWW browser, Netscape Navigator, as the standardized peer platform. (Note that WWW browsers were not designed for a peer-to-peer architecture, demonstrating the flexibility afforded by Java.)

Java-to-Share includes two helper programs, which are permanently installed in the originating peer and the responding peer, run in conjunction with WWW browsers, and incorporate a coordination protocol built on top of TCP/IP and HTTP. In addition, we assume that the originating peer owns the Java applets, which are exported by a local HTTP server. The

originating peer places a session establishment request, using the coordination protocol to alert the responding peer. If the session request is accepted, the helper programs instruct their respective WWW browsers to download the Java applets from an HTTP server at the originating peer. The Java applets first connect to the helper program of the initiating peer to synchronize and exchange information (e.g., the TCP port on which an applet is listening), and then interconnect. We have also developed two simple example applications – a text-based peer-to-peer application similar to the “talk” program, and a shared whiteboard – that can be deployed using Java-to-Share. Additional applications can be added by supplying interoperable peer Java applets.

4.0 Conclusions

Dynamic deployment avoids a major obstacle to user-to-user applications – the requirement that the users have previously obtained and installed the necessary interoperable application software – and thus will encourage a proliferation of innovative new user-to-user applications. As we have demonstrated, user-to-user applications in a peer-to-peer architecture can build on unmodified Java-enabled WWW browsers, supplementing their previous role in client-server applications. Java-to-Share allows a given user to participate in a user-to-user application with any other user running a Java-enabled browser and associated Java-to-Share helper, without the need for that user to have obtained the application description in advance.

Our approach to application deployment can be viewed as creating a new *spanning layer* [5] within the networked computing environment at the level of a Java virtual machine. Such a spanning layer will facilitate the dynamic deployment of interoperable descriptions of networked applications. We believe this has the potential to revolutionize the space of user-to-user applications, as it bypasses externality obstacles that have been a serious hindrance in the past.

5.0 Acknowledgments

This research is supported by the Semiconductor Research Corporation (SRC) (contract 96-DC-324), the State of California MICRO program, and Bell Communications Research and Pacific Bell. Useful comments from Weiyi Li, Robert Wilensky, and Scott Shenker are gratefully acknowledged.

6.0 References

1. K. Arnold and J. Gosling, *The Java Programming Language*, Addison-Wesley, 1996.

2. W.-T. Chang, W.-Y. Li, D. G. Messerschmitt, and N. Chang, "Rapid Deployment of CPE-Based Telecommunications Services," *Proceedings of GLOBECOM'94*, vol. 2, pp. 876-880, San Francisco, California, USA, Nov. 28 - Dec. 2, 1994.
3. D. G. Messerschmitt, "The Convergence of Communications and Computing: What are the Implications Today?", to appear in *Proceedings of the IEEE*, August 1996.
4. D. G. Messerschmitt, "The Future of Computer Telecommunications Integration," *IEEE Communications Magazine*, Vol. 34, No. 4, pp. 66-69, April 1996.
5. David Clark, "Interoperation, Open Interfaces, and Protocol Architecture", unpublished white paper contributed to the NII 2000 Steering Committee, National Research Council.