A Classification of Application Sharing Tools for Use in a Virtual Classroom

Gerhold B. Kooper¹, Prof. Alfredo Terzoli²
Computer Science Department, Rhodes University, South Africa
Tel: +27 46 6038247, Fax: +27 46 6361915
e-mail: gbkooper@gmail.com¹, A.Terzoli@mail.ru.ac.za²

Abstract – In this paper we classify different application sharing tools according to factors that would affect their use in a virtual classroom. We exemplify this classification with a subset of different types of application sharing tools, such as conferencing systems, thin-client systems and peer-to-peer systems. We also discuss performance issues relating to application sharing tools.

Index Terms - Application Sharing, Desktop Sharing, Virtual Classroom, Collaboration

I. INTRODUCTION

The Computer Science department at Rhodes University provides distance courses to the Computer Science department of the University of Namibia, to help with staff and research capacity building there. The teaching takes place over the Internet in a virtual classroom.

A. The Virtual Classroom

As stated in [10], “A virtual classroom should not be much different from a real classroom”. In a virtual classroom one would like to have the teaching and learning experience as close as it would be in the real classroom as possible. The difference from a real classroom should be only that a “virtual place” is created, which allows teachers and students to connect over any distance and from any geographical location.

Such a virtual classroom would be a very efficient and practical way of utilizing human resources in a university environment. Lecturers would then not need to prepare and present different lectures for the physical and the virtual classroom.

Within a virtual classroom there is a need for computer application sharing, for example to allow lecturers to run experiments with software.

In an African context, the choice of application sharing technique must take into account low bandwidth and unstable network connections.

B. Application Sharing

From the point of view of a virtual classroom, application sharing can be described as the sharing of the graphical user interface of an application amongst multiple users simultaneously in real-time. All participants have the same view of the application. The whole desktop might be shared rather than a single application, in which case one speaks of desktop or screen sharing. Application sharing and desktop sharing will be considered equivalent in this paper except where the difference is explicitly made.

In this paper “application sharing software” is the software that handles and connects the different participating computers to enable the sharing.

II. DESIGN ARCHITECTURE

An application sharing system can be categorized in terms of how it handles the shared application, namely centralized or replicated [6].

In the centralized architecture there is only one instance of the shared application, whereas the replicated architecture requires each participant to run locally his/her own copy of the shared application.

The replicated architecture, by its design, may be very bandwidth efficient and faster than the centralized architecture as only input streams are sent among participating computers. It was even stated by C. Ming and his team at the HP Software Technology Laboratories that they could share a 3D application over just 56kbps network connection [2].
The replicated architecture has however not survived in application sharing software over the years and we could not find usable software solutions using this architecture. It suffered greatly from the problem of keeping all shared application instances synchronized [5].

Replication architectures seem to be successfully applied in networked gaming. A gaming session may sometimes have more than tens of thousands of concurrent players. In these games, only the actions of the players would be distributed amongst players rather than the graphics updates [11].

III. TYPE OF APPLICATION SHARING SYSTEMS

We can further classify application sharing software in respect to the systems in which they are used. In the subset of application sharing tools presented in Table 1, all the tools use the centralized architecture, but they are different along other dimensions.

C. Application Sharing within Conferencing Systems

In many instances, application sharing would be part of a video or voice conferencing system. The sharing of applications in such systems would be referred to as data conferencing or desktop conferencing. Examples in Table 1 are Citrix GoToTraining, Cisco WebeX, Microsoft Live Meeting and Adobe Acrobat CONNETNOW.

These systems generally have a good mechanism of floor control to handle changes to the application being shared [3].

D. Application Sharing within Thin-Client Computing Systems

A thin-client computing system consists of a server and a client that communicate over a network using a remote display protocol. Thin client systems are increasing becoming popular over the internet with the increasing presence of technologies such as cloud computing. Application sharing can be done with thin-client systems although these systems are not primarily designed for desktop conferencing. With appropriate configuration, however, they can form a desktop conference session. Examples from Table 1 are FreeNX and TightVNC.

A desktop sharing conference was done with the FreeNX server on Ubuntu Linux by allowing participants to attach to an existing desktop display. Three participants shared one screen in this instance.

Sharing of the same display when using TightVNC can be achieved, for example, with the use of software called VNC Reflector that allows broadcasting of a display.

The limitation of these systems would typically be the lack of a built in mechanism for floor control. Floor control would have to be done through voice communication or another communication channel active amongst participants.

E. Application Sharing within Peer-to-Peer Systems

In its pure sense, peer-to-peer (P2P) networks or computing is a distributed system without any centralized control. Nodes act as both servers and clients. Hybrid P2P systems on the other hand have a central server that maintains, for example, information about registered users on the network, in the form of meta-data [1]. Many of the modern P2P systems are designed as hybrid P2P systems.

In the sample software in Table 1 we can regard Skype and MSN Messenger as Hybrid P2P software and they allow application or desktop sharing.

IV. SOFTWARE SOLUTIONS

A list of application sharing solutions was compiled by gathering as much information about these software solutions through the web and specific trials.

In table 1 we present a subset of 10 application sharing solutions from more than 50 solutions that were investigated. We have selected this subset because of the popularity and availability of these solutions, together with their diversity.
Notes:
* These solutions would attempt to use a direct connection. If it fails due to network restrictions, data would be relayed through an intermediate entity.

~ It could not be determined.

The columns in this table are described below.

A. Protocol

The “protocol” column indicates application level protocol used by the solution.

The ITU-T T.120 family of protocols is the only standard for application sharing protocols. Some of the other applications sharing protocols, even the proprietary ones, were developed by adding their own adjustments to the T.120 family.

It is popular for newer solutions to use the HTTP protocol to tunnel their proprietary protocols, as is shown in the subset of solutions presented in Table 1. This allows their solutions to work on different platforms and networks. Most universities and other corporations have network restrictions to protect their networks and may only allow HTTP internet traffic through their networks.

B. Relay

The “relay” column indicates whether the solution uses a central server to relay its screen update data.

C. Remote Control

The “remote control” column indicates whether the solution would allow for remote participants to control the hosted application.

D. Share Specific Application

This column indicates if the solution allows only selected applications to be visible to others in the application sharing session.

E. Change Presenter

The “change presenter” column indicates whether the solution allows the “presenter” role to be handed over to another participant. (A “presenter” in this context is the person in control of the application.)

F. Live Cursor

The “live cursor” column indicates whether every movement of the cursors would be visible on all participating computers in real-time.

G. Multiple Users

The “multiple users” column indicates whether the solution allows more than two participants to access the same shared application at the same time.

H. Type

The “type” column indicates in which type of system the application sharing is used as discussed earlier.
V. PERFORMANCE ISSUES

Centralized application sharing needs both high bandwidth and low network latency to perform effectively. The display area on the screen needs to be typically bigger than one would see for example in a Skype video call in order to read the text on the screen. This would generate large amounts of data to be transmitted over the network.

In terms of the end points, there are settings and configurations of the application sharing software that directly affect the performance. Important ones are the screen resolution, the colour depth, the display encoding, the compression algorithms used, and the cache settings.

The following factors measure the performance of the application sharing software in terms of the experience by the participants when using application sharing software.

A. Latency
Latency is the delay between the moment in which the presenter initiates an activity, such as mouse clicks or cursors move on the screen, and the moment in which the other participants observe the same action on their computers [8].

B. Smoothness
A smooth experience is one that flows easily with a minimum of “choppiness” and fuzziness of images [8]. Smoothness may improve focus of participants and reduce distraction.

C. Fidelity
Fidelity denotes how accurate a copy is to its source. When doing application sharing, presentations with perfect fidelity looks identical to the presenter and the other participants. With a good fidelity, the classroom session experience is improved, reducing the risk of participants to misread and misinterpret.

VI. DISCUSSION

In our virtual classroom it would be preferable to use a solution with the following characteristics:

- Support of multiple users as the classroom might have students in different locations.
- It would not be good for the solution to use relays. The servers used for relay by most tools are very often on other continents such as Europe or North America and they introduce high network latency.
- A solution within a conferencing system would be advantageous as these systems have built in floor control.

In terms of the performance of the application sharing solution chosen, the latency seems most important. Fidelity and smoothness may not affect the activity in the virtual classroom significantly, at least until the transmitted session degrades substantially.

VII. CONCLUSION

In this paper we have presented an initial application sharing classification, with a view of providing the right tool for use in virtual classroom activity. The classification was along dimensions such as the type of systems, the use of relay servers, the type of protocol and others.

We have also classified a subset of application sharing solutions currently in use according to the criteria listed.

Finally we have also discussed the application sharing solutions in terms of usability and experience by users.

VIII. REFERENCES


Gerhold B. Kooper is currently studying towards his Master of Science Degree in Computer Science at Rhodes University, Grahamstown. His main area of interest include application sharing in the virtual classroom.

Alfredo Terzoli is Professor of Computer Science at Rhodes University, where he heads the Telkom Centre of Excellence in Distributed Multimedia. He is also Research Director of the Telkom Centre of Excellence in ICT for Development at the University of Fort Hare. His main areas of academic interest are converged telecommunication networks and ICT for development.