Lightweight Scalable Tool Sharing for the Internet

Agustín J. González
Department of Electronics Engineering
Federico Santa María University
Valparaíso, Chile

Hussein Abdel-Wahab and J. Christian Wild
Department of Computer Science
Old Dominion University
Virginia, U.S.A.

July 2001
Hammamet, Tunisia
Outline

• Introduction
• Odust: a sharing tool engine
• Dynamic image transmission
• Floor control
• Conclusions and future work
Introduction

• Synchronous collaborative multimedia applications
• Basic Components of a synchronous multimedia application:
  – Audio
  – Video
  – **Shared data**
• One approach allows users to share existent single user applications.
• Challenges:
  – **Scalability**
  – **Adaptation to a dynamic set of members**
  – **Data manipulation by any member**
• Our scenario: Interactive distance learning, large group conferences
• Our target: a distributed sharing tool application for these scenarios
Oдust: a sharing tool engine

• Let's consider the following scenario

User: Rodrigo
OS: WinNT

User: Cecilia
OS: Solaris

User: Agustín
OS: Solaris

User: Eduardo
OS: WinNT

Multicast Network
Oduit: sharing tool engine

- Rodrigo’s view

A SEMANTIC-BASED-MIDDLEWARE-FOR-MULTIMEDIA

COLLABORATIVE APPLICATIONS

by

Agustin Jose Gonzalez
M.Sc. Computer Science, December 1997, Old Dominion University
B.Sc. Electronic Engineering, December 1995, Universidad Federico Santa Maria, Chile

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY

May 2000
Odust: sharing tool engine

• Cecilia’s view

UNIX
Odust: sharing tool engine

- Eduardo’s view

User: Rodrigo
OS: WinNT

User: Agustín
OS: Solaris

User: Cecilia
OS: Solaris

User: Rodrigo
OS: WinNT

Multicast Network

WinNT
O Dust: sharing tool engine

- Agustín’s view

UNIX
How does it work?

- It achieves tool sharing by distributing the images of the applications running on participants’ screens.
- It provides a floor control mechanism for any participant to operate shared applications.
- Scalability is achieved by using IP multicasting.
Protocol for Dynamic Image Transmission

• **Sender:**
  * **Temporal redundancy removal**
    » Sample image at regular period
    » Divide image in tiles
    » Process only changed tiles
  * **Spatial redundancy removal**
    » compress and send changed tiles

• **Receiver:**
  » Receive data unit
  » Decompress tile
  » Update tile in image
Overcoming losses

• Each tile is retransmitted after a random time.
• This also accommodates late comers.

Performance Study

* How to select a tile compression technique? (JPEG, GIF, PNG?)
* Is there a “best” tile size? What does it depend on?
* How often to sample the image?
* How can two tiles be compared efficiently?
* Maximum data transmission rate? What does it depend on?
Lightweight Floor Control

- Problem: How to manage exclusive resources in large-scale multimedia applications?

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Request</td>
</tr>
<tr>
<td>2</td>
<td>Granted</td>
</tr>
<tr>
<td>3</td>
<td>Granted</td>
</tr>
<tr>
<td>4</td>
<td>Taken</td>
</tr>
</tbody>
</table>

- Node (participant)
- Shared tool
- Preemptive
- TCP connection
- Heartbeat

Floor holder
Overall Architecture

Capture and Dynamic Compound Image Protocol
Sender

Native Library

Event Injector

Token Manager

Application A

WinNT

Capture and Dynamic Compound Image Protocol
Receiver and Display

Dynamic Compound Image Protocol
Receiver and Display

Token Client

Event Capture

Application A's View

JDesktop

Java VM

Sharing Tool Sender

Sharing Tool Receiver

Multicast

Temporary TCP

Method Invocation
Conclusions and Future Work

• Along with audio and video, data sharing is a crucial component in multimedia collaboration.

• Our data sharing tool distributes images of an application by sending tiles updates when changes are detected. It uses tile retransmission to overcome losses due to multicasting.

• It is based on Java except a small number of methods for image capture.

• We are currently working on using H.263+ as a compression scheme for the application views.

• We are also porting the sampling library to other platforms.