Collaborative Web Browsing Tool Supporting Audio/Video Interactive Presentations

Guillermo de Jesús HOYOS-RIVERA
Roberta LIMA-GOMES & Jean-Pierre COURTIAT

LAAS-CNRS
Toulouse, France

{ghoyos, rgomes, courtiat}@laas.fr

Roberto WILLRICH

Department of Computer Science (INE)
Santa Catarina Federal University (UFSC)
Florianópolis, Brazil

willrich@inf.ufsc.br
Presentation Outline

- Collaborative Web Browsing
- Synchronization Model
- Functional Architecture
- Operational behavior of CoLab
- Conclusions and Future Work
Collaborative Web Browsing

- Users are isolated when they browse the Web

We propose to provide Web users with a group awareness
Collaborative Web Browsing

- The number of Web pages containing embedded continuous media increases quickly
- New challenge for Collaborative Web Browsing
- Particularly useful in e-Learning
Collaborative Web Browsing

- Synchronization of Web Browsing
- Continuous media presentations

Diagram:
- HTTP Server
  - User 1
  - User 2
  - User 3
  - HTTP GET
- RSTP Server
  - User 1
  - User 2
  - User 3
  - RSTP

Buttons:
- PLAY!
- PAUSE!
- STOP!
Collaborative Web Browsing

- Web browsing synchronization solutions
  - Communication tools
    - e-Mail
    - Windows Messenger
    - ICQ
  - Browsing indexing
    - CoBrow
    - Let’s Browse
    - PROOF
Collaborative Web Browsing

- Web browsing synchronization solutions
  - Browsing synchronization
    - Application sharing (VNC)
    - E-coBrowse
    - WABX
    - PROOF
    - Integrated collaboration environments (NetDive, PlaceWare, WebEx, …)
  - Possibility of flexibly creating/releasing browsing synchronization relations
Collaborative Web Browsing

- It seems that there is no currently existing solution allowing users to synchronize embedded continuous media presentations in Web pages.
- Some problems arise, due to:
  - Response times when clicking the “Play” button
  - Connection delays
  - Transmission delays of the RSTP packets
  - Transmission delays of the stream
  - Initial buffering times
Presentation Outline

- Collaborative Web Browsing
- Synchronization Model
- Functional Architecture
- Operational behavior of CoLab
- Conclusions and Future Work
Synchronization Model

- The synchronization state of a collaborative Web browsing session is represented in a tree structure called SDT (Synchronization Dependency Tree).
Synchronization Model

Browsing synchronization in SDTs
Synchronization Model

- Synchronization Relations are dynamically created and released by using Synchronization Requests, which are subject to an authorization protocol:
  - “I_Follow_You”
  - “You_Follow_Me”
  - “I_Leave_You”
Synchronization Model

- The “I_Follow_You” request:

  \[\text{Can I see what you are reading?}\]
  \[\text{Of course!}\]

SDT

\[\begin{array}{c}
A \\
\text{I_Follow_You}
\end{array}\]
Synchronization Model

- The “I_Leave_You” request:

I quit you, See you!
Synchronization Model

- The “You_Follow_Me” request:

A

B

OK, I’m coming!

Do you want to see this information?

You_Follow_Me

SDT
Continuous media presentations synchronization is made according to the synchronization relations among users: the user at the root of a SDT has the control.
Presentation Outline

- Collaborative Web Browsing
- Synchronization Model
- Functional Architecture
- Operational behavior of CoLab
- Conclusions and Future Work
Functional Architecture

- **Client**
  - Login/out & Synchronization Actions
  - Browsing Actions
  - Session Information Exchange
  - Resource Retrieval

- **Broker**
  - Integration API
  - Session Manager
    - Session Information Exchange
    - Browsing Requests

- **Browser**
  - Browsing Controller
  - Media Controller
    - Playback Control & State Changes

- **MediaSync Manager**
  - Media Streaming
  - Media Controller
    - Media Sync Manager

- **External Tools**
  - Integration API

- **Integration API**
  - Capture state changes and execute Playback Control messages
  - Capture state changes and execute Playback Control messages

- **Session Manager**
  - Session Information Exchange
  - Login/out & Synchronization Actions

- **Media Controller**
  - Playback Control & State Changes

Guillermo de Jesús HOYOS RIVERA
Presentation Outline

- Collaborative Browsing
- Synchronization Model
- Functional Architecture
- Operational behavior of CoLab
- Conclusions and Future Work
Current Implementation

**MediaSync Manager**
- Cache Module
  - Caching
  - Get cached

**Browsing Manager**
- Translator
  - Trans. req.
  - Trans. resp.
  - Trans. resp.

**Cache**
- Trans. req.
- Get cached
- Cached?
- Get cached

**Internet/Intranet**
- HTTP request
- HTTP response

**MediaSync Manager**
- Synchronization Module
  - Can access?
  - Synchronize?

**Session Manager**
- Can access?
- Synchronize?

**Broker**
- HTTP req.
- HTTP resp.
- HTTP req.
- HTTP resp.
- HTTP req.
- HTTP resp.

**Synchronous User**
- Playback Control
- State Change

**Asynchronous User**
- Playback Control
- State Change

**Synchronization Module**
- Slave ⇒ URL

**Playback Control**
- State Change

**Playback Control**
- State Change

**Asynchronous User**
- Data
Current Implementation

- Media synchronization mechanism
  - The “PrepareToPlay” command allows us to control the synchronization of the state of the whole presentation in spite of the non deterministic connection and buffering times
Current Implementation

CoLab

The Flexible Collaborative Web Browsing Tool

Please choose the session you want to access...

Available sessions

Distributed Systems
Databases

Submit

Please choose your role and select an username...

Available roles
Student
Teacher

Role password:

Username:
Memo

Submit
Current Implementation

GUI Browsing Frame

[Image of a computer screenshot showing a web browser interface, including Yahoo! and CoLab: The Collaborative Browsing Tool]
Current Implementation

- User’s Identity and Session Name
- Awareness & Synchronization State
- Browsing Controls
- Synchronization Controls (for Users and Roles)
Current Implementation
Current Implementation

$1 \leq N \leq 165$

![Graph showing average retrieval time versus number of users]
Presentation Outline

- Collaborative Browsing
- Synchronization Model
- Functional Architecture
- Operational behavior of CoLab
- Conclusions and Future Work
Conclusions

- Allowing flexible synchronization relations creation/release is a good issue.
- Usable on any standard browser following the W3C standard specification and supporting the automatic proxy configuration and on any HTML resource.
- Continuous media presentations synchronization: a new issue particularly useful in e-learning.
- Flexible extensible architecture.
- Good performance.
Future Work

- Implementation of a distributed version
- API’s development for integrating our platform with other collaborative tools
- Implementing enhanced synchronization mode using roles privileges:
  - “I_Spy_You” & “You_Join_Me”
Thanks

Ευχαριστώ
Obrigado
Grazie
Merci
Gracias
謝謝