The Reconstruction of User-Sessions from HTTP Traces in RIAs

Sara Baghbanzadeh, Salman Hooshmand, Seyed M. Mirtaheri, Muhammad Faheem, Gregor v. Bochmann, Guy-Vincent Jourdan, Iosif Viorel Onut
School of Information Technology and Engineering - University of Ottawa

Introduction

In a Web Application, each user-session generates a series of HTTP requests and responses regardless of technology/device used. It is beneficial to reconstruct user’s session from HTTP traces for several reasons, including:

- Automatic testing: replaying what a user has done
- Debugging: when a bug is reported, we can reconstruct what was actually done to automatically reproduce the fault
- Automatic login: Crawlers can learn how to login automatically to continue their work

Input and Output:
- Input is HTTP traces of user’s previous session recorded by proxy.
- Output is a series of DOMs and the XPath of the elements on which the user has interacted, and inputs were provided by the user during the session

Background

Some methods have been proposed to capture and replay user’s actions in JavaScript applications, e.g.

- Mugshot: logs sequence of JavaScript events executed in a browser to be sent to developers for debugging
- Timelapse: records all events inside browser’s web debugger, with ability to go back and forth for execution
- ClickMiner: reconstructs user sessions from traces recorded by a passive proxy

However, these have either require installation of additional software on user’s machine (as in Mugshot and Timelapse) or has limited support for handling JavaScript events and no ability to extract user-inputs (as in ClickMiner).

We have developed a session reconstruction (SR) tool which reconstructs user’s session based on a set of previously recorded HTTP requests/responses.

The SR tool has two components:

1. SR proxy which responds to HTTP requests from the SR browser based on the traffic captured earlier. The SR proxy replaces the actual application server.
2. SR browser which loads a page, selects and executes events on the DOM and communicates with the SR proxy to rebuild the user session.

The user session is reconstructed using the following approach:

Handling AJAX

AJAX calls are asynchronous, how does SR browser handle this?
- Our SR browser keeps track of sent requests and received responses.
- No event is selected/executed and no sequence check is done while we have pending requests.

Methodology

The SR tool is designed to capture and replay the user’s interactions in AJAX applications. The tool consists of two main components:

- SR Proxy: intercepts all HTTP requests and responses and processes them accordingly.
- SR Browser: interacts with the SR Proxy to reconstruct the user’s session.

The SR tool uses a combination of implicit and explicit clues to reconstruct the user’s session. Implicit clues include HTTP requests and responses, while explicit clues include user input and session state.

References