CSE 608 Fall 2013
Project 2 Report

Vinay Krishnamurthy, Amogh Akshintala

November 1, 2013

Introduction

This project aimed at instrumenting the Chromium browser to record certain events on the browser. Specifically, we had to instrument the following events on the browser:

- Navigation events logging
- DOM events logging
- UI events
- Sensor events
- HTML content

Chromium is a port of Google Chrome browser that has been open sourced. Since we were instrumenting Chromium on Android devices, we had to instrument the browser using both Java and C++ (native code).

List of changed/added files

Here's the list of files that we have modified to instrument the required events on the browser: Note: Here, $ROOT = chromium/src directory.

1. HTTP
   (a) $ROOT/net/http/http_request_headers.cc
   (b) $ROOT/net/http/http_response_headers.cc
   (c) $ROOT/net/url_request/url_request_redirect_job.cc
   (d) $ROOT/net/url_request/url_request.cc
   (e) $ROOT/net/http/http_response_body_drainer.cc
2. GPS
   (a) $ROOT/content/browser/geolocation/location_api_adapter_android.cc
3. Accelerometer
   (a) $ROOT/content/browser/device_orientation/data_fetcher_impl_android.cc
4. Touch Screen events
   (a) $ROOT/content/public/android/java/src/org/chromium/content/browser/TouchPoint.java
5. Media content
   (a) $ROOT/media/base/android/media_player_bridge.cc
6. HTML content
   (a) $ROOT/third_party/WebKit/Source/core/html/HTMLFormControlElement.cpp
   (b) $ROOT/third_party/WebKit/Source/core/html/parser/HTMLSourceTracker.cpp

(c) $ROOT/third_party/WebKit/Source/core/html/parser/HTMLSourceTracker.h

7. DOM Events
   (a) $ROOT/third_party/WebKit/Source/core/loader/FrameLoader.cpp
   (b) $ROOT/third_party/WebKit/Source/core/events/FocusEvent.cpp
   (c) $ROOT/third_party/WebKit/Source/core/events/TouchEvent.cpp
   (d) $ROOT/third_party/WebKit/Source/core/frame/DOMWindow.cpp
   (e) $ROOT/third_party/WebKit/Source/core/dom/Document.cpp
   (f) $ROOT/third_party/WebKit/Source/core/page/EventHandler.cpp
   (g) $ROOT/third_party/WebKit/Source/core/page/FocusController.cpp
8. IPC related files
   (a) $ROOT/content/common/content_message_generators.h
   (b) $ROOT/ipc/ipc_message_start.h

The following new files were added to the Chromium source tree.

1. base/instrumenter.h - Header file with the definitions for the .cc file
2. base/instrumenter.cc - Source that contains the instrumenter class. This class has methods to convert any type to C++ string, and log the data
3. content/renderer/render_instrumenter.h - Header file
4. content/renderer/render_instrumenter.cc - This file contains definition that will dispatch a message using IPC that is to be received by a component on the browser side.
5. content/browser/renderer_host/instrumenter_ipc_listener.h - Header file
6. content/browser/renderer_host/instrumenter_ipc_listener.cc - File that contains class which is supposed to receive IPC message from the renderer.
7. content/common/renderer_instrumenter_messages.h - This file has the necessary IPC macros definitions to facilitate IPC between browser and renderer part of the instrumentation.

Where are the logs written to?

The set of instrumented components that are managed by the browser kernel such as HTTP header logging, redirection and HTTP response logging are written to a file called Instrumented_data_PID.xml file. It is written on the /sdcard directory. The file name and the location are configurable with the *instrumenter* object constructor.

The set of components that are managed by the renderer are still written to logcat, with log level = verbose and *Proj3* as the tag. We tried writing the logs generated by this component to a file, however we had major issues with this approach. This is due to the fact that the renderer process does not have permissions to write to file system, by design. Hence the write operation will have to be delegated to the browser kernel component using IPC. However, the IPC documentation for Chromium is very poor, and they're very confusing, at best. We've included these files, which have IPC communication between renderer and browser, as well as part of our submission.