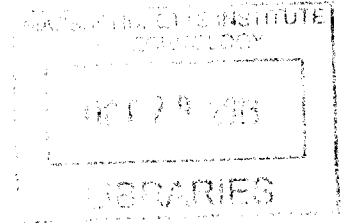


# High Fidelity Website Research - Using a Browser Extension to Provide a Natural Environment

ARCHIVES



by

Brandon H. Baker

S.B., C.S. M.I.T., 2012

Submitted to the Department of Electrical Engineering and Computer  
Science

in Partial Fulfillment of the Requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

© Massachusetts Institute of Technology 2013. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
..... May 24, 2013

Certified by .....  
.....  
Glen L. Urban  
David Austin Professor in Management  
Thesis Supervisor

Accepted by .....  
.....  
Prof. Dennis M. Freeman  
Chairman, Masters of Engineering Thesis Committee



# High Fidelity Website Research - Using a Browser Extension to Provide a Natural Environment

by

Brandon H. Baker

Submitted to the Department of Electrical Engineering and Computer Science  
on May 24, 2013, in partial fulfillment of the  
requirements for the degree of  
Master of Engineering in Electrical Engineering and Computer Science

## Abstract

People are spending ever increasing amounts of time online. As a result, companies are investing greater amounts of money into online advertising in an effort to influence their behavior. The impact and effectiveness of these ads is still an open question. One possible method of analyzing the effectiveness is through the analysis of clickstream data. However, this data may be difficult to obtain and does not measure behavioral change. Behavioral change is a change in consideration or preference. Surrogate sites can be used to study behavioral change but are difficult and time-consuming to create and do a poor job of mimicking certain classes of sites. This is particularly true of social media sites where the amount of content is impossible to fully reproduce and yet this content is the defining component of the web site.

In this thesis, I present a Chrome extension that can be used for conducting high fidelity web site market research. The framework provides the opportunity to measure behavioral change and provide a natural environment almost identical to the actual sites. I detail the implementation of the extension and its use in a web-based media experiment with a sample size of 13,000. Preliminary results and learnings are discussed. Results suggest that use of the extension is feasible and is capable of producing significant changes in consumer consideration.

Thesis Supervisor: Glen L. Urban

Title: David Austin Professor in Management



## Acknowledgments

This thesis would not have been possible without the support and hard work from a great number of people.

First I would like to thank my advisor Glen Urban. He has been very kind and understanding this past year and a half. I have learned a lot from him and feel quite lucky to have had the chance to work with him. I would also like to thank professors Gui Liberali, Yakov Bart, and Catherine Tucker for their academic insights.

A huge amount of thanks goes to my fellow students Ryan Ko, Neel Hajare, and Qui Nguyen. Ryan and Neel contributed a great deal of work last year on this project in its initial stages. Thanks to Qui for being the best UROP ever. She has been a good friend and has put in a great deal of hard work into this project. It could not have been completed without her. Also, many thanks to Harvey Xiao for his insight on China related issues.

I would like to thank General Motors for sponsoring my work. In addition, I owe many thanks to Mike Hegener, Phil Keenan, Joyce Salisbury, and Tracey Sheets for all their work to make this project a successful and worthwhile endeavor. Also thanks to Brandon Benvenuti and his team at Gongos Research for their support with technical issues and making the study a reality. Finally, Patricia Hawkins has been an incredible project manager. I owe her a lot for all the key organizational work she has contributed.

Thank you to my fraternity Massachusetts Gamma Phi Delta Theta and the MIT Sport Taekwondo for keeping me sane and allowing me to relieve stress. These are two of the most incredible communities I have ever been a part of and joining them has been two of the best and most impactful decisions I have made while at MIT.

Finally, I would like to thank my incredible family. I would not be where I am today without the incredible amounts of support and love you have given me these past 23 years. I will be forever in your debt.



# Contents

<b>1</b>	<b>Introduction and Background</b>	<b>13</b>
1.1	Marketing Research . . . . .	13
1.1.1	Surrogate Sites . . . . .	14
1.1.2	Clickstream . . . . .	14
1.2	PMMA Media-Mix Experiment . . . . .	15
<b>2</b>	<b>Objectives</b>	<b>17</b>
2.1	Summary . . . . .	17
2.2	Motivation . . . . .	18
2.2.1	Technological goals . . . . .	18
<b>3</b>	<b>Methodology</b>	<b>21</b>
3.1	Market Research methodology . . . . .	21
<b>4</b>	<b>Browser Extension</b>	<b>23</b>
4.1	Browser Extension Background . . . . .	23
4.1.1	Browser Extension Basics . . . . .	23
4.1.2	Applicability to Research Goals . . . . .	24
4.1.3	Choice of Implementation Browser . . . . .	25
4.2	Chrome Extension Architecture . . . . .	27
4.2.1	Chrome Extension Basics . . . . .	28
4.2.2	Manifest Files . . . . .	28
4.2.3	Content Scripts . . . . .	29

4.2.4	Background page . . . . .	34
<b>5</b>	<b>Serving Stimuli</b>	<b>37</b>
5.1	Survey Integration . . . . .	37
5.1.1	Verification . . . . .	37
5.2	Display of Stimuli . . . . .	38
5.2.1	Consumer Research Panels . . . . .	38
5.2.2	Enforcement Issues . . . . .	38
5.2.3	Enforcement in US study . . . . .	40
5.2.4	Enforcement in China/Netherlands . . . . .	40
<b>6</b>	<b>Results</b>	<b>41</b>
6.1	US PMMA Media-Mix Experiment Results . . . . .	41
<b>7</b>	<b>Challenges and Learnings</b>	<b>45</b>
7.1	Challenges . . . . .	45
7.1.1	Web site Variability . . . . .	45
7.1.2	AJAX . . . . .	46
7.1.3	Consistent State . . . . .	49
<b>8</b>	<b>Conclusion</b>	<b>51</b>
8.1	Contributions . . . . .	51
8.2	Future Work . . . . .	51
<b>A</b>	<b>Source Code</b>	<b>55</b>

# List of Figures

4-1	Most popular Firefox extensions . . . . .	24
4-2	Summary of browser statistics from 2012 to 2013 . . . . .	27
4-3	Screenshot of Baidu with ad inserted by content script . . . . .	33



# List of Tables

6.1	Base Case - Pre-Post Consideration . . . . .	42
6.2	Sensitivity to Involvement . . . . .	43



# Chapter 1

## Introduction and Background

This chapter provides introductory information to help put the problem in context and provide the motivation for its solution. It also provides a summary and further reference material about the media-mix experiment conducted with the use of the extension.

### 1.1 Marketing Research

The definition of marketing given by the American Marketing Association is the following: [1]

Marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large.

The definition of market research given by the American Marketing Association is the following: [1]

Marketing research is the function that links the consumer, customer, and public to the marketer through information—information used to identify and define marketing opportunities and problems; generate, refine, and evaluate marketing actions; monitor marketing performance; and improve understanding of marketing as a process. Marketing research spec-

ifies the information required to address these issues, designs the method for collecting information, manages and implements the data collection process, analyzes the results, and communicates the findings and their implications.

One method of market research is called clickstream analysis. This involves analyzing the actions taken by a user on a web site. The downside of this form of analysis is that it does not measure behavioral change - a change in consumer consideration or preference. To measure behavioral change, one would conduct a controlled pre-post experiment. In the experiment, subject's preferences are measured in a pre-survey. They are then introduced to some stimuli on a web site. Finally, their preferences are measured again and compared to pre-measures. This type of experiment is often done through the use of surrogate sites. But, as section 1.1.1 details, surrogate sites suffer from a "high fidelity" experience. It is the goal of the system described in this thesis to solve this problem by providing a "high fidelity" and natural environment in which to conduct pre-post experiments measuring behavioral change.

### **1.1.1 Surrogate Sites**

Employing the use of surrogate sites is one method of conducting behavioral market research. This practice involves building a site based on a desired web site's real-life version. They can be useful when the researcher's do not have the ability to modify the actual site for experimental purposes - which is often the case. The primary issue with surrogate sites is that they are often of much lower quality than the real-life versions. The lower quality is not a reflection of the ability of the creators but is a result of the simple fact that a few programmers can not reproduce the work of an entire company in a relatively short period of time.

### **1.1.2 Clickstream**

A clickstream is a record of the actions a visitor takes on one or more web sites. It represents the user's decisions while using the site. The information collected includes

which parts of the site are visited, which actions are taken (editing a profile, purchasing an item), as well as timing information. [2] This information is best recorded in logs of servers owned by the company responsible for the web site. Unfortunately, market researchers are often not working for the company that hosts the web site they want to study. Thus, a significant problem with clickstream data is that it can be difficult to obtain. The format can also make it difficult to filter and decipher the data. [2]

## 1.2 PMMA Media-Mix Experiment

This section describes the media-mix experiment conducted using the browser extension during the summer of 2012. A total of 13,000 respondents from across the US took part in the study. PMMA (prospective meta-analysis) is a concept in which multiple distinct experiments, conducted using the same protocol and under similar conditions, merge their results and are treated as a single study. [7] This increased sample size can lead to greater statistical significance in the results and can reduce random sampling bias. In this instance, Chevrolet was interested in investigating how best to spend its advertising dollars. Currently, Chevrolet spends money on a wide variety of media from Facebook to banners on sites like Edumunds.com to TV commercials and Youtube video campaigns. The goal of this media-mix experiment was to best determine how to allocate resources among the different types of media. The study included a full factorial design on four main stimuli and a separate partial factorial design on three additional media. [7]

The advertising media in the study were organized as follows:

1. Main Stimuli
  - (a) Facebook.com - Ad in news feed
  - (b) Edumunds.com - Topmost banner ad
  - (c) Google.com - Topmost search ad result
  - (d) TV - 30 second ad spliced into a 90 second clip of a TV show

## 2. Additional Stimuli

- (a) Twitter.com - Promoted tweet
- (b) YouTube.com - Topmost sponsored ad result
- (c) Jalopnik.com - Blog post

Expansions of the same study are to be conducted in China and then the Netherlands during the summer of 2013. The main difference is in the number and specific stimuli presented. The China study will consist of a factorial study of 3 stimuli. SinaWeibo.com will replace Facebook.com, Baidu.com will replace Google.com, and the third stimulus will be a TV ad spliced between clips of a Chinese TV show. The exact stimuli to be used in the Netherlands is still under evaluation.

# Chapter 2

## Objectives

### 2.1 Summary

This section provides an overview of the contents presented in this thesis.

Chapter 3 describes the experimental methodology used for studies conducted with the Chrome extension.

Chapter 4 provides background information about browser extensions and why they are a good fit to solve this market research problem. It also details the implementation and architecture of the extension.

Chapter 5 discusses the serving of stimuli for a study using the Chrome extension. It includes issues regarding integration with the research survey such as adapting stimuli based on answers in the pre-survey and enforcement.

Chapter 6 shows results from a mix-media study conducted in the US with a sample size of 13,000. Results show that the downloading and installation of a new browser and extension are feasible at scale. In addition, preliminary analysis shows that significant changes in consumer preference can be seen.

Chapter 7 describes some of the challenges and learnings of implementing and using the extension in a real-life setting.

Chapter 8 summarizes the contributions of this thesis. It also includes future areas of research and improvement to the extension.

## 2.2 Motivation

The primary objective of this system is to build an IT framework for conducting market research capable of measuring behavioral change. The framework should create a natural environment for measuring these changes in contrast to the surrogate sites usually created. The goal is to have the users participate in settings that are as realistic as possible in the belief that this will translate to more credible results when applied to the real world.

### 2.2.1 Technological goals

This section describes the technical goals of the browser framework. The process of creating surrogate web sites does not scale well to multiple web sites. This is because adding web sites requires the creation of a completely new surrogate site for each site to be included. This creation process takes months of work and still produces sites of inferior quality when compared to real-life counterparts. Thus one of the goals of the framework is that it be extensible. Adding new sites should not require months of work. Ideally, it should be relatively easy to include another site in a study.

One of the main issues with surrogate sites is their lack of quality and depth when compared to their real-life counterparts. This is understandable as web sites such as Facebook have thousands of employees putting years of work and millions of dollars into the creation and maintenance of the web site. It is completely impracticable that a small group of technologists helping conduct market research could build something similar in the span of months or even a year. Even if it were practical, another substantial problem is the generation of content. It is impossible to create realistic content on those scales with just a few people. However, it is this content that drives these web sites and thus it is critical that a natural environment be created for the purpose of conducting market research. Thus, another goal of the framework is to have the ability to create a high-fidelity experience for research. The goal is for survey participants to be able to interact with the real-life production version of the web site being studied.

The extension must also be able to adapt the stimuli based on the pre-survey. This allows the stimuli to change based on a number of factors such as consumer preferences or demographics information. In addition, users may be assigned to different cells which see different subsets of all stimuli in the study. The extension must be able to properly display the required subset in a random order and communicate this order back to the post-survey. Finally, an enforcement mechanism is needed to prevent users from manipulating the survey and ensure they actually interact with the stimuli.



# Chapter 3

## Methodology

This chapter describes the market research methodology intended to be used in combination with the browser extension.

### 3.1 Market Research methodology

Proper, methodical marketing research studies share multiple best practices:

**Pre- and post-measures** Detailed and carefully crafted surveys are taken before and after the study. These surveys are designed to minimize biases and other methodological problems such as demand characteristics.

**Validation** Some form of validation is built into the study to ensure users are engaged and not manipulating the survey.

**Pre-test Group** A specially picked group of initial subjects is chosen to establish a baseline measurement.

**Controlled Environment** Participants are exposed to the stimuli in as controlled an environment as possible. This is important for trying to show causal results and not simply correlations.

In addition to these best practices the studies done with the Chrome extension share the following characteristics:

**Factorial experiments** Participants are assigned to a block (cell) according to the different factors in a factorial design. This helps the process of evaluating the effects and possible interactions of multiple independent variables.

**Adapt stimuli based on pre-survey** The study should be able to adjust the stimuli based on the pre-survey. For example, depending on the cell participants are assigned to in the study, they will need to see different numbers of stimuli. The particular stimuli that are shown may also be impacted by questions answered in the pre-survey. For instance, in the media-mix study users are placed into different vehicle segments (small, medium, and large) based on their response to preferences in the pre-survey.[7] That way someone who is interested in small cars does not see advertisements for large cars which they might not be interested in.

**Record clickstream** A clickstream is recorded during the study so that the level of interaction with a stimulus can be quantified. This clickstream can also be important for verification/enforcement purposes in the study.

# Chapter 4

## Browser Extension

This section gives an overview of browser extensions and the architecture of the Chrome browser extension used to conduct marketing research. Section 4.1 gives background information about browser extensions. Section 4.2 details the architecture of the Chrome browser extension.

### 4.1 Browser Extension Background

#### 4.1.1 Browser Extension Basics

A browser extension allows developers to add new functionality to a browser or alter web sites. They can be written in a variety of languages and often have little to no user interface displayed. Extensions are bundled into a single file which can be downloaded and installed easily. These extensions are becoming increasingly popular. There are extensions that provide a wide range of functionalities in categories ranging from productivity to fun to sports to shopping. A list of the most popular Firefox extensions (add-ons) can be seen in Figure 4-1.

The fact that extensions are becoming more popular with the public is important as survey participants may be less reticent to install an “untrusted” extension on their browser in order to participate.

## Most Popular Extensions

Sort by: [Featured](#) | [Most Users](#) | [Top Rated](#) | [Newest](#) | [More](#) -

 [Subscribe](#)



### Adblock Plus NO RESTART

Annoyed by adverts? Troubled by tracking? Bothered by banners? Install Adblock Plus now to regain control of the internet and change the way that you view the web.

A short video overview is available at <http://www.youtube.com/watch?v=oNvb2SjVjil>

☆☆☆☆☆ (3,679) · 15,664,967 users



### Video DownloadHelper FEATURED

The easy way to download and convert Web videos from hundreds of YouTube-like sites.

This works also for audio and picture galleries.

☆☆☆☆☆ (3,819) · 6,834,875 users



### Firebug NO RESTART

Firebug integrates with Firefox to put a wealth of development tools at your fingertips while you browse. You can edit, debug, and monitor CSS, HTML, and JavaScript live in any web page...

☆☆☆☆☆ (1,505) · 3,170,853 users



### Greasemonkey

Customize the way a web page displays or behaves, by using small bits of JavaScript.

☆☆☆☆☆ (818) · 2,247,441 users



### Easy YouTube Video Downloader NO RESTART

Fastest, easiest & most reliable free YouTube downloader. Highest sound quality available with M4A, MP3, MP4, AAC, FLV and HD formats! Single click, non-intrusive, direct download button works directly within the YouTube page!

☆☆☆☆☆ (1,244) · 2,231,191 users

Figure 4-1: Most popular Firefox extensions

### 4.1.2 Applicability to Research Goals

In the past when researchers wanted to measure the behavioral change of a website on consumers experimentally, they would have to create a surrogate site. A surrogate site is a scaled-down version of a web site that has enough features to conduct the experiment and mimics the original site enough that users will not dismiss it. Depending on the site being emulated this could take months of work. In addition, the rise in complexity in modern sites has made this process even more difficult. This is

most clearly seen in social media sites such as Facebook and Twitter. These web sites aggregate information from millions of users “on the fly” and use thousands of servers to be able to display a web page in a short amount of time. Aside from many hours exhausted building these systems and the resources spent on servers and infrastructure, the most difficult part to reproduce is the sheer quantity and individuality of the content. The rate at which content is generated is impossible to replicate. The ability to replicate content may not be an issue for web sites in general but social media sites are dependent on the content generated by their users. This, therefore, presents a significant problem for one attempting to study the effectiveness of advertising on one of these sites.

This is where browser extensions shine. One of the main advantages a browser extension provides is that users can interact with the actual version of the website being studied. For example, users can login and use their own Facebook account and see actual friends and all the actual content generated. This provides a much more realistic experience than directing them to a simulated Facebook account that does not look nearly as polished and lacks the vast amounts of content the real site provides.

Another plus to using browser extensions is that it is easier to scale to multiple web sites than using surrogate sites. When creating a new surrogate site it is likely that most of it will have to be built from scratch since modern websites vary greatly in design and functionality. With the extension architecture described in section 4.2 a significant amount of code and infrastructure can be re-used between sites. Of course, incorporating a new site into the extension will still require some work but, it will be limited to essentially adding a single new file and determining the content to be displayed.

### **4.1.3 Choice of Implementation Browser**

Unlike a normal web page a browser extension is built for a specific browser and uses that browser’s extension API. Therefore a choice had to be made about which browser to use. There are frameworks available that attempt to make it possible to

develop an extension for multiple browsers at the same time. However the quality of these frameworks was unknown and the risk of having issues and extra time from debugging in different browsers was deemed too costly. Exploring these frameworks is discussed more in section 8.2.

There were multiple criteria used to decide which browser to use. One was the usage percentage of the browser. This is important because all users who do not have the required browser will have to download and install that browser. The installation process and unfamiliarity with a new browser can lead to drop-outs which can greatly increase the cost of conducting the survey. An additional important criterion was the ease of use of the API and simplicity of implementation. Ideally, programming the extension would require minimal knowledge outside of the extension API. Another criterion used was the features of the API provided by the browser. The extension needs to:

1. Read and alter the DOM of the a web page
2. Make HTTP requests - so that data it collects can be written to a database
3. Store cross-domain state
4. Read URL - so GET parameters can be passed to the extension

Figure 4-2 is a graph detailing the usage percentage of different browsers around the world obtained from [gs.statcounter.com](http://gs.statcounter.com).

This graph clearly eliminates all browsers except Chrome, Internet Explorer, and Firefox since otherwise the incidence rates for people with the browser already installed would be too low. Chrome has the advantage here as it is the most commonly used.

With regards to simplicity of implementation Internet Explorer was the most difficult. It required learning about the Component Object Model (COM) and the features appeared limited. [8] It was not clear after reading the documentation how actions like sending HTTP requests or altering the DOM would be performed. The documentation seemed focused on features like adding menu items and explorer bars

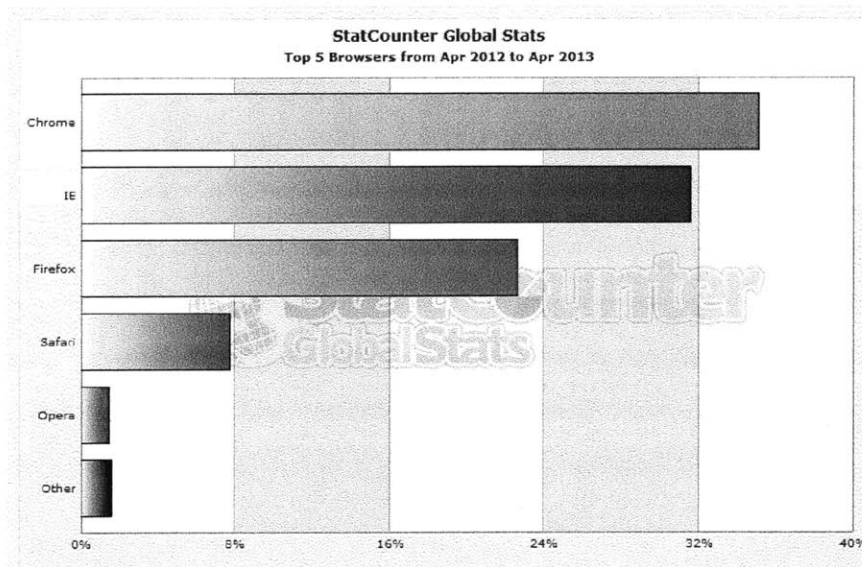


Figure 4-2: Summary of browser statistics from 2012 to 2013

which are not needed. Thus IE was discounted from the list leaving the decision between Firefox and Chrome.

The extension APIs for both Firefox and Chrome were similar in features. One downside of Firefox was that it required learning XUL (XML User Interface Language) which is Mozilla's XML-based language used for building application user interfaces. [9] Google Chrome browser extensions can be written completely in HTML, CSS, and JavaScript. This simplicity and the fact that the team members were familiar with the languages had appeal. The documentation for Chrome was also found to be understandable and well-organized into sections. Chrome also had around 2x the market share of Firefox. Therefore, the decision was made to create a browser extension for Google Chrome.

## 4.2 Chrome Extension Architecture

This section details the architecture of the Chrome browser extension.

### 4.2.1 Chrome Extension Basics

A Chrome extension is simply a zipped bundle of files (HTML, JavaScript, etc...) that add functionality to the Browser. [5] In addition they have access to the APIs that browsers give to web pages for actions such as XMLHttpRequests to HTML5 features. An extension contains the following files:

1. A manifest.json file
2. One or more HTML files
3. Any other files such as CSS or JavaScript needed by the extension to run

Most extensions have a background page which holds the main logic and state of the extension. They often also have content scripts which can interact with web pages. The content scripts and background page communicate by asynchronous message passing. In addition, extensions can save data using HTML5 storage APIs like localStorage.

### 4.2.2 Manifest Files

Every extension must have a manifest.json manifest file. It contains important metadata about the extension such as its name, version, content scripts, minimum Chrome version, and permissions. <http://developer.chrome.com/extensions/manifest.html> describes each field in detail. The most important field for purposes of this extension was the content\_scripts field. A content script was needed for each web page in the study and the content\_scripts field defined them all and mapped each one to the URLs it should run on. Here is an example snippet:

```
{  
  "name": "Survey Extension",  
  "version": "0.2.4",  
  "manifest_version": 2,
```

```

    "description": "Survey Extension",
    "icons": { "16": "icons/GIR_logo.png",
"48": "icons/GIR_logo.png",
"128": "icons/GIR_logo.png"},
    "browser_action": {
"default_icon": "icons/GIR_logo.png"
    },
    "permissions": ["tabs", "history", "http://74.207.227.126/*"],
    "background": {
"scripts": ["background.js", "js/jquery-1.7.2.min.js"]
    },
    "update_url": "http://74.207.227.126/china/updates.xml",
    "content_scripts": [
//qualtrics
{
"matches": ["*://survey.qualtrics.com/*"],
"js": ["contentscripts/survey.js"],
"run_at": "document_end"
    },

```

You can see the various fields such as name, version, and an example entry for a content\_script.

### 4.2.3 Content Scripts

Content scripts are JavaScript files that run in the context of web pages.[4] They can directly access the Document Object Model (DOM) of these web pages allowing for a great deal of power to alter the entire web page. However, they do have some limitations. Here is a list of limitations copied from the documentation site [4]

1. Use chrome.\* APIs
2. Use variables or functions defined by their extension's pages

### 3. Use variables or functions defined by web pages or by other content scripts

The good news is that some of these limitations, such as the use of variables defined by their extension's pages, can be bypassed by sending messages to their parent extension's background page.

The content scripts are run after the DOM is loaded so that the appropriate HTML element can be found and replaced. This method works well although it has a few downsides. One is that since the script must wait for the DOM to be loaded there can be a significant amount of time where the old advertisement is seen before it is replaced. This can cause users to see a flicker.

The content script is also responsible for click tracking. This is important for determining if users are engaging with the stimuli. The content scripts were created by first examining the page and determining where the ad should go. Once the location was decided, the next step was examining the HTML using the Chrome Developer Tools to determine how the site was structured. The content script would then write into the correct location with the same HTML structure and CSS classes as the original but with our content. Here is part of the content script for Baidu.com to be used in the China experiment.

```
1 /*Script for Baidu
2
3 For any search query submitted on Baidu by the user:
4 -Records the query in the database as a search type action, with the
   query
5 text as the target
6 -If text ads occur naturally for that query, then the first text ad
   is
7 changed to the desired stimuli (based on segment of user)
8 -If text ads do not occur naturally, then the script inserts the
   stimuli
9 at the top of the page
10 -Records any clicks on the inserted stimuli
11 -Any brand zone advertisements are not affected by the script
12 */
```

```

13
14 //-----Record the search query-----
15
16 var searchForm = $('form[name="f"]');
17 var searchBox = $("#kw");
18 searchForm.submit(function(event) {
19     processSearch(event);
20 });
21
22 //ensures that if the user submits the query by pressing enter in
    the
23 //text box, our listener is still called
24 searchBox.keyup(function(event) {
25     if (event.keyCode == 13) {
26         event.preventDefault();
27         searchForm.submit();
28     }
29 });
30
31 function processSearch(event) {
32     var query = searchBox.val();
33     chrome.extension.sendRequest({type: "writeSearch", stimID: "1",
        target: query});
34 }

```

baidu.js

This portion logs the search query for later analysis. The content script does so by sending a message to the background page. Here is another snippet that adds our content to the DOM and adds click tracking.

```

1     //Create ad with values
2     var newAd = "";
3     newAd = newAd.concat('<table id="4001" width="635" cellpadding="0"
        cellpadding="0" border="0" bgcolor="#f5f5f5" '
4 + 'style="table-layout:fixed;" class="EC_mr15"><tbody><tr><td
        width="90%" style="line-height:24px;" class="f16 EC_PP" id="

```

```

    taw0"><a target="_blank" style="display:block;width:100%;text-
    decoration:none;overflow:hidden;white-space:nowrap;" href="');
5  newAd = newAd.concat(adMainRowLink);
6  newAd = newAd.concat(' " id="aw0"><font clicktext="title_link" size
    ="3" style="text-decoration:underline;">');
7  newAd = newAd.concat(adMainRowTitle);
8  newAd = newAd.concat('</font><font clicktext="model_link" size
    ="-1" color="#008000" style="margin-left:6px;">');
9  newAd = newAd.concat(adLinkName);
10 newAd = newAd.concat(' </font></a></td><td width="10%" align="
    right" style="width:10%;line-height:24px;"><a target="_blank" '
11 + 'href="http://e.baidu.com/?refer=666"><font color="#666666"><
    font color="#666666"> &#25512;&#24191;&#38142;&#25509; </font>'
12 + '</font></a></td></tr><tr><td style="line-height:16px; padding:0
    0 1px 0;*line-height:20px;*padding:0;overflow:hidden;width
    :100%;white-space:nowrap;" id="taw0" class="f16 EC_PP" colspan
    ="2"><font size="-1">');
13 newAd = newAd.concat(adSecondRowText);
14 newAd = newAd.concat('</font></a></td></tr><tr><td colspan="2">');
15 newAd = newAd.concat(adThirdRowLinks);
16 newAd = newAd.concat('</td></tr></tbody></table>');
17
18 //-----Insert stimuli and add click listener
    -----
19 var container = $("#container");
20 var topTextAd = container.find("#4001"); //Top text ad always has id
    4001
21 var firstResultsTable = container.find("#content_left").find("table"
    ).filter("#1");
22
23 if (topTextAd.length == 0) { //No text ads occur naturally for this
    query,
24         //so insert stimuli before top result
25     newAd = newAd.concat("<br>");
26     var newNode = $(newAd);

```

```

27     newNode.mousedown(function(event) {chrome.extension.sendRequest({
        type: "writeClick", stimID:"1", target: event.target.
        getAttribute("clicktext")});});
28     newNode.insertBefore(firstResultsTable);
29 }
30 else { //Change top text ad to our stimuli
31     topTextAd.html(newAd);
32     newNode.mousedown(function(event) {chrome.extension.sendRequest({
        type: "writeClick", stimID:"1", target: event.target.
        getAttribute("clicktext")});});
33 }
34 });

```

baidu.js

Finally, figure 4-3 shows what baidu.com looks like after the script has run. Our ad is circled in red.

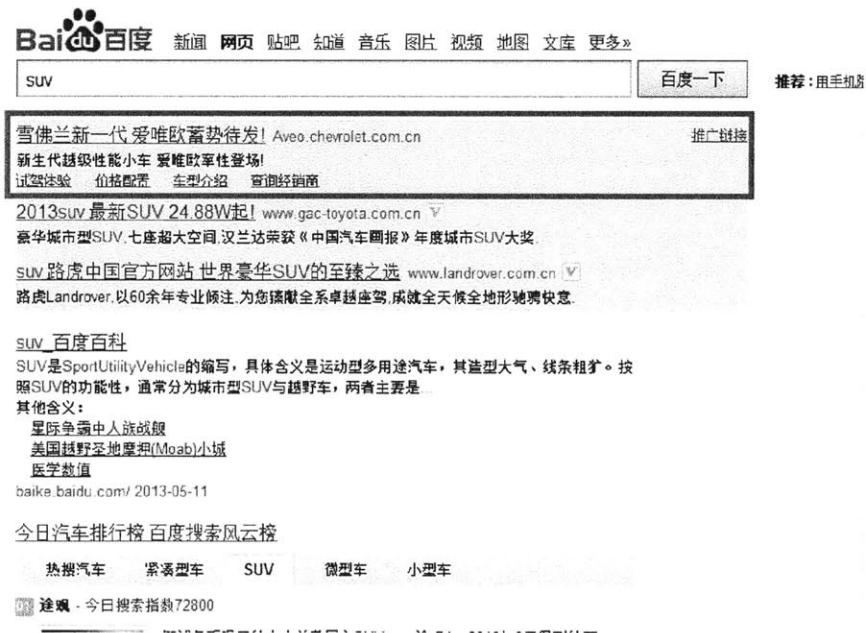


Figure 4-3: Screenshot of Baidu with ad inserted by content script

This approach worked but is poor at adapting to changes in the HTML of the page. Section 7.1.1 discusses this challenge in more detail.

## 4.2.4 Background page

The background page is the final piece of the extension puzzle. The key responsibilities of the background page are the following:

1. Store the state of survey participant (stimuli seen, time spent, etc...)
2. Send data received from content scripts to PHP server which persists information to database
3. On completion of stimuli inform survey of order stimuli were seen

The background page communicates with a PHP server. This server is responsible for persisting the information to the PMMA database. For more information about the structure of the database see Ryan Ko's thesis. [7]. Originally, most of the state was kept in the PHP session but, this solution did not work for people who completed the survey in multiple sessions. This caused most of the state to be stored in the background page in HTML 5 localStorage. For more information at keeping the state consistent see section 7.1.3. Here is an example snippet of part of the background.js which shows code for some of the features such as showing stimuli and setting user information.

```
1
2 chrome.extension.onRequest.addListener(
3   function(request, sender, sendResponse){
4     if(request.type == "showStim"){
5       var stimUrl = stimURLs[request.stimtype];
6       chrome.tabs.create({"url": stimUrl});
7     } else if(request.type == "setUserInfo"){
8       seg = request.seg;
9       userid = request.userid;
10      sendResponse({text: request.type + " successful"});
11    } else if(request.type == "getUserID") {
12      sendResponse({uid: userid});
13    } else if(request.type == "getUserSegment"){
14      sendResponse({segment: seg});
```

15

```
}
```

background.js

Above the background page is communicating with content scripts by passing messages. To communicate with the PHP the background page uses HTTP POST.



# Chapter 5

## Serving Stimuli

This chapter discusses the issue of serving stimuli with the browser extension. This involves both integration with the survey and the actual display of the stimuli. Significant challenges in serving stimuli include enforcing the requirement that the user interacts with the stimuli and preventing manipulation.

### 5.1 Survey Integration

This is an important issue because the methodology used involves adapting the stimuli based on responses in the pre-survey. Therefore, information about choices made in the pre-survey must be communicated to the Chrome extension. Otherwise the extension will not know how to adjust the stimuli.

#### 5.1.1 Verification

The flow of the experiment goes from the pre-survey to the stimuli site, powered by the extension, and then to a post-survey. Therefore it is critical that the installation of the extension be confirmed before the participant is transferred to the stimuli site. Thus, the survey must be able to verify that the extension is installed and working so that it will only transfer users if they have installed the extension and will force them to do so otherwise. Since the extension can write to the DOM of any page it has

permissions to interact with, a simple solution for verification was found. A content script for the survey site is created and writes a div with a unique id shared with the survey developers which they can easily check.

Here is an example snippet of code:

```
var div = document.createElement('div');
div.innerHTML = "1";
div.style.display = 'none';
div.setAttribute('id', 'ChromeExtensionCheck');
document.body.appendChild(div);
```

One potential error is that the extension may try to write the div before the DOM is completed and thus will be overwritten. Therefore, it is important to set the `run_at` property of the content script in the `manifest.json` of the extension equal to `document_end`.

## 5.2 Display of Stimuli

### 5.2.1 Consumer Research Panels

Since it is a difficult and time-consuming process to recruit thousands of survey participants, researchers often hire market research firms to recruit their sample. These companies have a number of panels grouped by various demographic information. The people in these panels are rewarded per study they complete. It is often the case that longer studies give greater rewards and thus are more expensive. This system gives an incentive for people to take the longer surveys as fast as possible to maximize their rewards. Therefore, some form of enforcement is required to prevent those who are trying to manipulate the system for their own benefit.

### 5.2.2 Enforcement Issues

As the study is predicated on measuring the behavioral change after interacting with the stimuli, it is key that the user actually interact with the stimuli. Therefore the

Chrome extension needs a way of enforcing involvement and/or preventing manipulation (or “cheating”). However, there is a trade-off. On one hand the system should make sure the user actually engages and spends enough time interacting with the stimuli. On the other hand, if the criterion is too stringent the user may become upset or frustrated and quit the survey if the requirements seem unreasonable. A couple of different enforcement criteria will be discussed.

One possible enforcement criterion is a minimum time limit. This approach has the benefit of being conceptually simple although it has issues in practical implementation. However, determining the time limit is difficult. If the set time is too small then cheating users may be able to get through without actually performing the desired interaction. Setting the value to be too large can also cause issues. If it is too large then users who are legitimately interacting might get frustrated if repeatedly told to stay longer. In the worst case users would drop out of the study from frustration. Aside from the issue of duration another problem with a simple time limit is that it does not measure any form of interaction. A time limit can not distinguish between someone who used the site for a minute and someone who went to the site and then proceeded to do something else unrelated for a minute.

Therefore, another possible enforcement might include a way of checking for interaction such as a minimum number of clicks. This type of enforcement method will be able to distinguish between a user actually using the page and one who just visited but is actually focused on something else. However, this method is also not without its downsides. Again, there is a question of how many clicks should be required by the user. If the number is too low then the requirement is useless - too high and it is potentially aggravating to the user. Even using the number of clicks as a criterion does not completely solve the issue of enforcing interaction since it depends on how a click is defined. Does a click anywhere within the web page count? Maybe the user is just randomly clicking but not paying attention. Should it only be clicks on the advertisement? Maybe the user has read the ad and interacted with the site but does not want to actually click on an ad. As evidenced, choosing and implementing an enforcement method is not a simple task.

### **5.2.3 Enforcement in US study**

This section will discuss the enforcement implementation used in the US media-mix experiment. A two-tiered time-based approach was implemented. If the user attempted to progress to the next stimuli in under 10 seconds they were stopped and given a message telling them to spend more time. If they tried to progress while between 10 and 30 seconds (50 seconds for TV - which fell about halfway through the ad) they were given a “soft” message. This message encouraged them to spend more time but they could progress if desired. This two-tiered approach was chosen as a compromise between making sure the users legitimately interacted with the stimuli and forcing users to interact for a frustratingly long amount of time. Out of the 14,035 people who completed the study, 5,155 of them were soft enforced at least once and 2,744 of them were hard enforced at least once. These numbers demonstrate the importance of enforcement when conducting a study.

### **5.2.4 Enforcement in China/Netherlands**

One of the results of the US media-mix study was that active involvement with stimuli was key in driving consideration. Therefore, we are changing the approach to enforcement in future studies to enforce interaction and not a time limit. For China, advancing on the Sina Weibo and Baidu stimuli will require a minimum of 3 clicks. Clicks anywhere on the web site will count due to the traditionally single-digit click-through rate of ads. In addition, it does not make sense to click on the same advertisement multiple times. The TV stimulus will still use a time based approach as enforcing a minimum number of clicks does not make sense with such a passive stimulus.

# Chapter 6

## Results

### 6.1 US PMMA Media-Mix Experiment Results

During the summer of 2012 a media-mix study was conducted in the US that included 13,000 respondents. To read a more detailed description of the experimental setup and stimuli see Ryan Ko's thesis. [7] This section details some initial analysis of the results. For more a more detailed analysis and discussion of results see the unpublished paper by Urban et. al.

The analysis involved looking at the clickstream data recorded by the extension. Analysis of clicks recorded on the actual stimuli, such as the Google ad, was used to condition by involvement with treatment. The statistical model used focused on Chevrolet models only and the pre-post consideration probability was treated as dependent. As mentioned in the background the study consisted of two factorial experiments.

Involvement for treatment was considered true for a given stimuli according to the following conditions:

**Facebook** Clicked on the Chevrolet ad in the newsfeed (sample = 578)

**Google search** Typed in a word related to automotives and clicked on the Chevrolet ad or explicitly searched for Chevrolet (sample = 261)

**Edmunds banner ad** Clicked on Chevrolet banner ad or Chevrolet page (sample = 386)

**TV** Watched the forced Chevy ad (sample = 4,018)

Consideration was calculated from 0 to 10 on a probability scale. Here are the base case results when looking at consideration.

Table 6.1: Base Case - Pre-Post Consideration

Variable	Coefficient
Intercept	.34
TV	.69
Google Search	.65
Banner	.74
Facebook	.27
Age	.01
Buy Months	.00
Income	-.01
Education	-.05

It can be seen from table 6.1 that TV, banners, and search all had significant effects on car model consideration. In fact, all media had positive effects. The amount of positive effects depended on age, purchase media, and media interactions. These significant coefficients indicate that the extension was successful in providing a high fidelity experience. This is a critical result that validates the use of this extension to conduct pre-post experiments and create behavioral change. In order to better examine the impact on involvement similar analysis was done where the levels of involvement for each stimuli are described below:

**Weaker -**

**Google Search** Some car word was searched

**Banner** 3 clicks anywhere on the page

**Facebook** Clicked on the Chevrolet ad in the newsfeed (this is same as base criterion)

**Base -**

**Google Search** Typed in a word related to automotives and clicked on the Chevrolet ad or explicitly searched for Chevrolet

**Banner** Clicked on Chevrolet banner ad or Chevrolet page

**Facebook** Clicked on the Chevrolet ad in the newsfeed

**Stronger -**

**Google Search** Typed in a word related to automotives and clicked on the Chevrolet ad

**Banner** Clicked on Chevrolet banner ad

**Facebook** Interacted with Chevrolet ad by clicking on story and voting, liking, commenting or sharing.

You can see the results in table 6.2.

Table 6.2: Sensitivity to Involvement

Variable	Base	Weaker	Stronger
Intercept	.34	.37	.35
TV	.69	.67	.71
Google Search	.65	.05	.66
Banners	.74	.00	.92
Facebook	.27	.33	.32

These results imply that involvement is key and makes a significant difference especially for search and banners. Therefore, it is important that the expansions of this study to China and the Netherlands take this information into account. Finally, it should be noted that similar results were seen for pre-post dependent variable and a post only dependent variable and pre as an independent variable.



# Chapter 7

## Challenges and Learnings

This section discusses some of challenges involved with building the extension, conducting an experiment of 13,000 people using it and the lessons learned from dealing with said challenges.

### 7.1 Challenges

#### 7.1.1 Web site Variability

Each web site that needs to be altered must have its own content script in the extension. In the case of the media-mix experiment this content script was responsible for replacing certain advertisements with advertisements of our choosing and associated logging. Today's web sites are very complex and differ greatly in implementation, most specifically with regards to the DOM hierarchy and the naming scheme of elements and other attributes like CSS classes. Some web sites even use custom HTML attributes. This can make the process of determining the exact HTML needed so that the advertisement is properly inserted difficult and involves almost no overlap between sites aside from the clicktracking infrastructure. But this is only part of the challenge. The other difficult aspect is that these modern web sites are constantly in flux and can change the structure of their HTML or CSS at any time without warning. A good example of this is Facebook. The news feed has been updated multiple

times in the past couple of years and each update brings a change in DOM layout and HTML element naming schemes. Since it will never be possible to have complete control of every web pages implementation the best way to deal with the problem is to be notified of any issues as soon as possible. When the content script detects that the expected elements do not exist a message can be sent to the background page. This background page can then send a request to the PHP server alerting it to the issue and initiating an email message detailing the page whose DOM has changed. One will then have to manually inspect the new layout to fix the content but they are at least notified as soon as possible. Another possible solution that consists of inserting content automatically is discussed further in section 8.2.

### 7.1.2 AJAX

In order to achieve better performance and responsiveness web sites are using increasing amounts of AJAX (Asynchronous JavaScript and XML). The main benefit of AJAX is that it can update parts of a web site without needing to update the entire page. For example, AJAX is the technology behind Google Instant which dynamically fetches search results and displays them without refreshing the entire page. AJAX presents a challenge for content scripts because they are run once per page load. Thus technologies like Google Instant cause difficulty because different search results and ads are shown without any page refreshes. Therefore the content script will not be able to replace the desired advertisement because it will have been run before the ad was even displayed. Even if the first ad is successfully replaced all subsequent ads will be missed. A solution would involve detecting when these AJAX updated elements change and then performing the needed action. After researching we found the following JavaScript written by Brock Adams to be useful.

```
1 /*--- waitForKeyElements(): A utility function, for Greasemonkey
   scripts,
2     that detects and handles AJAXed content.
3
4     Usage example:
```

```

5
6     waitForKeyElements (
7         "div.comments"
8         , commentCallbackFunction
9     );
10
11     //--- Page-specific function to do what we want when the
12         node is found.
13     function commentCallbackFunction (jNode) {
14         jNode.text ("This comment changed by waitForKeyElements
15             ().");
16     }
17
18     IMPORTANT: This function requires your script to have loaded
19     jQuery.
20
21     */
22     function waitForKeyElements (
23         selectorTxt, /* Required: The jQuery selector string that
24             specifies the desired element(s).
25             */
26         actionFunction, /* Required: The code to run when elements are
27             found. It is passed a jNode to the matched
28             element.
29             */
30         bWaitOnce, /* Optional: If false, will continue to scan for
31             new elements even after the first match is
32             found.
33             */
34         iframeSelector /* Optional: If set, identifies the iframe to
35             search.
36             */
37     ) {
38         var targetNodes, btargetsFound;
39
40         if (typeof iframeSelector == "undefined")
41             targetNodes = $(selectorTxt);

```

```

38     else
39         targetNodes      = $(iframeSelector).contents ()
40                               .find (selectorTxt);
41
42     if (targetNodes && targetNodes.length > 0) {
43         btargetsFound    = true;
44         /*--- Found target node(s).  Go through each and act if they
45             are new.
46         */
47         targetNodes.each ( function () {
48             var jThis      = $(this);
49             var alreadyFound = jThis.data ('alreadyFound') ||
50                 false;
51
52             if (!alreadyFound) {
53                 //--- Call the payload function.
54                 var cancelFound      = actionFunction (jThis);
55                 if (cancelFound)
56                     btargetsFound    = false;
57                 else
58                     jThis.data ('alreadyFound', true);
59             }
60         } );
61     }
62     else {
63         btargetsFound    = false;
64     }
65
66     //--- Get the timer-control variable for this selector.
67     var controlObj      = waitForKeyElements.controlObj || {};
68     var controlKey      = selectorTxt.replace (/[^\w]/g, "_");
69     var timeControl     = controlObj [controlKey];
70
71     //--- Now set or clear the timer as appropriate.
72     if (btargetsFound && bWaitOnce && timeControl) {
73         //--- The only condition where we need to clear the timer.

```

```

73     clearInterval (timeControl);
74     delete controlObj [controlKey]
75 }
76 else {
77     //--- Set a timer, if needed.
78     if ( ! timeControl) {
79         timeControl = setInterval ( function () {
80             waitForKeyElements (   selectorTxt ,
81                                   actionFunction ,
82                                   bWaitOnce ,
83                                   iframeSelector
84                                   );
85         },
86         300
87     );
88     controlObj [controlKey] = timeControl;
89 }
90 }
91 waitForKeyElements.controlObj = controlObj;
92 }

```

waitForKeyElements.js

At a high level the code works by periodically polling to the DOM to see if some set of elements have been created according to a jQuery selector. Then when a new element has been created a function is run. Thus given the correct selector for a Google or Facebook advertisement we can update the content even if the original advertisement is generated dynamically using AJAX.

### 7.1.3 Consistent State

The final big challenge in the implementation involved keeping consistent state across multiple locations. In the original implementation state was kept in the PHP server's SESSION storage and a MySQL database. The SESSION state kept track of the users current progress with study - what stimuli they had seen, what stimuli they

still needed to see, etc. When the user completed the study their information would then be persisted to the database. This approach was simple and worked well. The problem was that some users took the survey across long periods of time (hours) and multiple sessions. Thus, their old session information would become lost - causing major issues. Therefore, in the next implementation their state was moved to the extension and kept in both JavaScript objects and stored in HTML5 localStorage for cross-session persistence. localStorage has the benefit that it will persist until explicitly removed. This allows a user to complete the study in multiple sessions. The problem was that it increased the complexity of the code needed to manage state. This is because it was no longer safe to assume that either the user had not started, was in progress with all the necessary state available in memory, or had completed the study. The code now needed to distinguish between a user having not started and a user having started in a different session and thus needing to load persisted state back into JavaScript. In the future it is recommended to have state in as few places as possible. In addition, automated testing would be useful in detecting sometimes hard to catch memory race conditions.

# Chapter 8

## Conclusion

### 8.1 Contributions

In this thesis I have detailed the implementation and results from using a Chrome browser extension to conduct high-fidelity web site market research in a study of 13,000 US participants.

One key point is that using the extension to conduct market research at a large scale is feasible. It was shown that the process of downloading and installing both Chrome and the extension could be achieved with reasonable incidence rates. This bodes well for future experiments conducted with the extension. Challenges associated with implementing the extension were discussed in hopes of helping others who desire to conduct similar experiments.

In addition, the extension was successful in changing consumer consideration without introducing bias. This shows the benefits of providing a natural environment for conducting market research. It is my hope that the extension will be used to conduct more experiments in the future.

### 8.2 Future Work

Although one study has been completed and results suggest the use of the extension is feasible and produces usable results, there are still many areas for possible

improvement. Currently the extension can only be used with the Chrome browser, which forces all users who do not have Chrome to download and install it. This might be too much effort for some respondents which may cause them to drop out. One possible solution to this issue is to use a browser extension framework. The Kango cross-browser framework claims benefits such as: [6]

1. Single code base for all major browsers
2. JavaScript is only language needed
3. Powerful API for common tasks

If these claims are indeed true incidence rates would likely increase in the survey since users would be able to use whatever browser they currently have installed - although they may need to upgrade their browser version depending on potential compatibility issues.

One of the most difficult parts to implement in the extension are the content scripts which alter the content of specific web pages. A significant difficulty comes from the fact that the individual DOM of each website must be analyzed to determine the exact HTML elements and IDs that need to be replicated to insert our version of the ad. In addition, modern web sites are subject to change at any time which may result in the breaking of content scripts if their layout changes. Therefore, a way of automatically inserting content into a web site could solve these problems. The specific web site and content would have to be specified but then the extension would automatically determine where to insert the content. It would potentially be similar to the popular extension AdBlock which claims to block ads for any website. [3] However, instead of hiding them it would replace their contents and add clicktracking and other types of logging. The main difficulty in this approach arises from the diversity in web site implementation - specifically with regards to the DOM structure and naming of element IDs and CSS classes.

Finally, the pre-test, stimuli based on pre-test, and post-test methodology could be utilized in a number of studies apart from the media-mix experiment described

earlier. The effectiveness of specific web sites could be examined in greater detail. For example, Facebook has advertisements in multiple locations but which are the most effective at reaching their audience? How do differences in web site design or performance impact consumer preference or consideration? It is my hope that this extension can be used in a number of studies for years to come.



# Appendix A

## Source Code

Here is the full background.js. The rest of the code is available on request.

```
1 //general handler for all messages targeted at the Chrome Extension,
   sent by content script
2
3 var stimURLs = {
4   "Weibo": "http://www.weibo.com",
5   "TV" : "http://74.207.227.126/china/tv/tv.html",
6   "Search": "http://www.baidu.com",
7   "Control": "http://74.207.227.126/china/c/b.html"
8 };
9
10 //User data
11 var seg = -1;
12 var userid = -1;
13
14 //For each stimuli, the time they started, for enforcement
15 var start = -1;
16 var end = -1;
17
18 chrome.extension.onRequest.addListener(
19   function(request, sender, sendResponse){
20     if(request.type == "showStim"){
21       var stimUrl = stimURLs[request.stimtype];
```

```

22     chrome.tabs.create({"url": stimUrl});
23 } else if(request.type == "setUserInfo"){
24     seg = request.seg;
25     userid = request.userid;
26     sendResponse({text: request.type + " successful"});
27 } else if(request.type == "getUserID") {
28     sendResponse({uid: userid});
29 } else if(request.type == "getUserSegment"){
30     sendResponse({segment: seg});
31 }
32 else if(request.type == "writeBrowsingHistory"){
33     var startbrowse = request.startTime;
34     var endbrowse = request.endTime;
35
36     chrome.history.search({text: "", startTime: startbrowse,
37                           endTime: endbrowse}, function(results){
38         //loop through backwards so that entries are recorded in
39         //chronological order
40         for (var i = results.length-1; i>=0; i--){
41             $.post('http://74.207.227.126/china/writetodb.php', {
42                 requestType: "browsingHistory", url: results[i].url,
43                 stimID: request.stimID, time: results[i].lastVisitTime,
44                 userID: request.userID});
45         }
46     });
47 } else if(request.type == "recordTimes") {
48     start = request.start;
49     end = request.end;
50     sendResponse({text: request.type + " successful"});
51 } else if(request.type == "getTimes") {
52     sendResponse({startTime: start, endTime: end});
53 } else if(request.type == "updateStop") {
54     $.post('http://74.207.227.126/china/writetodb.php', {
55         requestType: "updateStop", stimID: request.stimID, userID:
56         request.userID}) //why is this from request?
57 } else if(request.type == "writeClick") {

```

```
51     $.post('http://74.207.227.126/china/writetodb.php', {
        requestType: "action", actionType: "click", target: request
        .target, stimID: request.stimID, userID: userid});
52 } else if(request.type == "writeSearch") {
53     $.post('http://74.207.227.126/china/writetodb.php', {
        requestType: "action", actionType: "search", target:
        request.target, stimID: request.stimID, userID: userid});
54     }
55 });
```

background.js



# Bibliography

- [1] American Marketing Association. Definition of marketing. <http://kangoextensions.com/>, May 2013.
- [2] Randolph E Bucklin, James M Lattin, Asim Ansari, Sunil Gupta, David Bell, Eloise Coupey, John DC Little, Carl Mela, Alan Montgomery, and Joel Steckel. Choice and the internet: From clickstream to research stream. *Marketing Letters*, 13(3):245–258, 2002.
- [3] Adblock Developers. Adblock. <https://chrome.google.com/webstore/detail/adblock/gighmmpiobklfepjocnamgkkbiglidom?hl=en>, May 2013.
- [4] Google. Content scripts. [http://developer.chrome.com/extensions/content\\_scripts.html](http://developer.chrome.com/extensions/content_scripts.html), May 2013.
- [5] Google. Overview. <http://developer.chrome.com/extensions/overview.html>, May 2013.
- [6] KangoExtensions. About. <http://www.marketingpower.com/AboutAMA/Pages/DefinitionofMarketing.aspx>, May 2013.
- [7] Ryan Ko. Prospective marketing meta-analysis and a novel web-based media-mix modeling experiment. Master’s thesis, MIT, 2012.
- [8] Microsoft. Browser extensions. <http://msdn.microsoft.com/en-us/library/aa753587%28v=vs.85%29.aspx>, May 2013.
- [9] Mozilla. Xul. <https://developer.mozilla.org/en-US/docs/XUL>, May 2013.