FlashDOM: Interacting with Flash Content from the Document Object Model

Kyle I. Murray
Department of Computer Science
University of Rochester
Rochester, NY 14627 USA
kyle.murray@rochester.edu

ABSTRACT

Assistive technologies are often only adapted to emerging web technologies after they are common enough to generate demand for relevant assistive technology. This paper proposes a general approach to speed up and simplify the process of making assistive technologies compatible with innovation on the web by providing hooks to the browser's standard Document Object Model that expose the capabilities of a new technology.

In this paper, we will illustrate the utility of this model by applying it to Adobe Flash content, which, despite years of attempts to produce more accessible content, remains woefully inaccessible.

We show how this approach enables screen readers that would normally not have permission to read Flash content to access this content, and how it can enable interactivity with Flash content in a modern browser.

Categories and Subject Descriptors

K.4.2 **[Social Issues]**: Assistive technologies for persons with disabilities; H.5.2 **[Information Interfaces and Presentation]**: User Interfaces

General Terms

Design, Human Factors, Standardization

Keywords

Accessibility, assistive technology, Flash, screen reader, document object model

1. INTRODUCTION

Adobe Flash Player has achieved near-ubiquity on the web as a plugin that enables various types of multimedia in the browser [2]. Despite the prevalence of open formats on the web and the existence of an open specification for the SWF format that Flash Player reads, content developed in Flash is often entirely inaccessible to assistive technologies that seek to convey information in a form that is different from the form that the content author intended. Because the source code to Flash Player is not available to the public, and because there are no fully-featured alternatives to the player, the implementation of facilities for assistive technologies to hook into are limited to those that Adobe includes in the player.

FlashDOM is an implementation of our approach to exposing Flash content to the browser in a way that allows extant assistive technologies to interface with Flash content in the same way that they interact with web formats such as HTML and the Document Object Model that is exposed by JavaScript and the browser.

We built FlashDOM into the web-based screen reader WebAnywhere [4]. When a user visits a page with Flash content, the web proxy that sits between WebAnywhere and the web detects that Flash content is being loaded. Instead of serving the exact file that is requested, FlashDOM returns a wrapper file that subsequently loads the original file.

The wrapper enables FlashDOM to extract information about the structure and interactive features of the Flash content that are sent to the HTML host page as DOM elements, just like the DOM elements that appear throughout the rest of the web.

2. RELATED WORK

2.1 Accessibility Evaluators

Saito *et al.* [5] used a similar Flash-JavaScript bridge strategy to extract information from running Flash content in order to perform accessibility evaluations on websites. The data collected was transformed into XML that would be presented to an evaluator. Although the extraction approach is similar, the data was intended for evaluation and not as information for standard web components that could be utilized by assistive technologies. Consequently, their evaluator could not be used by users of a screen reader in a situation where the user wanted to have Flash content read in the same manner as an HTML-based page.

3. IMPLEMENTATION

3.1 DOM Extractor

The proxy that WebAnywhere uses to mirror web pages to the user was modified to load all Flash files via a FlashDOM-aware wrapper. The purpose of using the wrapper is that extra code is included that explores the structure of the original content and transmits that structure, via JavaScript, to the page being read by WebAnywhere. Previously, Flash content was inaccessible to WebAnywhere and available to a very limited extent to other forms of assistive technologies.

The FlashDOM extractor supports many of the user interface elements from the HTML DOM that have analogues in Flash Player in addition to simply extracting text. Specifically, it extracts text input fields, clickable regions like buttons, and structured lists of elements. After user input, the extractor updates the HTML DOM to include the newly-updated Flash content.

The structure of the Flash content that is sent to the DOM is not simply static information. Because the event model used by Flash Player is essentially the same as the DOM event model, interactions like keyboard strokes, mouse pointer clicks, and link activations can be passed back and forth between the HTML structure generated by FlashDOM and the Flash plugin instance. As a result, typing in an HTML text field will immediately be reflected in the Flash movie, and pressing a button can trigger Flash to submit data collected in a form.

3.2 Bytecode Analyzer

The bytecode analyzer is a tool that parses the SWF file and the ActionScript bytecode [1] inside of it that was generated by a content developer writing ActionScript code. Once parsed, the bytecode is represented as a class-level abstraction that is manipulated by the wrapper. Its primary purpose is to analyze the bytecode sequences for instructions that make so-called URLRequests, or any sort of network communication that would bypass the WebAnywhere proxy. Any offending bytecode sequences are rewritten in such a way that the URLs that are on the runtime stack are passed through a proxy function that maps the original URL to one that WebAnywhere proxies.

The changes to the bytecode are written back into an abstract representation of an SWF file [3] that is loaded into the running wrapper that can communicate the actual structure of the Flash content to the web page. Effectively, these modifications allow Flash content that relies on relative URLs to run without breaking in an environment mirrored by a proxy. Furthermore, the generic bytecode abstraction developed for FlashDOM allows for future changes to be made to the bytecode modifier.

4. DISCUSSION

FlashDOM provides a generic way to allow a Flash movie to expose its information and interactivity in a way that allows existing assistive technologies made for accessing traditional web content like HTML to access information that was previously difficult to access given the closed nature of Flash Player.

In actual usage of our implementation, there are capabilities of Flash that are not adequately replicated or areas that are not replicated at all, such as the streaming videos and communications over web sockets. These are limitations of our design and of the extant proxy in WebAnywhere. These limitations made user-testing infeasible in FlashDOM’s first iteration, since a few of the most popular Flash-based sites tend to use technologies like video streaming which requires a more complex proxy than the one used by WebAnywhere.

Due to the complexity of the ActionScript bytecode specification [1] and the added complexity of merging that bytecode with the SWF format [3], our implementation of FlashDOM experiences some level of instability when faced with the huge variety of Flash content on the internet. Particularly, legacy scripts written for versions of Flash Player prior to version 9 are not considered by the bytecode analyzer. Although these instability problems are not inherent to our approach, their appearance in our implementation diminishes its current utility.

5. FUTURE WORK

The prevalence of assistive technologies for the HTML- and DOM- based web suggests that reflecting information about Flash as part of that DOM would have more applications than integration with WebAnywhere. Particularly, a browser extension that provided FlashDOM’s functionality would reduce the dependency on WebAnywhere for people who use other assistive technologies or for automatic data-mining of Flash applications at runtime.

Additionally, more robust support for all of the capabilities that Flash provides would increase the utility of FlashDOM to sites with heavy streaming media or complex, socket-based server communication. The addition of these features would bring FlashDOM to the point where it would be practical to conduct tests with daily screen reader users.

6. REFERENCES

[1] Adobe Systems, Inc. 2007. ActionScript Virtual Machine 2 (AVM2) Overview. http://www.adobe.com/devnet/actionscript/articles/avm2overview.pdf.

[2] Adobe Systems, Inc. 2009. Adobe Flash Player PC Penetration. http://www.adobe.com/products/player_census/flashplayer/PC.html

[3] Adobe Systems, Inc. 2008. SWF File Format Specification Version 10. http://www.adobe.com/devnet/swf/pdf/swf_file_format_spec_v10.pdf.

[4] Jeffrey P. Bigham, Craig M. Prince and Richard E. Ladner. 2008. WebAnywhere: A Screen Reader On-the-Go. *In Proceedings of the International Cross-Disciplinary Conference on Web Accessibility* (W4A 2008), Beijing, China.

[5] Shin Saito, Hironobu Takagi, Chieko Asakawa. 2006. Transforming Flash to XML for Accessibility Evaluations, *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, October 23-25, 2006, Portland, Oregon, USA. DOI= http://doi.acm.org/10.1145/1168987.1169015