Design and Implementation of a Data Filtering and Sorting Interface in React

(Technical Paper)

The Death of Flash

(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree

Bachelor of Science, School of Engineering

Calvin Kuo

Fall 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _____ Date _____

Calvin Kuo

Approved _____ Date _____

Briana Morrison, Associate Professor of Computer Science, Department of Computer Science

Approved _____ Date _____

Joshua Earle, Associate Professor of STS, Department of Engineering and Society

Introduction

For my technical project, I will be producing a report on what I did during an internship this past summer. Along with my internship team, our goal was to design and implement a user interface using the React framework that would allow users without a technical background to filter and sort the data that the app collected. I plan to document the design decisions made during the project, as well as how it evolved in response to feedback from users. This will serve to systematize the knowledge I gained during this project, as well as allow me to reflect on how it could have been improved.

For my STS project, I will be investigating the demise of Adobe Flash over the past two decades. In its heyday, many of the top websites used Flash to create interactive websites; now, Flash content can't even be viewed in modern browsers. I will be focusing on the societal factors and network effects that led to Flash falling out of fashion, especially when compared to other web technologies such as JavaScript that remain in use today. Its death serves as a case study for how a dominant technology can lose its grip and be overthrown, even by a replacement that serves a similar purpose.

These projects are connected by the common thread of how to design an interactive website. Had Flash not fallen out of favor, it's quite possible that I would have been writing a Flash app during my internship instead.

In this prospectus, I will be describing my technical project and my STS project in turn.

Technical Project

Design and Implementation of a Data Filtering and Sorting Interface in React

The Neonatal Antibiotic Stewardship at the University of Virginia provides a web form to help clinicians prescribe the appropriate dosage and type of antibiotics. Although the system collects the inputted data into a MongoDB database, there was no user-friendly way to access the data. Using an agile development methodology, my internship team designed and implemented a user interface with the React framework to display this data. This enabled the user to specify an arbitrary number of filters and sorts, allowing a non-technical user to easily view the desired subset of the data. In addition, we also designed the interface to allow later extension and customization with additional features and visualizations of the relevant data.

Antibiotics have saved millions of lives since the discovery of the first antibiotic, penicillin, almost a century ago. They work by killing bacteria, which prevents infections from progressing and causing further illness. However, in recent decades, biologists have found an increasing number of strains of bacteria have evolved so that they can survive antibiotic treatment. This phenomenon is known as antibiotic resistance.

To prevent this development from undoing the past century of medical advances, clinicians need to use a treatment that meets two criteria. First, it should be a narrow-spectrum antibiotic, which affects only the type of bacteria at issue. This minimizes the collateral damage caused by killing good bacteria, such as the ones that aid digestion in the gut. Second, the dosing regimen should ensure that all of the bacteria of interest are killed. Otherwise, the bacteria which survive treatment may continue to multiply, causing an infection resistant to the previous treatment used.

Under the supervision of David Kaufman, MD, of the University of Virginia School of Medicine, the Neonatal Antibiotic Stewardship and the University of Virginia Development Hub

developed the Clinical Decision Support Tool, a web form which helps neonatal clinicians determine which course of antibiotics would be the most appropriate based on a number of factors.

The primary technologies used in both the Clinical Decision Support Tool and the Data Trends app was the React framework and the MongoDB data platform. React is an open-source frontend JavaScript library, originally developed by Facebook, that allows developers to build user interfaces out of reusable components. MongoDB is a JavaScript-based document-oriented web platform that served as the database backend for this project. JavaScript is a programming language that runs in browsers and is what makes many websites interactive.

STS Project

The Death of Flash

Research Question

The technology I am examining is Flash Player, which was a software platform that allowed developers to create interactive websites before its discontinuation at the end of 2020 ("Adobe Flash Player EOL", 2021). It was primarily developed by Macromedia prior to their acquisition by Adobe in 2005. My research question focuses on how Flash came to die, despite the first-mover advantage and dominance it had on the Web. Specifically, I would like to investigate what problems and shortcomings Flash had, and how replacement technologies such as HTML5[1], CSS[2], JavaScript[3], SVG[4], and WebGL[5] became widespread and ultimately replaced most uses of Flash.

---

[1] HTML5 is the latest version of HTML (Hypertext Markup Language), which describes the content of a web page. It introduces new features such as videos and canvases.
[2] Cascading Style Sheets (CSS) describes the layout and styling of elements with a web page, such as colors, borders, backgrounds, and positioning.
[3] JavaScript (JS) is a programming language that runs in web browsers and is what makes many websites interactive.
[4] Scalable Vector Graphics (SVG) is a file format which stores images as lines and shapes, rather than pixels.
[5] WebGL (Web Graphic Library) is a way for web pages to take advantage of 2D and 3D graphics rendering.

This topic is important because it will teach us more about how to distinguish between technologies that have so far had staying power (IP[6], HTTP[7], HTML/CSS/JS) versus those that have fallen into obscurity (ActiveX[8], Java applets[9], IRC[10], Usenet[11]). What caused browsers to implement replacements for Flash? What motivated companies to migrate Flash web pages to newer technologies? Were there any missteps on Adobe's part that led to Flash being superseded? And was it merely a matter of fashion, or were there fundamental problems with Flash that were later exposed?

Relevant Social Groups

The relevant social groups are developers and users of web applications on desktop, laptop, phone, and tablet devices on the Windows, Mac, iOS, and Android platforms using the Internet Explorer, Mozilla Firefox, Google Chrome, and Safari browsers. These are the major device categories, operating systems, and web browsers during this timeframe (Statcounter, 2022). I plan to focus primarily on the United States and other Western countries, especially English-speaking ones. Because the major software developers for all of these are based on the West Coast, many of the decisions they made can be considered almost entirely within the U.S. domestic market. As the United States technology sector has an outsize influence on the rest of the world, foreign users would be impacted as well, though it is doubtful they had nearly as much of an impact on the

[6] The Internet Protocol (IP) defines the rules for transmitting data across the internet. Almost all applications that use the internet ultimately rely on IP at some level.
[7] The Hypertext Transfer Protocol (HTTP) is an application of the internet that focuses on transferring documents that link to each other.
[8] ActiveX was a software framework developed by Microsoft that ran in the Internet Explorer browser. It was discontinued in 2013.
[9] Java applets were a way to run Java code within a web browser using a plugin. They can no longer be run in web browsers as of 2017.
[10] Internet Relay Chat (IRC) is an instant messaging system. It has declined in popularity with the rise of alternatives such as AOL Instant Messenger (AIM), Skype, Facebook Messenger, Slack, Discord, and other social media.
[11] Usenet is a distributed discussion system where users can subscribe to newsgroups and read and post messages. It has declined in popularity with the rise of bulletin board systems (BBS), forums, Reddit, and other social media.

decisions made by West Coast developers. It has been comparatively rare for foreign internet-related companies to gain a significant foothold outside of their own country. Additionally, the United States and other developed countries have had significant internet market penetration both during the heyday of Flash and today, allowing for a meaningful comparison between the two timeframes.

For the most part, I will not be considering the actions and motivations of end users directly. Instead, I plan to primarily view their actions through the lens of developers, divided into three major groups. The first group is the people who worked on the web browsers (Internet Explorer, Mozilla Firefox, Google Chrome, and Safari) and the operating systems (Windows, Mac, iOS, and Android). The second group is the developers who were creating the websites that people visited, primarily in the commercial, enterprise, and entertainment spheres. I will most likely focus on the roles of larger websites such Facebook, Amazon, Reddit, and Wikipedia, rather than smaller websites such as individual businesses or internal software. The third and final group is the developers at Adobe, who were in charge of maintaining the Adobe Flash Player browser plug-in. Note that some actors fill multiple roles, such as Google, which serves as both a web browser developer with Chrome and a website developer with google.com.

One of the major groups that is being left out is internet users and developers in China. Due to the Great Firewall and restrictions on foreign companies, the internet landscape there is walled off from the rest of the world. For instance, Google and Facebook are among the top websites in nearly every country in the world, but many of thes top websites internationally have limited to no presence in China (Su and Shyong, 2019). As the Chinese internet has taken such a different path from the rest of the internet, it would certainly not be appropriate to generalize these trends directly to the Chinese market.

Methods/Frameworks

I will primarily be utilizing the Social Construction of Technology (SCOT) framework and examining previous literature and primary sources. By 2008, games, animation, interactivity and video on the web appeared to be a solved problem: use Flash Player. But instead, the launch of the iPhone appears to have marked the beginning of Flash's decline, and soon different technologies came to take its place. This is a prime example of how an apparent state of technological closure can be upended, especially when you start thinking about the interests and motivations of the pertinent actors. With internet technologies, there is also a significant chicken-and-egg problem where new technologies have to become available on a critical mass of devices before general developers can take advantage of them, so I'm interested in investigating the process of exactly how that occurred in this case.

Key Texts

One of my major sources is *Flash: Building the Interactive Web*, a book by Anastasia Salter and John Murray. It dives into the history of Flash, from its rise to its decline, mainly through the lens of video games and interactive art (2014).

An important primary source is "Thoughts on Flash" by Steve Jobs. This post explains why Apple did not include Flash support on iOS devices, which was one of the major contributors to why web developers abandoning Flash (Jobs, 2010).

This WIRED article covers the history of Flash from its rise to its demise, with Apple's iPhone playing an integral role (Bedingfield, 2019).

These are some contemporaneous articles discussing transitioning websites from Flash to HTML5. The first article by Dave Meeker is targeted towards a more technical audience (2012). The second article is targeted more towards people from the marketing side of a company, to try and convince them of the value in migrating (Pasco, 2016).

This is a 2016 case study detailing the conversion process of a web site from Flash to HTML5. This will serve to illustrate the possible costs and benefits of migrating a web site, with this one prominently featuring Flash games (Santally, et al., 2016).

These two papers by Maheshwari and Reddy detail automated conversion of pre-existing Flash content into HTML5. Understanding the challenges here will shed light on why this may or may not have been able to be deployed at a wide scale, as well as the differences in capabilities between the two technologies (2015; 2017).

A paper by Vannoy and Palvia attempts to describe how technology is adopted across social networks, as opposed to the traditional organization/firm-centered approach (2010). This will be a particularly useful framework for examining the social or viral adoption of Flash, as well as those same forces that supported its replacements.

References

Adobe Flash Player EOL General Information Page. Adobe. (2021, January 13). Retrieved October 11, 2022, from https://www.adobe.com/products/flashplayer/end-of-life.html

Bedingfield, W. (2019, September 18). *The rise and fall of Flash, the annoying plugin that shaped the modern web*. WIRED UK. Retrieved October 27, 2022, from https://www.wired.co.uk/article/history-of-macromedia-flash

Jobs, S. (2010, April). *Thoughts on Flash*. Apple Inc. Retrieved May 1, 2010, from https://web.archive.org/web/20100501010616/http://www.apple.com/hotnews/thoughts-on-flash/

Maheshwari, Y., & Reddy, Y. R. (2015). Transformation of flash files to HTML5 and JavaScript. *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*. https://doi.org/10.1145/2811681.2811686

Maheshwari, Y., & Reddy, Y. R. (2017). A study on migrating flash files to HTML5/javascript. *Proceedings of the 10th Innovations in Software Engineering Conference*. https://doi.org/10.1145/3021460.3021472

Meeker, D. (2012, September 27). *Make the Move to HTML5? Everything You Need to Know to Have the Conversation*. Dave Meeker. Retrieved October 11, 2022, from https://davemeeker.com/website-magazine-article-the-flash-to-html5-evolution/

Pasco, M. (2016, January 14). *Flash Forward: The Future Is HTML5*. Website Magazine. Retrieved October 11, 2022, from https://www.websitemagazine.com/blog/flash-forward-the-future-is-html5

Salter, A., Murray, J., & Ebook Central - Academic Complete (2014). *Flash: Building the Interactive Web*. Cambridge, Massachusetts: The MIT Press.

Santally, M. I., Sungkur, R. K., Sooltangos, N., Mounier, P., & Mira-Bahadoor, R. (2016). Flash to HTML5 adaptations: A case-study for uptoten.com. *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. https://doi.org/10.1109/icacce.2016.8073778

Statcounter Global Stats. StatCounter. (2022, September). Retrieved October 11, 2022, from https://gs.statcounter.com/

Su, A., & Shyong, F. (2019, June 3). *The Chinese and non-Chinese internet are two worlds. here's what it's like to use both*. Los Angeles Times. Retrieved October 11, 2022, from https://www.latimes.com/world/la-fg-us-china-internet-split-20190603-story.html

Vannoy, S. A., & Palvia, P. (2010). The social influence model of technology adoption. *Communications of the ACM*, 53(6), 149–153. https://doi.org/10.1145/1743546.1743585