

HTML5 vs. Adobe Flash: From the End User Standpoint

by

Alexandra Jiang

S.B EECS MIT 2011

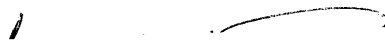
ARCHIVES


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

May 2012

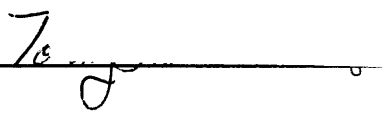
©2012 Massachusetts Institute of Technology
All rights reserved.

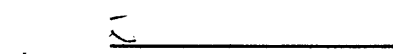
Author:





Department of Electrical Engineering and Computer Science
May 24, 2012


Certified
by:





Tommy Li
User Experience Architect, NetApp
VI-A Company Thesis Supervisor
May 24, 2012

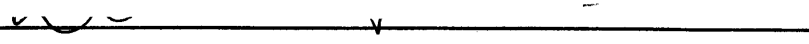
Certified
by:




Hal Abelson
Professor, Department of Electrical Engineering and Computer Science
MIT Thesis Supervisor
May 24, 2012

Accepted
by:




Dennis M. Freeman
Professor, Department of Electrical Engineering and Computer Science
Chairman, Masters of Engineering Thesis Committee


Prof. Dennis M. Freeman
Department of Electrical Engineering and Computer Science
Room 38-476, M.I.T.
Cambridge, MA 02139

Subject: Master of Engineering Thesis of Alexandra Jiang

Dear Professor Freeman:

I have reviewed the attached thesis of Alexandra Jiang on behalf of NetApp. The thesis is within the scope of the thesis proposal as previously approved and does not contain any material that is objectionable to NetApp. It is also approved for its technical content.

It is understood that the actual thesis document will be the permanent property of M.I.T. and will be placed in the M.I.T. Library within one month after the date of submission. NetApp agrees that M.I.T. shall have the nonexclusive right to reproduce, publish, and distribute the thesis.



Authorized Official of Company

HTML5 vs. Adobe Flash: From the End User Standpoint

by
Alexandra Jiang

Submitted to the
Department of Electrical Engineering and Computer Science

May 24, 2012

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

For NetApp products to be the best in the market, NetApp has to make the correct technology decisions to insure it leads in innovation. In this thesis, I focus on a product called NetApp Unified Manager, examining whether HTML5 satisfies its needs. I evaluate NetApp from a user's perspective using four criteria: data visualization and customization, platform, NetApp's special needs, and interoperability. From researching about these criteria, I evaluate the best decision for NetApp OnCommand's user experience, moving forward.

I compare Adobe Flash based and HTML5 based data visualizations and reach several conclusions: their functionalities are comparable; HTML5 is completely capable of representing large amounts of data; HTML5 does not have all the features that Adobe Flash has on all browsers, but its disadvantages will decrease over time. Additionally, I assess NetApp's inclination toward mobile development and decide that running an application native to the browser is best because mobile application development is in the future. Accordingly, I recommend that NetApp migrate their applications to HTML5.

I have made three main contributions in this thesis: I have framed a method for researching for and writing a recommendation. I have identified four criteria on which to assess NetApp's needs, emphasizing data visualizations. Finally, I have argued that, for NetApp, HTML5 is superior to Flash in the advantages it brings now.

MIT Thesis Supervisor: Hal Abelson
Professor, Department of Electrical Engineering and Computer Science

Company Thesis Supervisor: Tommy Li
User Experience Architect, NetApp

Acknowledgements

I would like to thank my group at NetApp for their support and expertise. Special thanks to John Bagby, former head of the User Experience group, and Tommy Li, also a member of our group. The two of them meet with me often to iron the details of my thesis and to throw ideas around. Another thanks to Tommy Li, NetApp thesis adviser, for reviewing my thesis and giving me suggestions during a busy time. Thank you to Brian Hackworth, NetApp Technical Director, for reviewing my thesis proposal and making sure that it contains none of NetApp's secrets. Thank you to Subhashni and Victoria for their friendships and good company. Thank you to Nancy, my dear supportive friend. Thanks also to Darren Wong for being there for entertainment and for teaching Jess, Nancy, and I self-defense techniques when we needed a break from work.

Thank you to Hal Abelson, my MIT thesis advisor, for his recommendations and discussing my thesis proposal with me. He was incredibly patient with me after I neglected to remain in contact with him. His turnaround time for my thesis was unbelievably fast. Thank you also to Patrick Winston for reviewing my abstract and giving me advice.

I am truly thankful to Steve Miller for forging the VI-A relationship between MIT and NetApp. I've had a great time and have learned a lot about working in a company from this program.

Thanks to Kathy Sullivan, Anne Hunter, Linda Sullivan, and Vera Sayzew for their support and reminders throughout my undergraduate and graduate careers.

Finally, thank you to my family for the continuing support and encouragement.

Table of Contents

1. Introduction.....	8
1.1. Adobe Flash.....	8
1.2. HTML5.....	9
1.3. Move towards HTML5.....	10
2. NetApp.....	10
2.1. NetApp now.....	10
2.2. OnCommand Unified Manager.....	11
2.3. NetApp Web Framework.....	12
3. Constructing a recommendation.....	13
3.1. Learning from a consultant.....	13
3.2. Learning from a student consultant.....	14
3.3. Investigating a similar decision - Scratch.....	15
3.4. Summary.....	17
4. Approach.....	18
4.1. Development efficiencies.....	18
4.2. Business drivers.....	19
4.3. End-user satisfaction.....	20
4.4. Summary.....	20
5. Criteria for Evaluation.....	20
5.1. Data visualization capabilities.....	21
5.2. End-user customization.....	22
5.3. Platform.....	23
5.4. Special Needs.....	23

5.5. Interoperability	24
5.6. Summary.....	24
6. Data Visualization and Customization.....	25
6.1. Fusion Charts.....	26
6.2. Highcharts.....	27
6.3. Summary.....	30
7. Platform Support.....	30
7.1. Browser.....	30
7.2. Benchmarking.....	32
7.3. Mobile.....	33
7.4. Summary.....	35
8. NetApp's Special Needs	35
8.1. Speed	35
8.2. Large Datasets	37
8.3. Summary.....	38
9. Interoperability.....	39
9.1. Microsoft and VMware Plugins	39
9.2. Summary.....	39
10. Recommendation	40
11. Contributions	42
12. References.....	43
13. Appendix.....	46

List of Figures

Figure 1. Dashboard of OnCommand Unified Manager	12
Figure 2. Example screen of a Scratch Project	16
Figure 3. An example of a treemap, a visual representation of how storage is divided.	22
Figure 4. Comparison of Flash and HTML5 FusionCharts	27
Figure 5. Example of master detail, pivot model, and multiple graphs.	28
Figure 6. Example of master detail and multiple chart types.	28
Figure 7. Example of drilling down.	29
Figure 8. Similar chart to bottom-left chart in Figure 1 using Highcharts.	30
Figure 9. Progression of HTML5 from 2008 to 2011.....	31
Figure 10. Interactive Feature support for Flash, HTML5, and HTML5 on mobile devices	32
Figure 11. Website used to measure speed of loading.....	36
Figure 12. Yahoo mail data visualization	38

List of Tables

Table 1. Animation frame rates from January 2012	33
Table 2. Animation frame rates from March 2011	33
Table 3. Comparison of website loading speeds.....	36
Table 4. Summary of Recommendation Criteria.....	40

1. Introduction

The on-going debate between Flash and HTML5 has lasted over five years. The debate was propelled by Apple's CEO, Steve Jobs, who criticized Flash in a public letter at the beginning of 2010 [1]. Since then, both technologies have evolved and many companies are being forced to evaluate which technology to use for their products.

The general agreement in the web community is that both technologies have their own unique strengths and weaknesses. One is not wholly superior to the other. My job was to evaluate NetApp's needs and to see which technology is best for NetApp and its customers as we move forward to future products and releases. This is an exciting time to be doing researching about Adobe Flash and HTML5 – new developments occur every month.

Chapter 1 will give background information on HTML5 and Adobe Flash, and Chapter 2 will address the current state of NetApp and the product in question. In Chapter 3, I present a process for how to go about making a recommendation. I decide which approach to take in Chapter 4 and narrow down my research criteria in Chapter 5. In the several chapters after that, I discuss the research criteria and evaluate HTML5 and Adobe Flash based on those criteria. In Chapter 10, I will summarize my recommendation. Finally, I will conclude with my contributions in Chapter 11.

1.1. Adobe Flash

Adobe Flash allows video, audio, interaction, and animation content to be easily integrated into webpages. Users need only to download a free plugin available for most web browsers and some mobile phones: Adobe Flash Player. Developers, however, need to buy a license. Flash exists

for most desktop operating system and has 99% penetration across all Internet desktop users. The only operating system Flash does not run on is iOS [2].

Because of Flash's many capabilities, many websites requiring complicated visualizations operate in Flash. The majority of web-based games are written in Flash due to the ease of capturing user input from the mouse and keyboard. Flash uses vector and raster graphics to easily draw animations of text and drawings. Many libraries and IDEs exist for Flash developers to easily create complex webpages.

1.2. HTML5

HTML5 refers to the combination of HTML5, CSS, and JavaScript, which is the convention I follow in this thesis. HTML5 is a markup language for structuring content, CSS is for creating the layout, and JavaScript is used for functionality, animations, and drawing. HTML is not proprietary; therefore anyone can use it as a development tool, and users can view content without any third party plug-ins. Previously, however, HTML had no video and audio support and limited drawing capabilities. Adobe Flash was the only option for developers who wanted to incorporate video, animation, and extensive user interactions. HTML5 was created to include these features into the web browser. HTML5 contains a new Video element, Canvas element, Audio element, offline storage, and others [3]. These new elements allow video and audio to run native to the web browser. HTML5 is still in development, and currently, all browsers do not fully support all its features. For example, developers need to keep that in mind that different browsers support different combination of formats for video and audio.

Another new feature in HTML5 is the Canvas element that allows animation to be easily written in JavaScript alone. Local database caching is also available now, allowing websites to

load faster and to operate offline. Many new elements are available in HTML5, but many of them are not fully developed, and common browsers are at various stages of compatibility.

1.3. Move towards HTML5

Many large companies are turning to HTML5 for their website needs. Microsoft Skydrive, a website that provides cloud storage, stopped using Microsoft Silverlight in favor of HTML5 [4]. As of November 2011, Adobe Flash ceased efforts in their mobile plug-in [5]. Instead, they have started to convert their own Flash mobile apps to HTML5. Google's doodle in July 2011, a swinging mobile that uses real-time 3D rendering to change depending on user interactions, was the first doodle to be created solely using HTML5 canvas [6]. Steve Jobs publically denounced Flash, stating that Apple developers should not have to rely on third party software and that Apple had no plans of making iOS compatible with Flash. Despite all of this, Flash is still the most reliable way to create interactive websites across all browsers.

In the next chapter, I will provide background into OnCommand Unified Manager, the NetApp product that I will be focusing my analysis on.

2. NetApp

2.1. NetApp now

This research focuses on NetApp's OnCommand, a suite of data management software that deploys, automates, protects, and monitors storage for customers [7]. It provides a unified management system that customers can use without the need for multiple applications that each performs only one kind of task.

Many parts of the OnCommand Management Software utilize Adobe Flash. Some applications contain a few Flash components while some applications like NetApp Workflow Automation are full blown Flex applications. The hope is to align all the applications onto one platform. My group, User Experience, has been primarily been concentrating their efforts in enhancing the user experience for OnCommand Unified Manager, the product my thesis will focus on.

2.2. OnCommand Unified Manager

Unified Manager allows storage administrators to view their storage environment with a “single pane of glass”. The continuous monitoring helps storage administrators visualize how to efficiently increase storage utilization and when and which storage environments need attention. Different policies can be specified for each dataset, allowing automation in provisioning and protecting similar types of data [8].

In particular, the dashboard interface is very useful for customers; it allows users to quickly evaluate the state of their storage by glancing at data visualizations. Data visualizations show which storage units are available and the current trend of data growth. From looking at the front-page dashboard, the user can choose which aspect of the storage he or she would like to focus on and select the appropriate tab or dashboard item. The dashboard also alerts the user to events that need attention. Currently, all of the data visualizations, including those not found on the dashboard, use Adobe Flash.

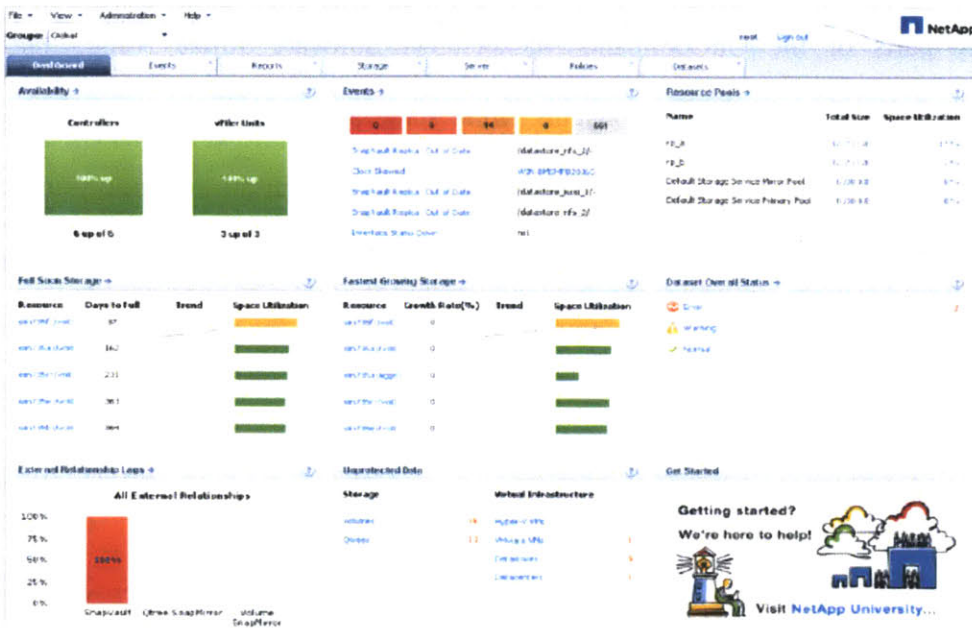


Figure 1. Dashboard of OnCommand Unified Manager

2.3. NetApp Web Framework

The OnCommand Unified Manager interface utilizes the NetApp Web Framework for all its layouts and widgets. The NetApp Web Framework (NWF) is a Rich Internet Application framework that was created to ensure that all NetApp web applications have a consistent look and feel. Developers at NetApp can use this library of widgets to easily create user interfaces.

The framework is built upon Google Web Toolkit (GWT) and Ext GWT (GXT), open source tools that allow developers to create JavaScript front-end applications using Java. GWT allows for easy debugging and compiles Java code into JavaScript. Ext GWT is an extension that extends the GWT library and allows front-end focused developers to create high-performance widgets.

NWF provides a unique set of tools that focus on data management with a set of widgets unique to NetApp.

My task was to review NetApp OnCommand's requirements and recommend whether they should focus on HTML5 or Adobe Flash moving forward. In the next chapter, I will address the methods I used to learn about constructing a recommendation.

3. Constructing a recommendation

In order to make a recommendation to NetApp, I had to understand the process of making a technical recommendation to a company. I investigated how top consulting firms approach their recommendations and how an MIT research group reached their decision on the same issue (HTML 5 vs. Flash). I spoke with a consultant from Booz Allen Hamilton, a student intern working at McKinsey, and a researcher at the MIT Media Lab. These conversations helped me identify a general approach to crafting this recommendation.

3.1. Learning from a consultant

Hal Abelson, my advisor, introduced me to Reggie VaLee, a Senior Partner at Booz Allen Hamilton and a great friend of MIT. He led the Booz Allen Hamilton initiative with MIT that led to the creation of OpenCourseWare. Through conversations with Reggie, I was able to understand how a top consulting firm approaches a problem similar to mine. As a consulting firm, Booz Allen Hamilton constructs many technical recommendations. As a result, Reggie is very familiar with the process of approaching technical decisions and producing a recommendation.

First and foremost, Reggie emphasized the selection of criteria before researching in detail. When making a recommendation, he first comes up with tangible criteria, such as speed and costs, and weighs them by importance. He evaluates the trade-offs between the criteria in order to narrow the focus and determine the order of importance. Asking users about their preferences helps with brainstorming criteria. Examining the users is very important, but users can also miss important criteria, so as the recommender, I would have to think harder to come up with criteria they miss. He also emphasized the value of using quantitative criteria over qualitative criteria to provide for a more straightforward and convincing end evaluation. Many of Reggie's recommendations involved competitive analysis by weighing cost analysis and economic trade-offs. Reggie suggested that the final recommendation should be presented in two ways: a full explanation and an executive summary. This accommodates both people who wish to understand every aspect of the reasoning behind the recommendation and people who would like the recommendation with simple explanations.

Finally, he provided me with general advice about "analysis paralysis", the tendency to keep researching as opposed to coming to a conclusion. The most important thing is to build confidence around an answer and to not being bogged down by all the research and data; after all, the research has to end at some point. That being said, it is still very important to spend time upfront building criteria and weighing those criteria.

3.2. Learning from a student consultant

Hal Abelson also recommended that I talk to Ryan Ko, his MEng thesis advisee, who worked at McKinsey & Company in Palo Alto, California for the summer. Ryan has experience in

consulting for software engineering companies, and I consulted with him about making recommendations and decisions for software companies.

Ryan compared a recommendation to an academic paper where the writer discusses background, previous research, data found, and then presents the recommendation and reasoning behind it. Ryan emphasized backing up recommendations with as much evidence as possible and providing reasons for and against it. In his work, he often gave his client five different options and explained to them which one he thought was best. He would also explain the pros and cons of the other options too. He also said it was important to mention weaknesses in the arguments and any assumptions made.

The final deliverable would be a detailed PowerPoint presentation of fifty to one hundred slides, which was presented in a three-hour meeting. The presentation would contain all research, including the sources, and read almost like a long research paper. There would also be a shorter version for distribution that was about ten slides long.

3.3. Investigating a similar decision - Scratch

The Lifelong Kindergarten group in the MIT Media Lab also made a decision that entailed choosing between HTML5 and Adobe Flash for one of their projects, Scratch, which was launched in 2007. Scratch is a graphical programming environment that is used by kids to create interactive stories, games, and simulations [9]. The programming is done visually by snapping together graphical programming blocks to create a flow. Treating programming as a form of story telling teaches children computer programming concepts.

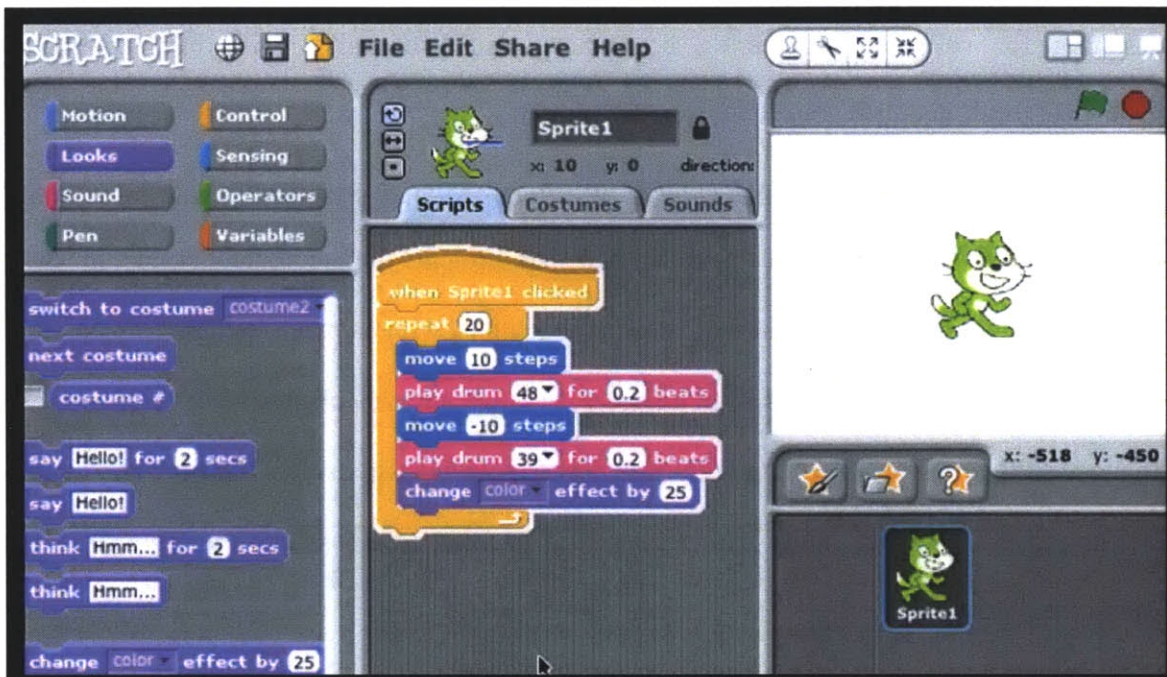


Figure 2. Example screen of a Scratch Project [10]

John Maloney is a staff researcher at the MIT Media Lab and is focusing on Scratch. He was involved in the decision to use Flash, instead of HTML5, for the Scratch project. I interviewed him about which aspects of the technologies he considered and why he ultimately chose Adobe Flash. Although Scratch and NetApp have different needs, it was instructive to see if any criteria from the Scratch project applied to NetApp. Seeing the process of how another project made their decision was also very helpful.

HTML5 was considered for Scratch 2.0, but ultimately, it did not have the features that Scratch required. Some these features included real-time audio generation, fast pixel manipulation, cross-platform support for microphone and video input. The HTML5 specifications are also not standardized across browsers; different features behave differently in different browsers.

John also mentioned that Scratch needs to support a user community with hundreds of thousand of users and over two million projects. This means that the current features in Scratch have to be supported in Scratch 2.0. Scratch places a particular emphasis on audio, which seemed to be better supported by Flash. He recommended, that if I started a new project, I would be in a better position to choose features that can be implemented in HTML5. By using HTML5, I would ensure that iPhones and iPads can support the application.

In the future, John is considering to port Scratch over to HTML5, after HTML5 has fully developed and ensured its features function across all browsers. There is currently a UROP student working on building the Scratch player in HTML5, and after considering the strengths and weaknesses of HTML5 that are revealed through that project, the group will reconsider their choice for the next release.

3.4. Summary

After discussing with two technical consultants and learning how MIT Media Lab's Scratch made their decision, I developed strategies to approach how to research for and construct a recommendation. An overarching theme was that the first step to any kind of recommendation is to come up with a set of critical criteria for the decision. After this set of criteria is narrowed down, the criteria have to be weighted for importance. The talks gave me insight into what kind of criteria to look for – the best approach is to examine the user population and their needs. Flash was chosen for Scratch because it focused on audio and pixel manipulation. The audio and appealing graphics were critical to motivating children to create their own story projects.

My final deliverable is this thesis. By reading my thesis, the User Experience and User Interface group at NetApp will be able to view my recommendation, my reasoning and evidence

behind my recommendation, and also advantages of using the other technology. From there, they can decide how to proceed. On my return to NetApp, I will set up a meeting to present my findings with a short slide deck.

My approach, which is described in the next two sections, will explain the process I went through to narrow down and weigh criteria.

4. Approach

There are many possible methods to approach the question of whether NetApp should use Flash or HTML5 in its management suite applications. A comprehensive analysis covering every facet would require much more time and is beyond the scope of this thesis. Instead, I chose to focus on examining one issue thoroughly. My manager, John Bagby, helped me brainstorm many ways to approach the problem, and after choosing an approach, I further narrowed down criteria in the next section. Below are a few of the approaches I considered.

4.1. Development efficiencies

One possible way to approach the problem would be to examine the advantages of each technology from the developer's point of view: the software engineers at NetApp. Because NetApp's user interaction currently utilizes Adobe's Flash, it would cost developers time and money to convert to another technology. This would involve evaluating whether the benefits of transitioning to HTML5 would outweigh the cost to convert to HTML5.

Evaluating the ease of developing with each technology would also be very important. This would require the investigation of development environments and assessment of the ease of

development in that language. I would also look at the libraries available to HTML5 and Flash, especially those dealing with data visualization components. Pre-existing libraries, such as FusionCharts and Highcharts, that can help developers produce data visualizations easily and efficiently would be potentially more cost-efficient for the company. The visualizations would also be more consistent, allowing NetApp to more easily produce a cohesive design.

Finally, I would also investigate the cost to license these technologies for business use. Other criteria, such as ease of use for development, would take precedent. Cost would only be taken into account if there were a large differential.

4.2. Business drivers

Another method to craft that can be used this recommendation is to examine the business drivers of each technology and NetApp's interests.

One aspect of investigation would be the openness and closedness of Adobe Flash and HTML5. HTML5 is completely open-source while some parts of Flash are closed. As a proprietary company, NetApp may want to take this into consideration when deciding between the two technologies. As Steve Jobs says, relying on a third party technology is disadvantageous because there exists an extra layer between the developer and the platform. The developer has to rely on the third party to create new enhancements.

Additionally, if NetApp were to change to a new technology, many customers would still use the old version of the software, and NetApp would have to support both versions, at least for a short time. The cost of this dual maintenance would be another aspect to consider.

Another aspect to evaluate would be NetApp's inclination towards mobile devices. Because Flash does not work on popular devices such as the iPhone and iPad, any customer using

a web application would not be able to view Flash components. Because HTML5 is native to the web browser, HTML5 visualizations would be able to show up.

Investigation of current trends in the industry would allow NetApp to be able to conform to industry standards.

4.3. End-user satisfaction

The final approach considered was from the end user's point of view. This would involve evaluating the needs of NetApp's customers who currently use the products to see which technology is best suited for their needs.

4.4. Summary

After much deliberation, I decided that evaluating these technologies depending on the end user's point of view would be the most valuable for NetApp. NetApp's strategy for gaining market share is based on building innovative products that delight customers. In this light, user needs is of paramount importance. My conversation with Reggie also reminded me that ultimately, the user's opinions matter the most. That said, NetApp should also look at development convenience and business drivers when making a decision.

In the next chapter, I discuss the criteria I use to evaluate the technologies from the user's point of view.

5. Criteria for Evaluation

Prior to starting my research, my manager and I discussed a few concrete criteria to assess the advantages of Flash and HTML5 from an end-user's point of view. For each of these criteria, I

examined each technology's capabilities and decided which technology had the advantage for each criterion. I then decided whether one technology had a dominating advantage over the other. The criteria are listed in order of importance.

5.1. Data visualization capabilities

Data visualization is the most important function that Flash provides to NetApp's products such as OnCommand Unified Manager. The graphs and charts allow end users to easily digest a large amount of data in a single glance. Currently, NWF uses a Flash library called FusionCharts to create most of the visualizations. As part of my investigation, I explored HTML5 options to see how they compared.

Some of the capabilities that I compared were hovering, master detail, pivot model, drill model, contextual actions, and indicators. Hovering, contextual actions, and indicators allow the user to interact with charts and to see more information if they wish. Master detail and the drill model allow information to be condensed and then expanded if the customer desires to see more details or has to look into a hierarchy of storage. The pivot model allows the customers to switch between different views and see a variety of information quickly, without consuming space on the interface.



Figure 3. An example of a treemap, a visual representation of how storage is divided. A user can hover over a block and see the status of the storage. They can also click and drill into it to view more details.

A side-by-side comparison of HTML5 and Adobe Flash capabilities will give a visual comparison of the two technologies. I extracted a few of the important graphs or figures from OnCommand and reproduced them using Highcharts, an HTML5 and JavaScript based charting tool. In the process, I realized how difficult or easy it would be transition to HTML5 from the developer's point of view.

5.2. End-user customization

NetApp has customers who operate in a variety of industries. Certain users might like to pay more attention or conduct a more in-depth analysis of relevant data that might not be important to other customers. It would be inefficient and expensive for developers to provide every customer with every customization that they require. The interface would also be very cluttered if every option were to be displayed.

Even within an industry, there are different kinds of users that use OnCommand. Some are day-to-day hands on operators, and others have a long-term planning viewpoint. Different kinds of users may want to focus on different aspects of the interface.

I conducted research on how easy it is to use HTML5 and Adobe Flash to create interfaces that are customizable by the user.

5.3. Platform

The first aspect of platform that I evaluated is which browsers support HTML5 and Adobe Flash and which products are compatible with which browsers. Internally, NetApp IT supports Internet Explorer 6.0 and 7.0. I investigate what the bulk of NetApp's products are compatible with and see whether that browser supports HTML5 and Flash and which technology performs better with each browser.

I also evaluated NetApp's inclination for mobile applications and decide which technology is best for that.

5.4. Special Needs

NetApp's products can have special needs that are unique to the data storage industry. One such special need is the ability to deal with very large amounts of data. In creating data visualizations from large amounts of data, some time might be needed to render the visualization.

I evaluated HTML5 and Adobe Flash based on the amount of time it takes for them to create a data visualization of data. Because Unified Manager already processes large amounts of data, I knew that Adobe Flash does a sufficient job of large dataset visualization. I investigated whether HTML5 can also effectively produce effective visualizations from large datasets. Efficiency and promptness is very important to the customers using an interface such as

OnCommand Unified Manager, where the goal is to be able to process information quickly. Waiting for the dashboard to load with its charts would not be conducive to managing a storage environment.

5.5. Interoperability

NetApp's management capabilities sometimes need to be integrated into the management software from other vendors, such as VMware. I investigated which platforms the other vendors are using, and the capability for plugging-in NetApp software into those platforms. If NetApp were to transition to HTML5, I needed to make sure that the products would integrate into other company products as seamlessly as before.

5.6. Summary

The main focus of this thesis is to compare data visualization options for HTML5 and Adobe Flash. OnCommand Unified Manager currently uses Flash for these purposes. When comparing data visualizations, I evaluated the ease of use for users to customize their interfaces. This will be the most important criterion. I also looked at platform and touch on the implications of going mobile. I then compare data visualization loading speed for HTML5 and Adobe Flash to assess the potential for HTML5 to represent large amounts of data. Finally, I examine how plug-ins for other companies affects the decision.

The next chapter is the beginning of my evaluation of the criteria. I begin with the most important criterion: the comparison of data visualization and user customization in HTML5 and Adobe Flash.

6. Data Visualization and Customization

Data visualization is the most important criterion because OnCommand is currently using Flash for its graphs and charts. Efficient and thorough data visualization enables users to quickly determine what data storage devices need his or her attention. First, I will explain which data visualization capabilities NetApp is looking for, and then I will compare NetApp's current data visualization chart, FusionCharts, that requires Adobe Flash, to other existing HTML5 charting applications. I will also try to reproduce the chart currently in use, using those HTML5 charting applications.

First, I will examine the different types of graphs and charts offered by the applications and how flexible they are, in case NetApp wants to create a chart that does not exist in the library. Next, I will consider capabilities that allow users to interact with the chart such as master detail, drilling into detail, hovering, contextual actions, and the pivot model. Master detail allows user to hover over a data point and allows the user to see more information and exact quantities about that data point that would otherwise clutter the graph. The drill model would allow information to be condensed and then expanded by user clicks when the customer requires more details.

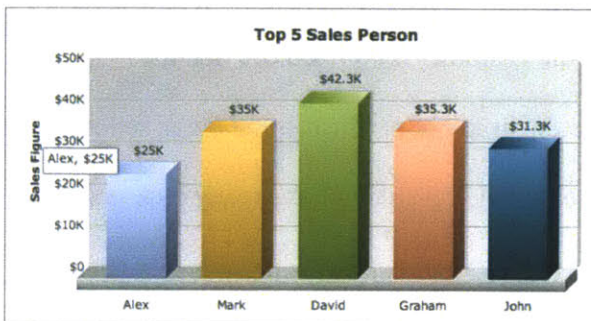
Contextual actions allow user to manipulate the graph or perform some other action without a need to move to another window. The user should be able to hover over data, resulting in a hover bubble with an associated action. The pivot model would allow customers to switch between different views and chart types and to quickly view variety of information, without taking up space on the interface.

6.1. Fusion Charts

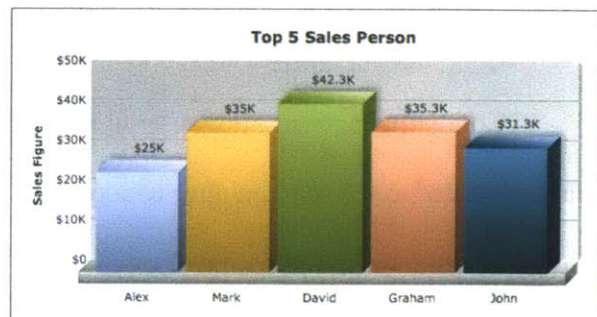
Currently, the OnCommand Unified Management interface uses the Adobe Flash version of FusionCharts. FusionCharts allows users to render charts in Flash or HTML5/JavaScript. At the time OnCommand was released and my thesis proposal, these alternatives had not been released yet. The FusionChart website provides detailed documentation for both versions and states that the use of HTML5 would only require copying and the JavaScript FusionChart files and not a single line of code would need to be changed. FusionCharts actually bundles Highcharts, a JavaScript charting solution, with the Flash charts to allow charts to be rendered in JavaScript[11].

The charts used by OnCommand contain appealing animation to users. For example, when the user navigates to a page displaying a column chart, the columns rise up to the appropriate height.

Below are comparisons of the same graph in JavaScript and Flash. The 3D graphs show some aesthetic differences while the 2D graphs are replicas of each other.



JavaScript



Flash

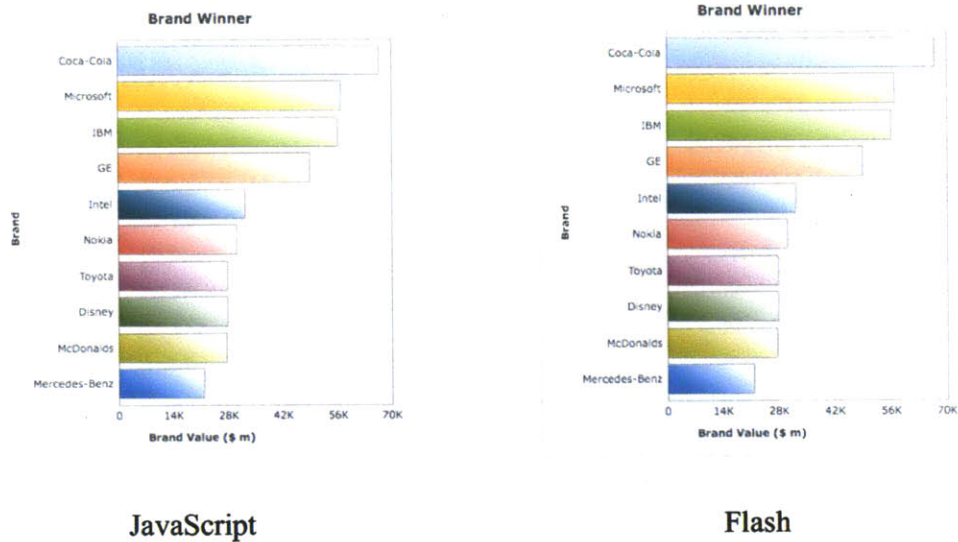


Figure 4. Comparison of Flash and HTML5 FusionCharts [12]

6.2. Highcharts

One of the most popular HTML5/JavaScript charting libraries is Highcharts. Highcharts supports many kinds of charts including, line graphs, scatter plots, bar charts, pie charts, column graphs, area graphs, and ways to combine these graphs into one [13].

Highcharts features all of the capabilities previously listed. Users are able to hover over data points and view more detailed information. In the examples shown, users can see the exact exchange rate of USD to EUR on a certain date (*Figure 5*) and see the amount of rainfall in Tokyo in a certain month (*Figure 6*). The drill down model allows users to click on data (*Figure 7*). In the example shown, browser market share is displayed on a column chart. By clicking on a column that represents the total percent of market share a browser owns, the user can drill down further and view column charts that break the browser share into specific versions of that browser. The pivot model allows users to seamlessly choose the range of data they would like to look at without opening a new window or navigating to another page.



Figure 5. Example of master detail, pivot model, and multiple graphs.

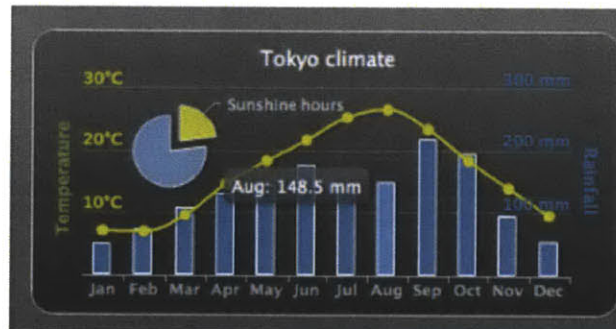


Figure 6. Example of master detail and multiple chart types.

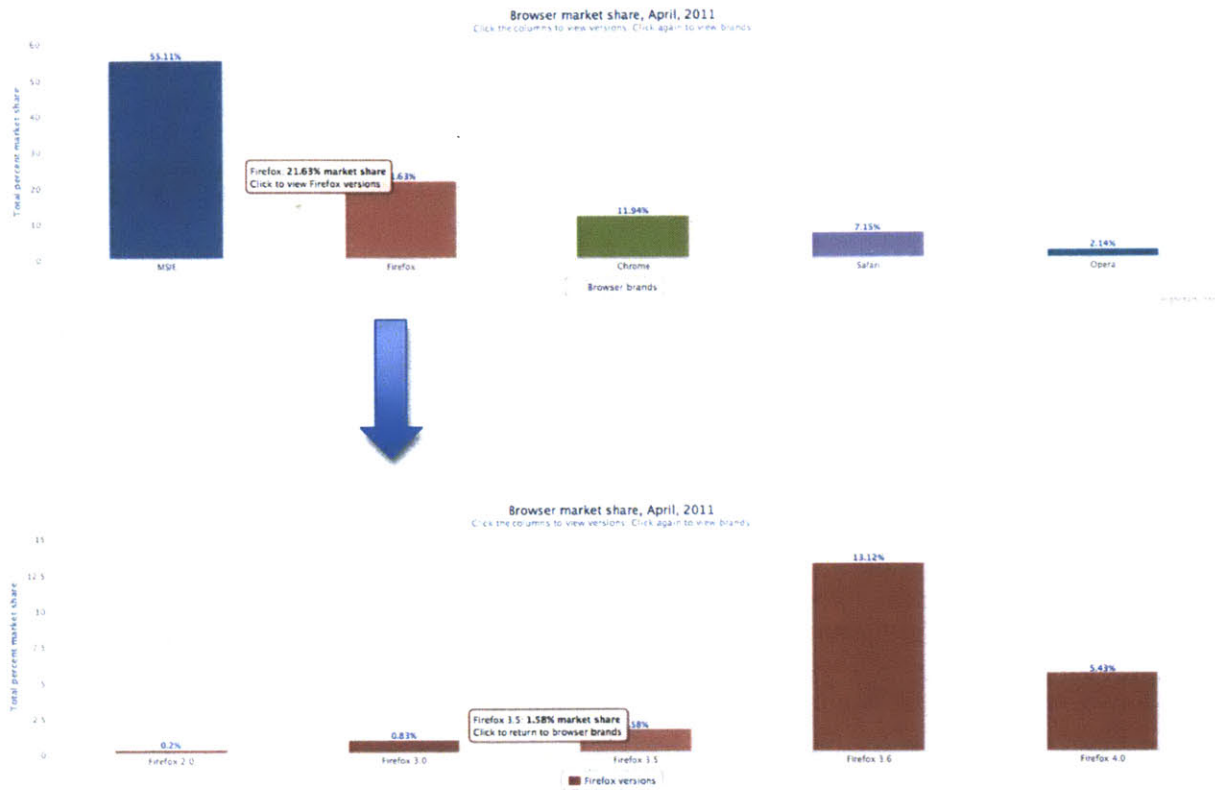


Figure 7. Example of drilling down.

Highcharts would be easily integrated with OnCommand via an existing open source API that allows Highcharts use within a GWT application [14]. Highcharts has been tested to be fully functional with Internet Explorer 6.0 +, Firefox 2.0 +, Chrome 1.0 +, and iOS 3.0 + [15].

Additionally, I looked at the graph detailing external relationships on the OnCommand Unified Manager Dashboard (*Figure 1*). I used Highcharts to replicate the FusionChart. Except for the data, which I changed to make the chart look more interesting, everything else is the same. I was able to manipulate the gridlines, axis labels, column labels, gradient, and various colors (*Appendix*).

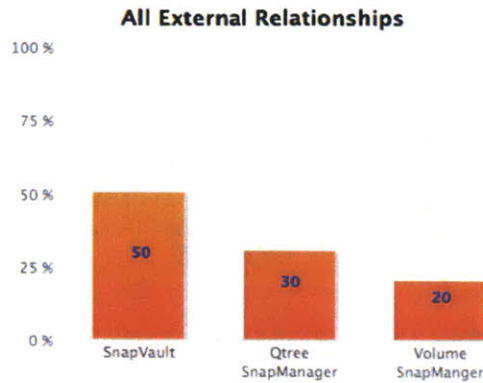


Figure 8. Similar chart to bottom-left chart in Figure 1 using Highcharts.

6.3. Summary

In terms of data visualization, HTML5 is on par with Flash. Although FusionCharts was previously only available in Flash, it can also be rendered in JavaScript with negligible differences in look and feel. Highcharts, which solely uses HTML5/JS is compatible with the most common browsers, and contains user customizations, like pivot model and master detail, which would be necessary to include in OnCommand Unified Management.

The next chapter will review browser compatibility, animation speeds, and mobile development.

7. Platform Support

7.1. Browser

To decide which technology to use, I investigated browser compatibility for each technology, which browsers NetApp OnCommand products support, and which browsers users

typically use. Because HTML5 is best for mobile development, I also evaluated the likelihood that NetApp would choose to go mobile in its product.

Currently, Unified Management is supported on Internet Explorer 8 and Mozilla Firefox, versions 3.0, 3.5, and 3.6. NetApp’s internal browser of choice is Internet Explorer [16].

As of March 2012, Flash reached 99% penetration through all browsers while HTML5 only reached 50%; this statistic is actually 10% higher than in May 2011, which shows the incredible growth of HTML5 [2].

The picture below displays the progression of HTML5 over the past four years (*Figure 9*). The spokes of the wheel represent different features of HTML5 and the different colored boxes represent different browsers. The spokes include Canvas, SVG, Audio, Animations, Forms, and other elements. It is obvious from the consistent increase of colored boxes that HTML5 support is growing more and more every year.

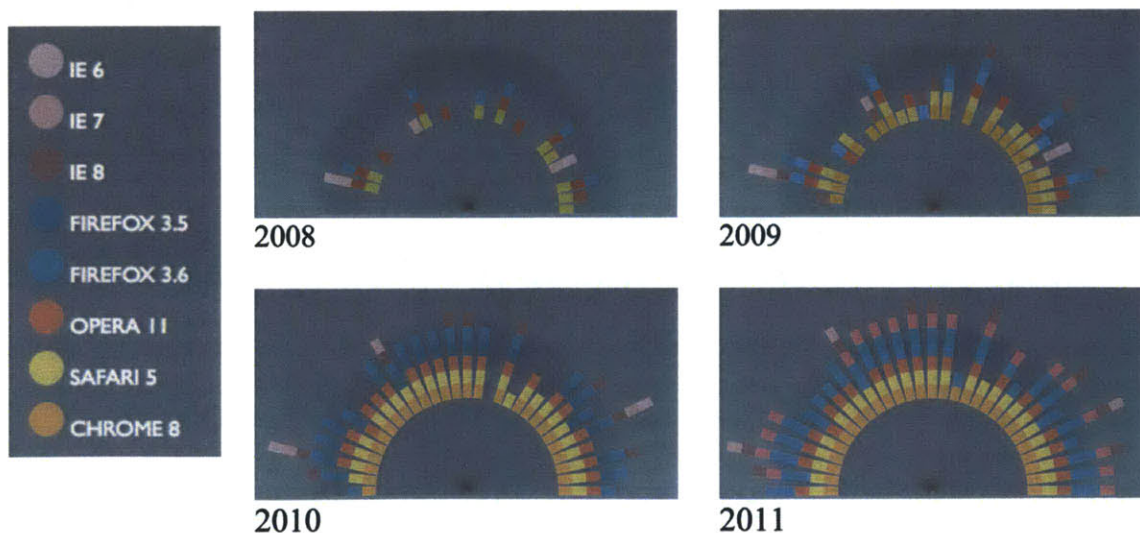


Figure 9. Progression of HTML5 from 2008 to 2011. [17]

Below is visual representation of which Flash and HTML5 interactive features are functional in which browsers (*Figure 10*). The first column is Flash, the second is HTML5, and third is HTML5 in mobile browsers. Flash's interactive features are compatible with all browsers while HTML5 has many features that have either no or partial support.

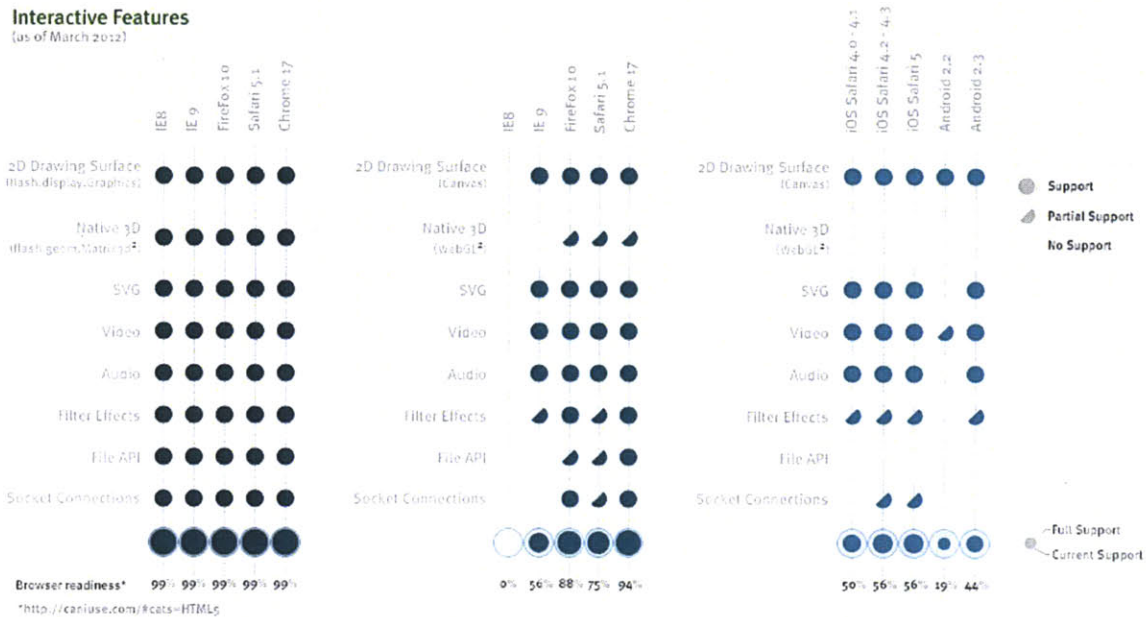


Figure 10. Interactive Feature support for Flash, HTML5, and HTML5 on mobile devices. [2]

7.2. Benchmarking

People have performed tests such as benchmarking CPU usage in video and measuring frames per second for animations. Because OnCommand does not use Flash for video, but does use it for chart animation, I will only post animation results. This tester used a simple animation to monitor the frames per second for HTML, Canvas, SVG, and Flash in Internet Explorer 9, Firefox 9, and Chrome 16 [18].

	HTML (FPS)	HTML5 Canvas (FPS)	HTML5 SVG (FPS)	Flash (FPS)
Internet Explorer 9	-	80	60	51
Firefox 9	27	69	3	51
Chrome 16	40	33	57	51

Table 1. Animation frame rates from January 2012

In summary, HTML5 performs better than Flash in Internet Explorer while HTML5 Canvas performs better than Flash in Firefox. Flash performs better in Chrome than Canvas but worse than with SVG. The tester also performed the same test a year earlier in March 2011, but only with Firefox and Chrome.

	HTML (FPS)	HTML5 Canvas (FPS)	HTML5 SVG (FPS)	Flash (FPS)
Firefox 4	19	27	3	56
Chrome 10	28	47	62	56

Table 2. Animation frame rates from March 2011

There is a definite improvement in the Canvas element of HTML5 for both browsers. This suggests that not only has HTML5 improved, but that later browser versions are making more of an effort to support HTML5.

7.3. Mobile

It is obvious that the world is moving towards mobile applications: more and more people rely on their phones to monitor different aspects of their daily life. As of May 2012, US smartphone penetration crossed the 50% line to 50.4% [19]. The question is whether NetApp is going mobile and whether OnCommand has a likelihood of going mobile. If NetApp has a

potential to go mobile, then OnCommand must drive for applications that will operate on mobile devices. Adobe Flash does not run on iOS, the operating system that runs on iPhone, iPad, and iTouch. Because Apple has no future plans of allowing Flash to run on its operating system, Flash is abandoning the Flash plug-in for mobile devices in favor of supporting HTML5, so HTML5 must be used by browser applications that hope to run on iOS. As the most popular smartphone, the iPhone must be taken into consideration when developing mobile applications. I believe that NetApp, like many other companies, is moving towards mobile and because of this, they should consider moving their applications to run solely on HTML5.

An example of NetApp's likelihood towards mobile development is exemplified by a new mobile application that was available on March 27, 2012 called the OnCommand Resource Navigator. This application allows customers easy, quick access to information about OnCommand such as datasheets, webpages, and community forums. There is also a storage problem solver where users can answer a series of questions to narrow down a problem and find a solution. The hope is for users to have the ability to open these applications on their phone while they are using OnCommand on their computer. Currently, there are only self-help resources available, but it is possible that in the near future, monitoring data can be done through the phone browser or through an application on the phone as well.

Additionally, during the summer, an intern's project was to create an iPhone application that could monitor and retrieve health information about filers remotely from the user's phone. Being able to monitor the health of a data center from the user's phone is not that far off from having a storage administrator monitor data from his or her phone.

7.4. Summary

Flash currently has the most penetration across all browsers with HTML5's features only functional in half. However, the use of HTML5 is growing. Last year, the growth was 10% and predicted to increase as demand for HTML5 increases. Native technology is also better because a supported website in HTML5 will automatically work on iOS. Because of NetApp's inclination to go mobile, it is necessary to choose a technology that fully supports that trend.

In the next chapter, I investigate the speed of webpage loading in Adobe Flash and HTML5 and the feasibility of visualizing large amounts of data in HTML5.

8. NetApp's Special Needs

8.1. Speed

Speed is crucial to the end-user experience because users do not want to wait a long time for charts and graphs to load. If one technology takes significantly longer than another to load data visualizations then speed is another factor that must be taken into account.

I used a website called WebWait [21] to benchmark how long it takes for a demo management dashboard to load in Flash and JavaScript. The graphs used are similar to those used in OnCommand Unified Management (*Figure 11*). For each browser that OnCommand Unified Management is compatible with (Internet Explorer 8 and Firefox 3.6), I ran the call fifty times with a five second break in between on a ThinkPad T400 computer for two similar websites. On the surface, the websites both look identical, but one is written in Flash and the other in HTML5 and JavaScript.



Figure 11. Website used to measure speed of loading. [20]

		Average (secs)	Median	Standard Dev
Internet Explorer	Flash	0.24	0.2	0.15
8	HTML5/JS	1.09	1.27	0.32
Firefox 3.6	Flash	1.12	1.07	0.36
	HTML5/JS	0.99	0.96	0.04

Table 3. Comparison of website loading speeds

Surprisingly, speed depended on which browser was used. Flash was faster in Internet Explorer 8 by .85 seconds, and HTML5/JavaScript was slightly faster in Firefox by .13 seconds. Internet Explorer was fastest overall. Because speed was dependent on the browser, it is hard to decide which technology is better. Unified Manager’s pages are not any more complex than the sample data visualization page above, so data visualization loading speeds would not differ significantly from those listed above. Web page loading speeds are also highly dependent on the computer and network connection; therefore, I do not feel the above differences significantly affect

the user experience. If the performance on the page is already poor, these speeds would not make the performance much worse.

8.2. Large Datasets

NetApp OnCommand Unified Manager processes large amounts of data, which is transformed into data visualizations to easily allow customers to assess their storage in one glance. I needed to ensure that HTML5 would not only be able to handle large of amounts of data, but also be able to rapidly update visualizations based on changing data.

Yahoo has used HTML5 to create data visualizations that use large amounts of data and update in real-time [22]. The Yahoo mail visualization tool illustrates mail activity around the world in real-time with circles constantly popping up to show the current volume of activity occurring in a specific region of the world. Users can also choose to see the large amount of spam Yahoo mail is filtering out. Graphs are displayed at the bottom to show the number of emails being sent over time. The visualization is very dynamic and users can interact with it and drill down into details by hovering over difference regions. This application works on the iPad and iPhone. Yahoo's visualization tool proves that HTML5 can provide constantly updated data visualizations that are interactive, visually pleasing, and use large amounts of data.

THE Yahoo! MAIL NETWORK IS DELIVERING 60,920 EMAILS PER SECOND WORLDWIDE.

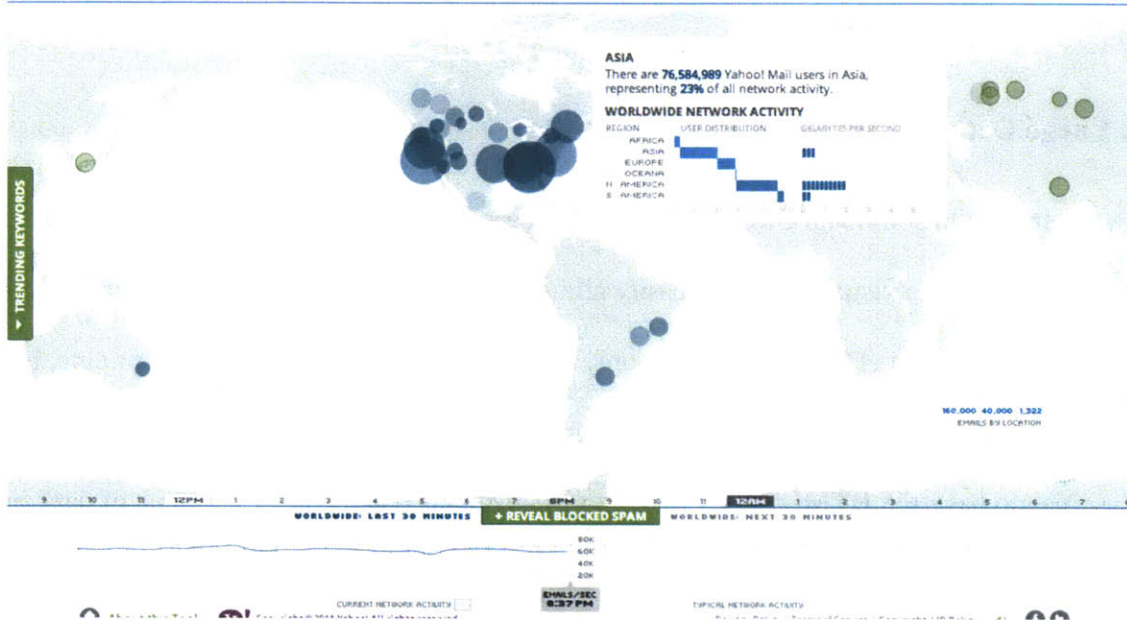


Figure 12. Yahoo mail data visualization [22]

8.3. Summary

This chapter has used a case study to demonstrate that HTML5 is able to handle large amounts of data to create interactive data visualizations. It also demonstrates how webpage loading speeds depend on the browser in use. They are slower for Flash in Firefox but faster in Internet Explorer.

In the next chapter, I will touch briefly on how plug-ins for other companies would be affected by the technology choice.

9. Interoperability

9.1. Microsoft and VMware Plugins

There exists OnCommand plug-ins for Microsoft and VMware storage administrators to further manage their storage environments [7].

Microsoft storage administrators can use the Microsoft OnCommand Plug-in (OCPM) to link with Microsoft System Center 2012. They can automate the monitoring of storage health, performance, and workflows. The disaster recovery procedure is also much simplified and along with other NetApp tools, administrators can mitigate damage and recover much faster. The Microsoft System Center is not based off Flash, so using HTML5 for OnCommand should be feasible.

The VMware plugin, however, encourages a Flash deployment because VMware vCenter web client, a tool to manage and monitor virtual servers and desktop, runs on Adobe Flash and requires installation of the Flash plug-in.

9.2. Summary

It is important that NetApp is able to integrate management software from other vendors with OnCommand. The VMware plugin is currently Flash based, but there is potential for that to change. We need to keep in mind that other companies will change their mindset about Flash and HTML5 as well. Hopefully, VMware will also be moving towards HTML5 and move their vCenter application off of Flash. Alternatively, Adobe may find a method to convert Flex applications into HTML5, or NetApp or VMware can use a bridge to link two applications.

In the next chapter, I conclude my findings by summarizing my recommendation on why NetApp OnCommand should focus on HTML5 moving forward.

10. Recommendation

Criterion	Conclusions
Data Visualization and Customization	HTML5 capabilities in data visualization and customization match that of Adobe Flash.
Platform Support	HTML5 browser support is improving and catching up to Flash. For mobile development, HTML5 is the best choice.
NetApp Special Needs	HTML5 is capable of producing data visualizations from large amounts of data and the speed it takes to load webpages is comparable to Flash.
Interoperability	Flash deployment is necessary for the VMware plug in, but market trends show that many large companies are moving towards HTML5.

Table 4. Summary of Recommendation Criteria

I believe that NetApp should follow the trend of the current market and move towards having all web applications run natively in the browser using HTML5. Data visualizations created from HTML5 are sufficient for OnCommand's purposes and provide a sufficient level of functionality and detail. Large companies, such as Yahoo, have already created dynamic, interactive data visualizations using large amounts of changing data. In order to compete in the growing mobile trend, NetApp should consider going mobile with HTML5. iOS will never be

compatible with Flash; therefore, the best solution would allow OnCommand to run on a mobile device via a web application without a need for additional plug-ins. I recognize that HTML5 is not yet complete in functionality yet and many common browsers do not fully support all its functions. However, based on all the evidence that suggests HTML5 is growing and improving (*Table 4*), I think HTML5 will be able to mimic Flash's capabilities in a few years.

The industry trends show that many well-known companies are moving towards HTML5. In addition to Google and Microsoft (Chapter 1.3), there are several more examples of this trend. YouTube launched their HTML5 Video Player Beta in January 2010 with very limited functionality. Since then, closed captioning, annotations, and native full screen support have been added [23]. YouTube is still actively working on supporting videos with ads and creating support for the h.264 codec in Firefox and Opera[24]. The progress YouTube has made shows efforts in improving HTML5 functionality. Additionally, Adobe released a tool called Adobe Edge in August 2011 to create animated and interactive content in HTML5, CSS3, and JavaScript [25]. The rich user interface allows developers to create content that runs on most browsers and mobile devices. Since then, many updates have been made; an updated version, Edge preview 6, was just released on May 11, 2012, demonstrating Adobe's desire to move forward with development in HTML5.

I will conclude with Adobe's comments on HTML5 after the announcement of discontinuing the mobile Flash plug-in. Adobe Flash stated in their Official Flex Team Blog that, "In the long-term, we believe HTML5 will be the best technology for enterprise application development... HTML5 related technologies (comprising HTML, JavaScript and CSS) are becoming increasingly capable, such that we have every reason to believe that advances in expressiveness (e.g. Canvas), performance... and application-related capabilities... will continue

at a rapid pace. In time (and depending upon your application, it could be 3-5 years from now), we believe HTML5 could support the majority of use cases where Flex is used today” [26]. HTML5 is the future of web application development.

11. Contributions

In this thesis, I have framed how to craft a recommendation and identified and weighed four criteria (data visualization and customization, platform, NetApp special needs, interoperability) to assess whether NetApp OnCommand should move towards using HTML5. I have demonstrated why NetApp should consider moving towards HTML5 and supported my conclusions with evidence of satisfactory data visualization and customizations, NetApp’s inclination towards mobile applications, and HTML5’s growth and progress. I have also shown how well known technology companies are moving towards HTML5 and have speculated that HTML5 will be the future web standard very soon.

12. References

- [1] Steve Jobs. *Thoughts on Flash*. [Online]. April, 2010. Available:
<http://www.apple.com/hotnews/thoughts-on-flash/>
- [2] Dino Citraro. *Our Research on Flash and HTML5: Ten Months Later, a Second Look*. March 21, 2012. [Online]. Available: <http://now.periscopic.com/2012/03/our-research-into-flash-and-html5-ten-months-later-a-second-look/>
- [3] Peter Wayner. *10 apps pushing HTML5 to the limit*. July 11, 2011. [Online]. Available:
<http://www.infoworld.com/d/html5/10-apps-pushing-html5-the-limit-699>
- [4] Mike James. *SkyDrive drops Silverlight*. June 21, 2011. [Online]. Available: <http://www.i-programmer.info/news/81-web-general/2640-skydrive-drops-silverlight.html>
- [5] Marc Chacksfield. *Adobe to abandon Flash plug-in for mobiles*. November 9, 2011. [Online]. Available: <http://www.techradar.com/news/phone-and-communications/mobile-phones/adobe-to-abandon-flash-plug-in-for-mobiles-1039691>
- [6] Sherry Stern. *Google Doodle: An interactive Alexander Calder mobile*. July 22, 2011. [Online]. Available: <http://latimesblogs.latimes.com/culturemonster/2011/07/google-doodle-an-interactive-alexander-calder-mobile.html>
- [7] *NetApp OnCommand Management Software*. [Online]. Available:
<http://www.netapp.com/us/products/management-software/>
- [8] *OnCommand Unified Manager*. [Online]. Available:
<http://www.netapp.com/us/products/management-software/oncommand/>
- [9] *Lifelong Kindergarten Projects*. [Online]. Available:
<http://llk.media.mit.edu/projects.php?id=783>

- [10] *Scratch*. [Online]. Available: <http://scratch.mit.edu/>
- [11] *FusionCharts Documentation*. [Online]. Available: <http://docs.fusioncharts.com/charts/>
- [12] *FusionCharts Column and Bar Charts*. [Online]. Available:
<http://www.fusioncharts.com/demos/gallery/#column-and-bar>
- [13] *Highcharts JS*. [Online]. Available: <http://www.highcharts.com/>
- [14] *GWT Highcharts*. [Online]. Available: <http://www.moxiegroup.com/moxieapps/gwt-highcharts/>
- [15] *Highcharts Documentation*. [Online]. Available:
<http://www.highcharts.com/documentation/compatibility>
- [16] *NetApp Library and Support*. [Online].
<https://library.netapp.com/ecmdocs/ECMP1119830/html/html/frameset.html>
- [17] *HTML5 and CSS Readiness*. [Online]. Available: <http://html5readiness.com/>
- [18] *Flash vs HTML5 Performance*. [Online]. Available: <http://pacoup.com/2011/02/03/flash-vs-html5-performance/>
- [19] Simon Sage. *U.S. smartphone penetration crosses 50%, but iPhone still trails Android*.
<http://www.imore.com/2012/05/07/smartphone-penetration-crosses-50-ios-tails-android/>
- [20] *FusionCharts Management Dashboard*. [Online]. Available:
<http://www.fusioncharts.com/demos/business/#management-dashboard>
- [21] *WebWait*. [Online]. Available: <http://www.webwait.com>
- [22] *Visualize Yahoo!*. [Online]. Available: <http://visualize.yahoo.com/>
- [23] Aayush Arya. *YouTube's HTML5 player steadily gaining ground on Flash*. [Online].
Available: <http://thenextweb.com/google/2011/11/21/youtubes-html5-player-steadily-gaining-ground-on-flash/>

[24] *YouTube HTML5 Video Player*. [Online]. Available: <http://www.youtube.com/html5>

[25] *Adobe Edge*. [Online]. Available: <http://labs.adobe.com/technologies/edge/>

[26] *The Official Flex Team Blog. Your questions about Flex*. [Online]. Available: <http://blogs.adobe.com/flex/2011/11/your-questions-about-flex.html>

13. Appendix

JavaScript code for Highchart

```
$(function () {  
  var chart;  
  $(document).ready(function() {  
    chart = new Highcharts.Chart({  
      chart: {  
        renderTo: 'container',  
        type: 'column'  
      },  
      credits: {  
        enabled: false  
      },  
      title: {  
        text: 'All External Relationships',  
        style: {  
          color: 'black',  
          fontWeight: 'bold'  
        }  
      },  
      xAxis: {  
        categories: [  
          'SnapVault',  
          'Qtree SnapManager',  
          'Volume SnapManger',  
        ]  
      },  
      yAxis: {  
        min: 0,  
        max: 100,  
        gridLineWidth: 0,  
        labels: {  
          formatter: function() {  
            return this.value + ' %';  
          }  
        }  
      }  
    });  
  });  
});
```

```

    },

    title: {
      text: null
    },
    stackLabels: {
style: {
  color: 'blue',
  fontWeight:'bold'
},
enabled: true,
verticalAlign: 'middle'
}
  },
  legend: {
    enabled: false
  },

  plotOptions: {
    column: {

      pointPadding: 0.0,
      borderWidth: 0,
    }
  },
  series: [{
    stacking: "normal",

    data: [50,30,20],
    color: {
linearGradient: [0, 0, 0, 300],
stops: [
  [0, '#CCCC00'],
  [1, '#FF0000']
]
}
  }
  ]
});
});
});

```