Dublin Institute of Technology
ARROW@DIT

Conference papers

School of Computing

2011-01-01

Accessible Rich Internet Applications: the Search Engine Challenge

Angela Kielthy
Dublin Institute of Technology, angela.kielthy@gmail.com

Follow this and additional works at: http://arrow.dit.ie/scschcomcon

Part of the Computer Engineering Commons

Recommended Citation

Kielthy, A.: Accessible Rich Internet Applications: The Search Engine Challenge. The IEEE Conference on Communication, Science & Information Engineering (CCSIE), 2011.

This Conference Paper is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License

Accessible Rich Internet Applications: The Search Engine Challenge

Angela Kielthy
School of Computing
Dublin Institute of Technology
Kevin Street, Dublin 8, Ireland
angelasaria@gmail.com

***Abstract*—The perception that Rich Internet Applications (RIAs) and Accessible Rich Internet Applications (ARIAs) are inaccessible to search engines is perhaps one of the main factors that hinder their wider adoption by the web development community. Recent announcements that RIAs and ARIAs are becoming more search engine friendly is provoking web developers to look for further information and evidence that will support or refute these announcements.**

This paper outlines research undertaken and tests performed to establish if RIAs and ARIAs developed using Adobe Flex are crawlable and indexable by the Google search engine by default.

The conclusion drawn from testing is that RIAs and ARIAs are not yet fully supported by the Google search engine. They can however be made search engine friendly by employing third party software and some imaginative coding techniques.

This conclusion contradicts various published statements from search engine providers such as Google, RIA software providers such as Adobe and numerous field experts.

Keywords-Rich Internet Application; accessible; search engine; crawl; index; SEO; Adobe; Flex; Google

I. INTRODUCTION

A web developer's finished product often has a very broad audience, and is targeted at any person with an internet connection. For web developers to be sure that their website is available to this wide audience, they must ensure that their website can appear in search engine result pages (SERPs) when relevant key words are entered into the search engine by the user [1].

Web developers must carefully consider the technology to be used to develop the website to ensure that this technology does not hinder the website's ability to appear in SERPs.

Rich Internet Application (RIA) technologies have in the past prevented search engines from extracting content as they are often compiled into a binary format. RIAs however build more interactive websites and so are still used by many developers, even if their potential audience is reduced.

Recently there have been developments made with RIAs built using Adobe Flash & Flex technology that has resulted in the Google search engine being able to include RIAs in SERPs. There is however still very little information available about how these RIAs are included. There are also a lot of reports of RIAs being excluded from SERPs with no explanation given by Google or Adobe as to why.

This level of uncertainty leaves some developers refusing to use the technology 'just in case', while others are taking a leap of faith using it only to be denied the benefits promised with no instruction given on how to remedy the situation.

In an endeavour to bring clarity to this area, material on the subject was gathered and studied with the knowledge gained from this research used to build applications that could be tested. These applications bring definitive results that contradict the claims made of compatibility between Adobe Flash applications and the Google search engine [2].

An alternative means of rendering RIAs is suggested that will allow text content of a Flex RIA to be returned in SERPs, without relying on the uncertain ability of the search engine to interpret the content within the RIA itself, and also not just in the Google search engine.

II. CHALLENGING QUESTIONS

Before undertaking a new project, web developers must ask themselves if the technology they are about to choose is suitable to the project at hand and if picking that technology will have any consequences or overhead. The following three questions are ones that a web developer must know the answers of before choosing to use Adobe Flex on their next project.

A. Are Rich Internet Applications crawlable by search engines?

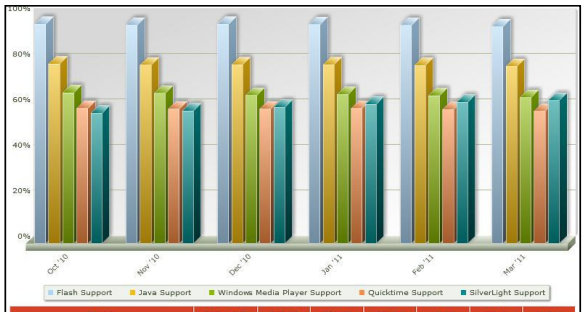
In 2008 both Adobe and Google announced that existing Adobe Flash and Flex content are now searchable using the Google search engine [2][3]. This implies that RIAs compiled into the SWF file format should now be automatically crawled and indexed by the Google search engine. Is this true?

B. Are Accessible Rich Internet Applications crawlable by search engines?

Search engines can in many ways be compared to Assistive Technologies (ATs) [5]. In particular they can be likened to a

This research has been funded by Carl Stuart Group
www.labunlimited.com

screen reader, which stores the information it encounters before converting it to spoken words. On this premise, can an Accessible Rich Internet Application (ARIA) be crawled and indexed by Google?



PLUGIN	TOTAL AVG	OCT '10	NOV '10	DEC '10	JAN '11	FEB '11	MAR '11
Flash Support	95.95%	96.28%	95.89%	96.30%	96.18%	95.84%	95.22%
Java Support	78.39%	78.91%	78.38%	78.52%	78.54%	78.26%	77.79%
Windows Media Player Support	65.28%	66.26%	66.00%	65.03%	65.47%	64.95%	64.13%
Quicktime Support	58.96%	59.38%	59.08%	59.09%	59.32%	58.86%	58.10%
SilverLight Support	60.32%	57.21%	58.23%	59.80%	61.19%	62.07%	62.82%

Figure 1. Adobe Flash Plug-in Market Share Oct 2010 to March 2011 [4]

C. *How can Rich Internet Applications be made crawlable by search engines?*

What measures can be taken to overcome the challenges encountered by RIAs & ARIAs with search engines? Are there techniques that can be implemented that will allow the search engine crawl the content of the RIA if it cannot interpret the RIA itself?

This paper addresses each of these questions and provides answers based on research and testing. Recommendations are made for overcoming the challenges that RIA and ARIA technologies present.

III. Rich Internet Application

RIAs are websites or portions of websites that provide a rich experience to the user. The concept of richness in RIAs extends the traditional web in three aspects: data, presentation & communication capabilities [6].

Typically a RIA is loaded by the client along with some initial data into a browser plug-in, independent sandbox or virtual machine. The RIA then manages data rendering and event processing, communicating with the server when the user requires or submits data [7]. RIAs combine the best user interface (UI) functionality of desktop applications with the broad reach and low-cost deployment of web applications as well as the best of interactive, multimedia communication.

A. *Adobe Flash / Flex RIAs*

RIAs can be developed using Adobe Flash Professional or the Adobe Flex Software Development Kit (SDK) as both compile their programs into the SWF file format. Adobe Flex was introduced in 2004 [8] to make it possible for developers to create RIAs for the nearly ubiquitous Adobe Flash Player (see Fig. 1) without the steep learning curve many experienced with the Adobe Flash Professional environment which works on a timeline and is mainly intended for animation.

IV. Accessible Rich Internet Application

People with different disabilities have different requirements from RIAs. ATs are often used by people with disabilities to interpret and interact with a website as the AT can transform the presentation of content into a format more suitable for the user [9].

A. *Adobe Flex ARIAs*

One drawback of using Flex as a technology for building RIAs is that it can sometimes cause problems for ATs or users who may use conventional technologies but with limitations, i.e. accessing a website through the use of the keyboard alone.

ATs obtain information about a Flex application from the Adobe Flash Player instance in which the application is executing. This information is provided via the Microsoft Active Accessibility (MSAA) Application Programming Interface (API), and Flex developers must take explicit steps to make accessibility information available to ATs.

The Adobe Corporation is a strong promoter of the benefits of ARIAs and participated in the publication of the Web Content Accessibility Guidelines (WCAG) 2.0 [10]. They also produce documents of best practices such as “Accessibility best practices for Flex” [11] and “Best practices for accessibility with Adobe Flex 4” [12] which was published in March 2011.

B. *Testing Adobe Flex ARIAs*

The most common method for testing the accessibility of web applications is to use ATs to determine the ease of interaction a disabled person would experience. This however poses the problem that ATs are not designed as testing tools and so do not provide feedback to developers that can help in identifying and fixing potential problems.

Developers of Adobe Flex ARIAs can use aDesigner [13] to inspect the Flex application as it executes and validate that it is exporting proper information via the MSAA API. This testing approach enables developers to verify that core MSAA information is being made available by the Flex application.

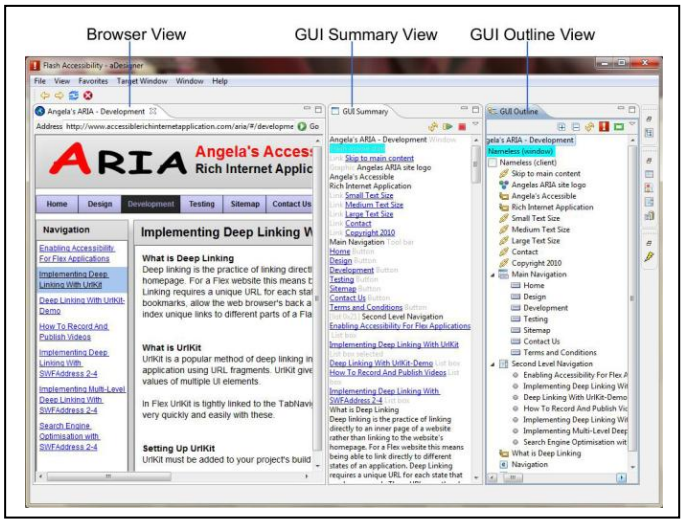


Figure 2. aDesigner Flash accessibility tester

Fig. 2 shows aDesigner in use and identifies its main areas. The Browser view shows how the application will appear in the browser, the GUI Summary view shows how the content will be read by a screen reader and the GUI Outline view shows the order in which elements will be identified and the names that they will be identified with to ATs.

V. SEARCH ENGINES

An internet search engine is an information retrieval system designed for searching and holding information from the internet. The search engine traverses the internet using web page addresses that it knows about, collecting the information held at these pages and following any links to find more pages with information to be collected. This process is called crawling.

The content amassed by these search engines is processed and stored in a database. This is called indexing and the database is referred to as an index database. These index databases are usually sorted alphabetically by search term with each index entry storing a list of documents in which the term appears and the location within the text where it appears.

A user can interact with the index database through a front end such as the Google or Yahoo! search engine web pages. When a search term is typed into the search box the search engine queries the index database and returns details of websites that it believes are the best matches to the search term entered. This interaction is illustrated in Fig. 3.

A. *Search Engines and RIAs*

In general, search engines are text based. This means that in order for content on a web page to be crawled and indexed, this content needs to be in a text format.

Traditional web pages are created using HTML which the search engine can read. It can interpret the HTML tags to understand which words are headings and which words have an emphasis applied. The text and the knowledge that these tags convey are added to the index database.

Adobe Flex content is delivered in SWF files which are compiled files. Accessing the textual content within these files is harder to do. The text content also does not have the same structure as HTML and so search engines do not have the benefit of tags to associate importance to sections of text. There have however been reports of improvements made in the indexing of Flash content held in SWF files.

B. *Google Indexing Of Adobe Flash / Flex Content*

Google announced in June 2008 [14] that they had completed development of a new algorithm for indexing textual content in SWF files and an integration with Adobe's headless Flash Player technology codenamed "Ichabod" [2].

The Ichabod Player runs the Flash or Flex application similarly to how it would be executed in a browser, except that it returns all text and link content that occur at any state of the application back to the search engine which can then index this content.

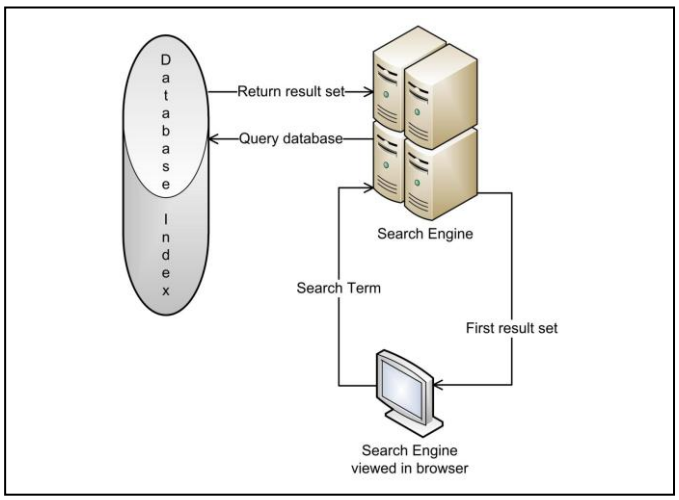


Figure 3. Search engine interaction

Further improvements were announced in 2009 [15] when Google revealed that externally loaded resources such as XML could now be indexed in context with where they were found.

The most recent announcement was in November 2010 [16] with an article that outlined improvements made in Google's ability to index Flash content, specifically relating to content designed to run in the latest version of Flash Player.

C. *Statistics of Google Indexing Adobe Flash / Flex Content*

The Google search engine currently returns approximately 156 million results[1] for the Adobe Flash and Flex file type 'SWF'. This shows that a lot of Flash and Flex content has been identified by the Google search engine.

There are however approximately 240,000 entries in the Google index that are of the type SWF and contain the exact term 'Loading Loading'. This shows that while some Flash files are being found, the content within is not being properly rendered to the search engine. In the book "Search engine optimisation for Flash" [17] (which was written in conjunction with Adobe) this is attributed to applications dynamically loading the content of the application, with the 'Loading' text being the only static content within the application that is displayed while the dynamic content is retrieved.

While there is no exact search that can display a definitive or approximate number, testing has further shown that there are also SWF files in the Google search index that have no content associated with them whatsoever. These files have been identified as the SWF file type by the search engine but the content within them could not be crawled.

These results throw shadows over the claims made by Google and Adobe about the ability of SWF file content to be indexed. Two direct quotes to this are:

"any type of SWF content including Adobe Flex applications and SWF created by Adobe Flash authoring will benefit from improved indexing and search results" Adobe 2008 [2].

[1] As of April 2011

"If you have Flash content on your website, we will automatically begin to index it" Google 2008 [14].

D. No Documentation for Better Crawling and Indexing

There are still many questions about how SWF files are crawled and indexed by search engines. And even more questions about what a developer can do to ensure their Flex website is crawled and indexed.

There are currently no documented guidelines available on how to build a RIA that search engines will be able to crawl.

Adobe has taken the standpoint that as with HTML content, best practices will emerge over time [2] while Google give no guidance on how to optimise a Flash or Flex application, but merely suggest that HTML equivalent content can be provided just in case there are difficulties encountered [18].

Many other resources such as books, papers, press releases, blogs and forums have been consulted in an attempt to uncover best practices for ensuring a crawlable RIA but none have been found.

E. Search Engines and Assistive Technologies

While each search engine is different in how it accesses a web page, and many of these techniques are proprietary, the commonalities they all share are that they cannot hear sound, interpret images or videos, see colours and most have limited capabilities with technologies used to create dynamic page content such as JavaScript. These limitations of search engines are very similar to the limitations that disabled users may encounter when using ATs to interact with a web page.

To this end, making a website accessible to all human users, regardless of the AT used to access the site should automatically make it more accessible to search engines.

This theory is supported by Andy Hagans who in his 2005 article "High Accessibility is Effective Search Engine Optimisation" [5] said that the goal of accessibility is to make web content accessible to as many people as possible, and that search engines can be thought of as users with substantial constraints. The constraints identified by Hagans included the inability to read text in images, to interpret JavaScript or applets, or view many other kinds of multimedia content. Hagans summarises that these are the types of problems that accessibility is supposed to solve in the first place.

F. Search Engines and WCAG 2.0

When the WCAG 2.0 is considered with search engines in mind, many of the requirements for users bear resemblance to Search Engine Optimisation (SEO) techniques that would be employed to ensure search engines have the greatest understanding and interaction with a web page.

Of the 38 WCAG 2.0 level 1 and level 2 guidelines, 23 have been identified as being beneficial for SEO also.

These observations made about the similarities between search engines and ATs have lead to the previously unasked question of 'Does an ARIA enjoy better search engine support than a RIA?'

VI. TESTING FOR SEARCH ENGINE CRAWLABILITY

There is a lack of tools available to test how a RIA or an ARIA will perform when interacted with by a search engine. Many of the tools available for SEO try to simulate how a web page will appear to a search engine, but because search engines are proprietary pieces of software these SEO tools are at best a guess of how the website will be rendered or interpreted.

The only way to truly test how a RIA or ARIA will perform in SERPs is to host the application on the internet, make the search engine aware of the web page where the application is hosted and then evaluate the SERPs once the search engine has crawled and indexed the page.

To this end, the applications that were developed for testing were hosted on www.accessiblerichinternetapplication.com. The interactions between search engines and the applications were monitored and evaluated, and the conclusions in this report are based on these interactions.

VII. RESULTS OF TESTING

As outlined in section II, there are three questions being asked and answered in this report. To answer these questions a single application was built that evolved and changed through the testing process.

The application built is a knowledge base website that holds information, recommendations and tutorials on building ARIAs using Adobe Flex that can be found in SERPs. This application consists of three main content sections: Design, Development and Testing. The content in these sections are delivered in text, images and subtitled videos.

The testing performed on this application was recursive; when a result was reached it was examined to see if a better result could be achieved. The sections that follow outline the three stages of development and testing that correlate to the three questions being asked by this paper.

A. Are Rich Internet Applications crawlable by search engines?

To answer this question a RIA was developed using Adobe Flex. The application was built using methods described in tutorials on the Adobe website [19] and in reference books [1] [8] [17] [20] [21]. These sources were used to ensure the application being built would be typical of applications being built by other developers.

When this application was hosted on the domain www.accessiblerichinternatapplication.com it began appearing in Google SERPs within a few days. It became apparent however that the search engine was only able to crawl and index the HTML page that the Flex application was embedded in and not the application itself.

B. Are Accessible Rich Internet Applications crawlable by search engines?

For this stage of the development, the application previously built was redesigned to comply with accessibility best practices published by Adobe [11] and the W3C [10] [22] to ensure the application was fully accessible to ATs, including

screen readers. Testing was performed using aDesigner which allows developers to examine the content that is presented to the AT during execution of the application.

This application replaced the first application, however Google SERPs once again showed that the HTML page was being crawled and indexed by the search engine, but the Flex application itself was not.

C. How can Rich Internet Applications be made crawlable by search engines?

The changes made during this redevelopment iteration saw no changes made to the Flex application itself, but instead changes were made to the HTML page that rendered the ARIA so that alternative content could be provided when different URLs were used to load the application. This change in strategy builds on the previous results that show that the HTML content is easily consumed by the search engine.

This application uses SEO techniques that work with the open source deep linking library for Adobe Flex called SWFAddress [23]. SWFAddress gives the ability to build a single HTML page using PHP that will render different HTML content based on the URL used to load the website. The same Flex application is always loaded, but again SWFAddress can interpret the URL and render the application with a specific view as dictated by the URL.

This website replaced the one already on www.accessiblerichinternatapplication.com. This new version was soon crawled by the Google search engine which found many pages of content that it began including in its index.

This application also enjoyed the added benefit of appearing in SERPs of other search engines such as Yahoo! and Bing that do not support Adobe Flex RIAs.

D. Complementary testing

Smaller test applications were developed in an attempt to uncover reasons why some Adobe Flex developers reported success while others reported failure. These tests show that applications built using older Flex SDKs are crawlable by the Google search engine, while the newest SDK 4 is not. Further testing showed that even with using these earlier SDKs, text content held in external files that are loaded at run time are not crawlable by the Google search engine.

VIII. CONCLUSIONS

Testing performed on applications built using Adobe published or endorsed material using the Google search engine show that the content of the RIAs and ARIAs cannot be crawled by the search engine by default.

RIAs and ARIAs can be made to return in SERPs with additional planning and work and the use of the open source library SWFAddress. The correct use of this library greatly improves a websites performance in all SERPs, not just with the Google search engine.

REFERENCES

[1] S. Johnston, L. McGee, 50 Ways to Make Google Love Your Website, Random House Business Books, London, SW1V 2SA, [ISBN: 978-1-905-21125-8], 2010.

[2] Adobe "SWF Searchability FAQ: Enhanced search indexing of SWF content", www.adobe.com/devnet/flashplayer/articles/swf_searchability.html, (accessed March 2011).

[3] Google "Google learns to crawl Flash", googleblog.blogspot.com/2008/06/google-learns-to-crawl-flash.html (accessed March 2011).

[4] StatOwl "Rich Internet Application Usage", www.statowl.com/custom_ria_market_penetration.php (accessed April 2011).

[5] A. Hagans, "High Accessibility is Effective Search Engine Optimisation", www.alistapart.com/articles/accessibilityseo (accessed October 2010).

[6] R. Morales-Chaparro, M. Linaje, J.C. Preciado, F. Sanchez-Figueroa, "MVC Web design patterns and Rich Internet Applications", Quercos Software Engineering Group, Universidad de Extremadura, 2008.

[7] A. Bozzon, P. Fraternali, S. Comai, G. Toffetti Carughi, "Capturing RIA Concepts in a Web Modeling Language" Dipartimento di Elettronico e Informazione, Politecnico di Milano, Milano, Italy, 2006.

[8] J. Tapper, M. Labriola, M. Boles, J. Talbot, Adobe Flex 3 Training from the source, Adobe Press book, Peachpit publisher, Berkeley, CA 94710, [ISBN: 978-0-321-52918-3], 2008.

[9] M. LaPlante, G. Hendershot, A. Moss, "Assistive Technology Devices and Home Accessibility Features: Prevalence, Payment, Need, and Trends", Advanced Data from Vital and Health Statistics; Number 217 Sept 1992.

[10] W3C, "Web Content Accessibility Guidelines (WCAG) 2.0," www.w3.org/TR/WCAG20 (accessed April 2011).

[11] Adobe "Accessibility best practices for Flex", www.adobe.com/accessibility/products/flex/best_practices.html (accessed October 2010).

[12] Adobe, "Best Practices for accessibility with Adobe Flex 4", Adobe System Incorporated, San Jose, CA 95110-2704, 2011.

[13] aDesigner, www.eclipse.org/actf/downloads/tools/aDesigner (accessed March 2011).

[14] R. Adler, J. Stipins, "Google learns to crawl Flash", googleblog.blogspot.com/2008/06/google-learns-to-crawl-flash.html (accessed April 2011).

[15] Google, "Flash indexing with external resource loading" googlewebmastercentral.blogspot.com/2009/06/flash-indexing-with-external-resource.html (accessed April 2011).

[16] Google, "What a feeling! Even better indexing of SWF content", googlewebmastercentral.blogspot.com/2010/11/what-feeling-even-better-indexing-of.html (accessed April 2011).

[17] T. Perkins, Search Engine Optimization for Flash: Best Practices for using Flash on the Web, O'Reilly Media Inc, Sebastopol, CA 95472, [ISN 978-0-596-52252-0], 2009.

[18] Google, "Flash and other rich media files", www.google.com/support/webmasters/bin/answer.py?answer=72746 (accessed June 2011).

[19] Adobe, Flex Devloper Center, http://www.adobe.com/devnet/flex.html (accessed June 2011).

[20] A. Cole, Learning Flex 3: getting up to speed with Rich Internet Applications, O'Reilly Media, Inc, Sebastopol, CA 95472. [ISBN: 978-0-596-51732-8], 2008.

[21] C. Kazoun, J. Lott, Programming Flex 3 O'Reilly Media, Inc, Sebastopol, CA 95472. [ISBN: 978-0-596-51621-5], 2008.

[22] W3C, "Flash Techniques for WCAG 2.0", www.w3.org/TR/WCAG20-TECHS/flash.html (accessed March 2011).

[23] SWFAddress, www.asual.com/swfaddress (accessed April 2011).