

Yin Tang. Making Sense of the Browser Jungle. A Master's paper for the M.S. in Information Science degree. 65 pages. April, 2003. Advisor: Gary Marchionini.

As the popularity of the World Wide Web increases, newer and better web browsers are continually emerging. The study described in the paper attempts to develop a standard framework for the purpose of evaluating and comparing web browsers as they become available. The framework criteria identified include performance, security, user interface design, usability, and Auxiliary Functionalities. Using these criteria, the author compared the new Safari web browser (public beta) by Apple Inc. with the more popular IE Mac 5 and the Camino browsers. The results indicate that using the framework makes different browsers more comparable and also allows the reviewer to provide users with a more complete picture of the products.

Headings:

WWW Browsers

Web - Browser Evaluation

Web - Browser Usability

Web - Browser Comparison

MAKING SENSE OF THE BROWSER JUNGLE

by
Yin Tang

A master paper submitted to the faculty
of the school of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science

Chapel Hill, North Carolina

April, 2003

Approved by:

Advisor

TABLE OF CONTENTS

INTRODUCTION	2
WWW AND WEB BROWSERS	4
CRITERIA FOR ASSESSING WEB BROWSERS	8
METHODOLOGY FOR IDENTIFYING THE CRITERIA	8
CRITERIA FOR ASSESSING WEB BROWSERS IDENTIFIED	8
1. <i>Performance</i>	8
2. <i>Security</i>	13
3. <i>User Interface Design / Usability</i>	14
4. <i>Auxiliary Functionalities</i>	23
COMPARING SAFARI AND ITS COMPETITORS USING THE CRITERIA	25
OVERVIEW	25
RESULTS	25
1. <i>Performance</i>	26
2. <i>Security</i>	31
3. <i>User Interface / Usability</i>	35
4. <i>Auxiliary Functionalities</i>	51
CONCLUSION	53
SUMMARY	55
BIBLIOGRAPHY	56
Appendix 1. Screenshot of the Rendering Speed Test	58
Appendix 2. Javascript Source Code of the Rendering Speed Test	59
Appendix 3. Complete Results of a Rendering Speed Test for Selected Mac Os X Browsers	60
Appendix 4. Displays of the espn.com 2003 Men's NCAA Tournament Bracket in Safari, Camino and IE Mac	61

INTRODUCTION

In January of 2003, Apple Computer unveiled its self-developed web browser called Safari. The company also claimed that this new browser, which has a very minimal user interface, is the fastest available. And in about three days after the free beta version of the Safari web browser became available to public, users had downloaded more than 500,000 copies.

When I first learned about the Safari browser, I was immediately attracted by its simple yet charismatic interface. Without much hesitation, I downloaded it and made it the default browser on my PowerBook. Although my initial experiences with the Safari browser are mostly satisfactory, I realized I am just one user using a subset of its many features. I was interested in finding out overall how it compares to other browsers in a systematic and comprehensive manner. However, since it is such a new product, I was unable to find many authoritative reviews of the browser at the time this paper was written. And also, to my surprise, comparisons and reviews of some long existing web browsers did not appear to be based on a standard framework either. Most of these review articles deal with only some aspects of browser features, and it is hard to figure out how browsers are compared to each other on the whole.

The study described in this paper attempts to first identify a standard set of criteria for assessing web browsers. Then using the criteria, the new Safari browser is compared with currently the two most popular browsers for the same MAC OS X platform, Microsoft Internet Explorer (IE) and Camino (formally know as Chimra).

WWW AND WEB BROWSERS

The idea of cross-referencing within and across documents was first explained in Vannervar Bush's famous 1945 article "As We May Think" (Bush, 1945). In the article, he described his vision of a desktop information-exploration tool, *memex*, which uses microfilm and eye-tracking technology. Memex allows readers to follow cross-references by merely staring at them. In the 1960s, Ted Nelson coined the term "hypertext" and defined it in his self-published *Literary Machines* as "non-sequential writing" (Nelson, 1965). At about the same time, Douglas Engelbart at the Stanford Research Laboratories implemented his own vision of the same idea and created an "oN Line System" (NLS) that consisted of a keyboard, monitors and a mouse and made extensive use of hyperlinking (Engelbart and English, 1968). Throughout the 60s, 70 and early 80s, many other hypertext systems were developed by various groups. Some of the ones that are worth noting include the University of Vermont's Patient Record and Outcome Management Information System (PROMIS), which was the first Hypertext system released to the user community (Myers, 1998); Ben Shneiderman's Hyperties, which was the first system where highlighted items in the text could be clicked on to go to other pages (Koved, 1977); and HyperCard from Apple (1988), which brought the idea to a wide audience.

In 1990, based on the hypertext idea that was originated about 40 years earlier, Tim Berners-Lee create the first Internet based hypertext system, also known by its more popular name World Wide Web (WWW), at the government-funded European Particle Physics Laboratory (CERN) (Cailliau, 1995). The WWW implementation copied the key innovation of having highlighted link words embedded in the text, so that with a simple mouse click, users can jump to related web pages, which may be delivered from millions of server computers located around the world. The next big event in the WWW history came three years later, as Marc Andreessen, an undergraduate student at the University of Illinois at Urbana-Champaign, worked on a project for the National Center for Supercomputing Applications ([NCSA](#)) and led a team that developed the graphic interface browser called Mosaic (Cailliau, 1995). The first pre-Beta version of Mosaic was released in February 1993. Version 1.0 of Mosaic (Figure 1) was released in September 1993 for the Windows, Macintosh and the X Windows System platforms.

The popularity of the graphical user interface (GUI) browser was immediate. People without computer expertise were able to use the graphical interface and just point and click to navigate the World Wide Web. The WWW grew quickly. Marc Andreessen left NCSA in March 1994. He and Jim Clark formed a company later known as Netscape Communication Corporation. For years, Netscape Navigator, a Mosaic based browser, was the undisputed leader in the browser market with over 80% of the market share in 1995. Microsoft Internet Explorer (IE), which later became its major rival competitor, had only 2.9% of the market at the time (Festa, 1997). Browser technologies grew quickly over the years and the market became highly competitive. Four years after the first browser was introduced to the public, Netscape Navigator remained the market

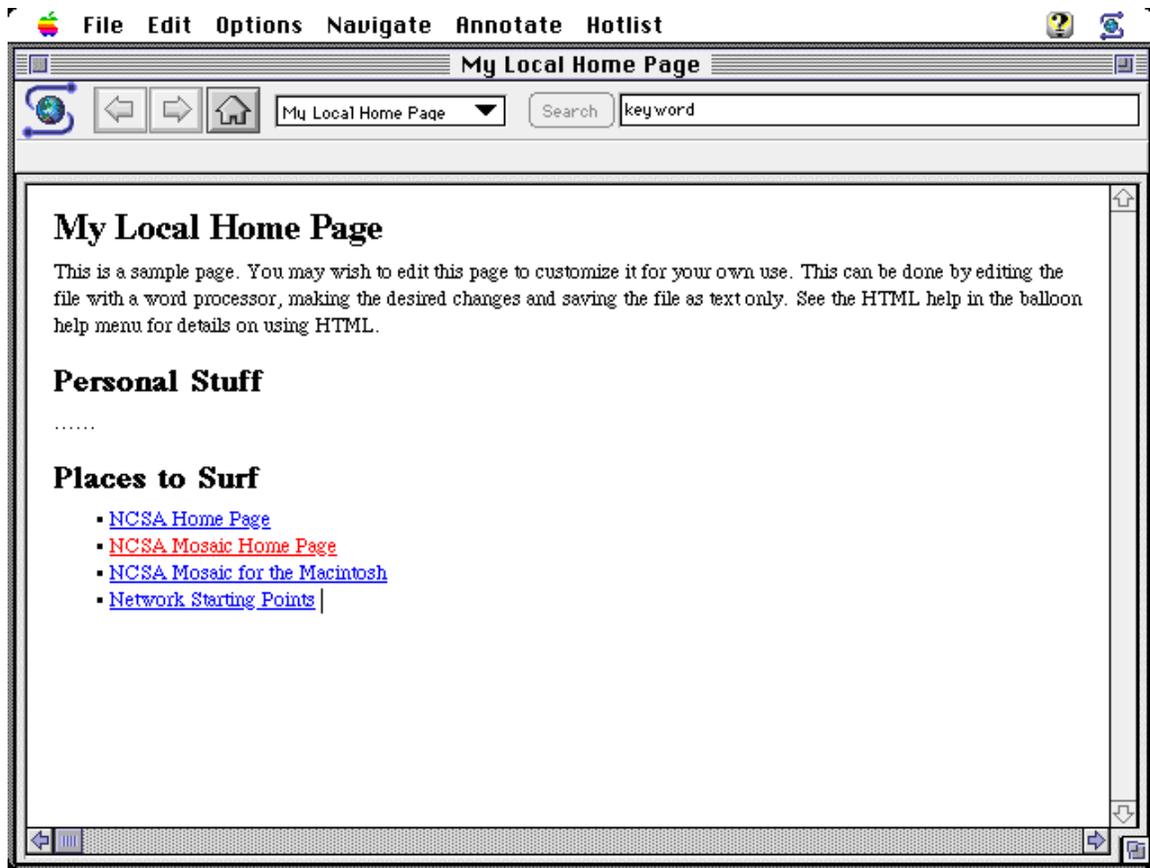


Figure 1. The first WWW browser

leader, but its share fell below 50% to 41.5% for the first time in July of 1998. At the same time, Microsoft IE browser popularity experienced a sharp increase and raised its market share to 27.5%. Another major competitor in the market then was the America Online (AOL) browser, which is an AOL flavored IE that had 16.2% of the market share (Festa, 1998). Many other browsers had been developed and matured over this period too. A comprehensive list of WWW browsers can be found at the evolt.org's web site (<http://browsers.evolt.org/>). One of them that made it to the top five most popular browsers is the Opera browser (www.opera.com). It was first released in 1996 and was marketed for its fast speed. The company is still in browser business today, but with less than 1% of the market share, it has yet to become a major player in the browser war.

Another browser that made it to the top-five list is the Mozilla browser, the product of an open source project. Its main targeted users are developers and it has less than 1% of the market share. Today, Netscape continues to lose momentum to Microsoft, and at the time this paper was written, it was only used by 1% of the global users while various versions of IE were used by over 95% of the users.

CRITERIA FOR ASSESSING WEB BROWSERS

Methodology for identifying the criteria

Online web browser reviews and web browser related journal articles were collected. The criteria for assessing browsers were extracted from these sources and then consolidated into a framework for comparing and evaluating browsers. The framework specified below is intended to be a comprehensive one, not only including the criteria recognized from the end users point of view, but also those of web developers and system administrators.

Criteria for assessing web browsers identified

1. Performance

Performance has three elements: speed, stability and compatibility. The details of each are described below.

□ Speed

One important aspect of browser performance is its speed. This is measured at several different levels:

- A browser *cold launch speed* refers to the time it takes for a browser application to start up. This can be measured without loading any web

content in the browser. Usually, a 'light weight' browser application loads faster than a 'heavy weight' one.

- A browser's *download speed* affects how fast it downloads files from the server to the client's local system. The effect of different download speeds may not be obvious when a user is just viewing plain HTML pages, but it becomes important when he/she tries to download large software installation files from the Internet.
- Since a browser's most basic job is to display web pages, the speed at which HTML pages are rendered (*rendering speed*) is also an important indicator of its performance. However, this measurement is hard to obtain because as a web page on the Internet is displayed in a browser, downloading and rendering are happening at the same time. That's why in most of the reviews, when it comes to browser speed, usually the time it takes to display HTML pages is presented instead of just rendering time. It is worth pointing out that when these test results are compared, one should differentiate the speed of loading a page for the first time versus that of loading a cached page since the latter reflects only the rendering time, and no downloading time is included.
- Since the development of the JAVA programming language in 1995, more and more web sites are employing JAVA technologies to offer more sophisticated features. This makes the *speed of the browser's Java Virtual Machine (JVM)* increasingly important too. For the purpose of testing a

browser's JVM, Pendragon's CaffeineMark 3.0, a series of nine tests that evaluate JVM, is available.

The measured performance of a browser varies a great deal depending on the hardware configuration and the network speed. Because both of these advance rapidly as technologies press forward, comparing browser speeds observed in different tests can be difficult because of the lack of a common benchmark.

□ **Stability**

More than speed, stability is a critical factor of a browser's performance. In today's world, web browsers are probably one of the most important applications on people's personal computers. People use it to get news, handle finances, and order products etc. Often, people will perform a number of tasks with a browser simultaneously. It is not hard to imagine the users' frustration and resentment when their browsers crash, especially in the middle of a transaction and all the data are lost. Frequent crashes can easily turn users away to a competing browser.

Browser crash is just one aspect of browser stability. When a browser's stability is assessed, one should also take into account the overall performance degradation of the computer system as a result of running a browser. For example, some poorly designed browser programs may use unnecessary large amounts of system memory and cause the system to slow down, or even to crash. The author was not able to find any standard benchmarks for stress testing

browser stability, or any quantitative way to measure browser stability. Perhaps one possible way to quantify this could be, under clearly defined standard conditions, counting the number of browser crashes in units of a hundred hours of use. And perhaps another way would be to develop a set of different hardware configurations (minimum RAM, disk, etc.) and compare effects on browsers' stabilities.

□ **Compatibility**

A browser's most basic job is to accurately display web-based information. Therefore compatibility is probably the most essential aspect of a browser's performance. Despite lightening speed and fancy features, a browser unable to render web pages accurately is useless to a user. There are wide varieties of tools to test how a browser complies with the current standards. However, these tools only give a snapshot of what is available at the current moment, because new web related standards are developed continuously and so are new web browsers. "With new browsers come new features, but also new bugs and incompatibilities." (Franklin, 2000). A common set of current standards includes, but is not limited to:

- HTML 3.2
- HTML 4.01
- XML/XSL
- XHTML 1.0
- CSS
- ASP
- PNG
- Frames
- I-Frames
- FTP (for downloading) - anonymous FTP and Authenticated FTP
- Java 1.4.0
- ActiveX
- Javascript
- Media file formats

- Plug-in / multi-media: RealPlayer, Quicktime, Flash, ShockWave, Windows Media Player, Acrobat Reader
- Unicode char support (International support bi-directional text)

Despite the presence of these standards, differences among browsers exist. It is, of course, not only the browser developers' responsibility to comply with these standards, web developers have the same responsibility as well. There are numerous third party products that claim they will assist web developers to create web sites that are viewable via any browser. There are also many valuable resources, from good practices to browser compatibility charts, on the Internet that aim to help web developers identifying the specific standard compliance of each browser. It is not uncommon for a browser to claim support for a certain technology yet not properly display a web site with that technology.

Testing whether a browser supports Internet standards can be difficult because poorly coded pages could also be the cause of incorrect rendering in browsers. The only way to find out if a browser supports a certain standard is to have it display a page that is known to comply with that standard. Luckily, the W3C has many test suites to test various Internet standards with. Currently, there are test suites for testing browser support for Cascade Style Sheet (CSS) level 1, CSS level 2, CSS level 3 Selectors, DOM level 1, HTML 4.01, http 1.1, PNG, XHTML 1.0, XHTML 1.1, XHTML Basic, XML 1.0, XML schema, XSL 1.0, XSTL 1.0 and many more (W3C Quality Assurance, 2003). Numerous other third party test suites can also be found on the Internet, but the ones from W3C are the most comprehensive and needless to say, the most authoritative as well.

2. Security

Today, more and more people are doing financial transactions, such as shopping, banking, and even tax filing, online. But at the same time, despite the conveniences the web has to offer, a lot of people are reluctant to expose any personal or confidential information online because they are concerned about web security. Generally speaking, there are three elements to web security, web security at the server level, at the networking level and at the client level. All three parts are equally important. The sensitive information a user shares with a web site is only as safe as the least secure part of the three. Because browsers are the focus of this paper, only web browser security will be discussed in detail here.

Client side security risks are often over looked because end users usually feel both safe and anonymous about web surfing. (Stein et al., 2003). It's not. Active content, such as ActiveX controls and Java applets, introduce the possibility that Web browsing will bring viruses or other malicious software into the user's system. Even without active content, the very act of browsing leaves an electronic record of the user's surfing history, from which unscrupulous individuals can reconstruct a very accurate profile of the user's tastes and habits. Some security related browser features to consider are:

- Is 128 bit security available?
- Is it able to manage individual certificates?
- Is it able to manage all certificates?
- Is it able to configure proxy server
- Does it support personal digital IDs?

- What kind of cookie management does it have?

3. User Interface Design / Usability

□ A brief overview of Human Computer Interaction (HCI) research

The rapid proliferation of interactive systems used by more and more people has resulted in an increasing interest in the quality of the user interface of interactive systems. Research in HCI has fundamentally changed the face of computing. One direct result of the HCI research is the spectacular growth of the World-Wide Web: applying hypertext technology to browsers allows one to traverse a link across the world with a click of the mouse. As suggested by the title, the study of HCI has three aspects, human, computer and interaction between the two. Specifically, the human refers to the end-user of a program, while the computer refers to the machine the program runs on. And the interaction between the two involves how the user 'tells' the computer what to do and the computer communicates results. Well-designed interactive computer systems should promote users' positive feelings of success, competence, and mastery and also allow users to concentrate on their work, rather than on the system. Explicitly, the objectives of HCI research are (Shneiderman , 1987) :

- *Decrease new users' time to learn to begin effective interaction with the system.*
- *Increase users' speed of performance.*
- *Decrease users' rate of errors*
- *Improve users' retention over time.*
- *Improve users' "satisfaction"*

It is important to note that although users share some common capabilities, diversities among individual users should be recognized. For the purpose of HCI research, users can be grouped into three categories: novice users, knowledgeable intermittent users and expert frequent users.

There are abundant resources, both online and in print, outlining good practices of user interface designs. Ben Shneiderman, a recognized expert in the field and the founding director of the HCI laboratory at the University of Maryland, has summarized these good practices into "Eight Golden Rules" [Shneiderman, 1987]:

Rule #1: Be consistent

Being consistent means sequences of actions, terminology, color, layout, capitalization, fonts, all should be consistent. Exceptions should be comprehensible and limited. Consistency allows skills learned once to be used universally. From the user's perspective, consistency also means "perceived stability". When faced with the complexity of a new application, users need stable reference points, for example, a standard graphic top menu bar. If users can re-use the knowledge they have, the time it takes for them to familiarize themselves with a new application is significantly shortened. For users with no prior computer interaction experience, an intuitive interface also means an interface that uses consistent and appropriate metaphors and terminology.

Consistency is measured at several levels:

- Is the system consistent within itself?
- Is the system consistent with its earlier versions?
- Is the system consistent with interface standards of the platform it runs on?

- Is the system consistent in its use of metaphors and people's expectations when compared to other similar applications?

Rule #2: Enable frequent users to use shortcuts

This is an ergonomic issue. Frequent users need a reduced number of interactions and increased pace of interaction. This rule emphasizes understanding the needs of the users. While shortcuts are an important part of this, it also means keeping the application accessible by all users. Or in other words, applications should be free of "cultural influences" and should provide access for people with various disabilities. Examples of implementations that follow this rule include mouse and keyboard shortcuts, abbreviations, hidden commands, and macro facilities. And as the technology improves, suitable shortcuts that actually improve efficiency need to be found, like for audio or motion-based input.

We can also interpret this rule as suggesting a good user interface should have multiple paths to accomplish the same task. So that users at different levels can take advantage of the approach that is best suited for them. A study (Nielsen, 1995) found that the popular Netscape browser provides nine different hypertext navigational mechanisms, including go to an absolute address (type in its URL), hypertext links (click on underlined text or an ISMAP) , breadcrumbs (underlined text changes color if you have seen the destination node it points to), landmarks (directory buttons like "What's Cool") , backtrack, history list, bookmarks, including bookmark annotation and structuring, prospective view (supported by showing a URL with path and filename in the footer before the

user has made the jump) , and the support for anti-homogeneity visual cues (supports the background attribute of the <body> tag to allow users to add background patterns to a web page). Although it looks like plenty of alternatives for a browser, Jakob Neilson pointed out that fifteen navigational support mechanisms are still missing, such as search, fat links (open many destination nodes at once) and others.

Rule #3: Offer informative feedback

Feedbacks give guidance and assurance to the users when they are performing tasks on a system. Offering informative feedback is most effective when modest feedback is given to frequent and minor actions, while substantial feedback is given to infrequent or major actions.

One of the ways to offer informative feedback is through giving users the ability to directly manipulate the graphical user interface objects, for example, drag-and-drop capabilities. Direct manipulation is generally described in the academic human interface literature (e.g., Schneiderman, 1983 & 1987; Hutchins et al., 1986; Norman, 1988; Rafaeli, 1990) as encompassing the following characteristics: (Heeter, 1991)

- *The objects and actions which can be applied to those objects are visible;*
- *The interface is transparent;*
- *The user interacts with objects instead of intermediaries;*
- *Using the interface feels like driving a car.*

Rule #4: Design dialogs to yield closure

In a nutshell, this means user actions should be divided up in small contingent handling sequences with a beginning, a middle part and an end. Sequences of actions should provide clear feedback upon completion.

Rule #5: Offer error prevention and simple error handling

This rule says the interface should be designed so that it constrains user actions to prevent error. In case an error is made, the system should be able to detect the error and allow for recovery. Warning messages should be displayed with reasonable frequency. Erroneous actions should leave application state unchanged, or in other words, the system should never crash.

Rule #6: Permit easy reversal of actions

Any one that has used the "Undo" feature of an application will be able to appreciate this rule. Giving user the ability to reverse his/her action encourages exploration of unfamiliar options. Whenever possible, the system should have multiple "levels" of reversibility. A first step towards easier reversal actions could be regular intermediate saves, suggested or automatically done.

Rule #7: Support user locus of control

This rule is all about user satisfaction. It suggests that the user should control the system and tell the system what to do, not vice versa. And on the hand, the system should not produce surprising actions or cryptic error messages, which causes anxiety and dissatisfaction to the user.

Rule #8: Reduce short-term memory load

There is a rule of thumb, that human can remember 'seven-plus or minus-two chunks' of information (like phone numbers) in the short term. So when it comes to designing user interfaces, it is a good idea to keep displays simple, and avoid a lot of multiple page screens. This way you will not stress the user's memory capacities with the handling of his/her tool.

Another major area of HCI research is the accessibility of the user interface. Has the GUI designer taken all types of user into consideration when the interface was designed? For example, is the interface friendly to color-blind users? Does it support speech input/output? How about internationalization?

□ **Criteria for assessing web browser HCI design**

Based on the objectives and principles of HCI design, the user interface and usability of a web browser can be evaluated based on the following four criteria.

1) Intuitiveness.

This criterion is concerned with how easy it is for a new user to learn a browser's interface and how visible the more advanced functions are. There is the famous quote that if the user cannot find it, then the function is not there. The intuitiveness of a browser is related to several HCI principles described above including consistency, use of informational feedback and direct manipulation, interface real estate management and over all look and feel.

2) Convenience

Convenience includes lots of different things. Depending on the user experience, convenience could mean offering short cuts, or having multiple navigational paths to the same destination. Offering the user the ability to customize the tool and menu bar can also be considered a convenience feature. The HCI golden rules discussed here include: enable frequent users to use shortcuts, permit easy reversal of actions, reduce short-term memory load, error prevention and simple error handling, and support user locus of control.

3) Accessibility

It is apparent that Internet has become an indispensable resource for scientific research, economic growth, prompting the attention of individuals, groups and governments all over the world. It enables people in all regions of the world to exchange various forms of information freely and quickly. Although it may be assumed that the benefits of the Internet are uniformly available to all people, unfortunately, the realities of the situation have proven otherwise. People with disabilities such as visual, mobility and language impairments, have special requirements when using the web. Specifically, these users have the following characteristics:

- Visual impairments include low vision, color blindness, and blindness. People who are blind cannot use a computer monitor and must receive information from their computers via another sense — hearing or touch. People with low vision can also receive information

through sound or touch, or they can modify their computer displays so the screen is more legible.

- Mobility impairments can be caused by a wide range of common illnesses and accidents such as arthritis, stroke, cerebral palsy, Parkinson's disease, multiple sclerosis, loss of limbs or digits, spinal cord injuries, and repetitive stress injury, among others. As a result of these accidents or conditions, individuals might be unable to use (or be without) arms or fingers to interact with their computers using the standard keyboard or mouse.
- Language impairments include conditions such as aphasia (loss or impairment of the power to use or comprehend words, often as a result of brain damage), delayed speech (a symptom of cognitive impairment), and other conditions resulting in difficulties remembering, solving problems, or perceiving sensory information. For people who have these impairments, complex or inconsistent visual displays or word choices can make using computers more difficult.

There have been numerous efforts by various organizations that aim to address this issue and to make possible for people with various forms of disabilities to access the information they need. One of these organizations is the User Agent Accessibility Guidelines Working Group (UAAG). This group produces guidelines for improving the accessibility of WWW user agents, applications that retrieve and render Web content, including text, graphics, sounds, video, images, etc. The most recent version of the guideline, the User Agent Accessibility Guidelines (UAAG)

1.0 (Jacobs, 2002), is currently a W3C recommendation. It covers a wide range of topics in the following twelve broad categories:

- 1) Support input and output device-independence (contains 3 checkpoints)
- 2) Ensure user access to all content (contains 10 checkpoints)
- 3) Allow configuration not to render some content that may reduce accessibility (contains 6 checkpoints)
- 4) Ensure user control of rendering (contains 14 checkpoints)
- 5) Ensure user control of user interface behavior (contains 5 checkpoints)
- 6) Implement interoperable application programming interfaces (contains 10 checkpoints)
- 7) Observe operating environment conventions (contains 4 checkpoints)
- 8) Implement specifications that benefit accessibility (contains 2 checkpoints)
- 9) Provide navigation mechanisms (contains 10 checkpoints)
- 10) Orient the user (contains 7 checkpoints)
- 11) Allow configuration and customization (contains 7 checkpoints)
- 12) Provide accessible user agent documentation and help (contains 5 checkpoints)

There are three types of checkpoints, priority one (P1), priority two (P2) and priority three (P3) checkpoints. A user agent's conformance level to the guideline is defined by a set of requirements based on checkpoint priorities. User agents that satisfy all P1 Checkpoints, receive a conformance level "A", and user agents that satisfy all P1 and P2 checkpoints, receive a conformance level "Double-A", while user agents that satisfy all P1, P2 and P3 agents receive a conformance level of "Triple-A".

Many of the things that are included in these twelve categories duplicate what is listed under the Intuitiveness and Convenience bullets, but here they are discussed from a slightly different view, the view of users with disabilities or language obstacles.

4. Auxiliary Functionalities

□ Integration with other applications

Many browsers are now offering features that go beyond basic browser functionalities. Examples include features like integrated email client, news group client, and instant messengers. Although these are not essential to what a web browser is intended to do, some users feel it is convenient to have one application take care of all their Internet needs.

□ Communication and collaboration features

Some browsers, such as Microsoft Internet Explorer, offer features that help to promote communication and collaboration among different groups in a corporate environment. An example of such features is the discussion functionality of IE 5 (Microsoft, 2002) that will let users add comments to any web page and then share those comments with others. This functionality, however, requires the existence of an IIS web server, and thus is out of the scope of this paper.

In addition, much effort has been made to bring collaboration and editing facility into web pages. Many shareware/freeware browser add-ons are developed and some browsers even have their own integrated tools that do the same job. Not everyone who uses a web browser will benefit from features like

these, but in some circumstances, they might be very valuable and should be taken into account when assessing browsers.

□ **Pop-up killer**

Anyone who surfs the web must have experience with the annoying pop-up advertisement windows. In response to the popular demand from end users, many browsers now offer a feature that will block these pop-ups. It saves the user from wasted bandwidth and CPU cycles.

COMPARING SAFARI AND ITS COMPETITORS USING THE CRITERIA

Overview

Because the new Safari browser from Apple Computer, Inc. was the original reason for the browser study described in this paper, it is chosen as an example to demonstrate how the browser-evaluating framework, which was identified as a result of the research, should be used. In the sections below, the Safari browser is compared against the more popular Microsoft Internet Explorer (IE) and Netscape Camino browsers. Because the comparison is intended to be an illustration of using the framework, it is not necessarily a scientific and precise analysis of the three browsers. Furthermore, because Safari is still in its Beta stage and is under constant upgrades and improvements, there are very little official published data on the browser's feature and performance numbers. As a result, the comparison in the paper is mainly based on the author's personal experience and opinions. Some user comments from several online forums are also included in the comparison. The versions of the three browsers are Safari Beta 1.0 v62¹, Camino version 0.7 and IE 5.2.2 for the Mac.

Results

¹ Safari Beta 1.0 V62 is a private Beta, obtained through a legal source.

1. Performance

□ Speed.

It was said that the top reason for developing the Safari browser is to build the fastest web browser on the Mac OS X platform (Melton, 2003). On the product's web site, the partial PC Magazine i-Bench benchmarks performance test results show Safari is much faster than its competitors in all three categories. However, it is well known that programmers optimize their software for the benchmarks. The author's everyday uses did not find any noticeable difference in the three browsers' speed. Therefore, more objective testing is needed.

From a user's point of view, the cold lunch speed is less important. In addition, to accurately measure cold lunch speed would require OS level programming tools and techniques, which is more than what this paper was intended for.

Testing the three browsers' JVM speed also turned out to be unnecessary. The well-known Caffeine Mark Benchmark does not work on the Mac OS X platform². As the author was searching for help to run the benchmark, a long time Mac enthusiastic pointed out that, unlike Windows platforms, browsers on the Mac OS X platform do not have their own JVMs, instead they share a common one provided by the operating system.

Downloading speed test was also omitted in this paper for the following reasons:

² A help message was sent to Apple Computer's tech support. No response was received.

- 1) It requires an isolated network with a web server setup to serve representative downloadable items. Additionally, measuring the download speed on the client side is too complex for web pages since each web page may have multiple connections to the server.
- 2) Downloading performance differs between one large-file download and many simultaneous small-file downloads, which is usually how a web page is served.

Because of the time limit of this paper, it was almost impossible to devise the suitable test with accurate measurement in place.

A simple rendering speed test was designed to compare the three browsers' speed to render a page from the local hard drive. The test was run on a PowerBook G4 with 800 MHz PowerPC CPU, 768 MB RAM, 40 GB IDE Hard drive, and Mac OS X 10.2.3.

To setup the test, a web page with contents in HTML, JavaScript, Flash, and various graphics was selected. In this test, the front page of www.msnbc.com on April 4th, 2003 (Appendix 1) was saved to the hard drive. The page was then modified to exclude any link to the Internet. Some JavaScript was then added to take a time reading at the very beginning of the page, and another reading at the very end of the page. The time readings and their difference were then displayed at the very end of the modified page (See Appendix 2 for the JavaScript source code).

To run the test, all other applications were turned off, and the computer disconnected from the Internet and local network. The test was run against the

three browsers one at a time. The first step was to load the page in the browser. Then a reading was taken each time the “reload” command was issued to the browser. The test was run 30 times in each browser to gather a large enough sample.

The result did not confirm Apple’s claim about Safari. IE was the slowest with an average render time of 1.212 second, although a user might think msnbc.com should have included special treatment for IE the way Microsoft does for other applications. Safari had an average render time of 0.910 seconds, far behind Camino’s 0.470 seconds (See Appendix 3 for detailed test result).

The test was a “quick and dirty” one and did not take into consideration if there is any optimization for hard drive access by the browser. It also assumes that all contents of the page including JavaScript are processed sequentially as the page is loaded.

The result should be a fair indication that the Gecko engine inside Camino did live up to its hype in rendering speed. However, this test alone does not prove any speed difference in the real use of the browser when other factors come into play.

One observation worth noting was that both IE and Safari tended to slow down in render time as the number of reloads increased (Appendix 3). This leads to some suspicion of memory handling problems in both of these browsers. Camino was free of this symptom.

□ **Stability**

For the three months that Safari was used, the author noticed that there were many times it quit responding for unidentified reasons. It then had to be closed and restarted to return to its normal state. The Camino browser is not much better in terms of stability. It often had “unexpected quit” in the middle of a “surfing session”. This behavior does not appear to be totally random either. For example, it seems to “unexpectedly quit” every time the News and Observer’s web site (<http://www.newsobserver.com/>) is visited, which suggests it might be a compatibility issue as well. However, the integrated talkback utility sends the crash information back to the Camino developers, and some bugs are fixed very quickly. For example, in version 0.6 of Camino (still name Chimera), it always crashed when the user dragged the scrollbar before a page finished loading. That problem was resolved in version 0.7. IE, on the other hand, being the most mature browser among the three, seemed to be the most stable one as well.

□ **Compatibility**

All three browsers claim they support the latest Internet standards, including HTML, XML, XHTML, DOM, Java, Javascript, CSS, and Quicktime, Flash and Shockwave plug-ins. For the purpose of writing this paper, there was no time to run the all the W3C standard test suites against the three browsers to find out how compatible they are with the current standards. The author’s personal hands-on experience seems to suggest that Camino is the best of the three in terms of rendering pages accurately. Safari, on the other hand, seemed to struggle in many

situations. It is particularly troublesome with table spacings. IE is somewhere in between. In most cases, web pages display nicely in IE, but occasionally, something strange shows up too. For example, the ESPN's 2003 NCAA Men's Basketball poll web page displayed nicely in Camino, but when the same web page is loaded in Safari, many of the table cells appeared to be mis-aligned and when it is displayed in IE, some of the advertisements appeared to be misplaced (Appendix 4, A-C). It is not surprising that Camino appears to be the best in this category, since it uses the open source Gecko browser engine, which has a reputation of being valiantly standards-compliant. Safari's rendering engine, KHTML, is reported to be less able to correctly display many real-world pages, and it is less featured than Gecko, for example, it does not yet support XSL, MathML, MNG, or SVG (Upsdell, 2003). Microsoft, the company that developed IE, takes an interesting position towards supporting standards:

Because a standard exists, that does not mean Microsoft will automatically implement it. Microsoft will implement appropriate standards that we believe are useful to our customers (www.microsoft.com).

The fact that Microsoft's various flavors of IE occupy more than 95% of browser market makes its standard-supporting position an influential one. This is especially true in cases where Microsoft decided to support standards that they developed themselves. When web developers start to take advantages of these non-standard things that the popular IE browser supports, designers of other browsers are put into a difficult position.

It is worth pointing out that although the capability of rendering real web pages correctly and the browser's level of standards-compliance are related to some

extent, there are differences between the two. The reason is because not all web pages are developed to follow standards. To maximize the correct rendering of real pages, many browsers, including Safari, IE Mac, and Camino, have two or more rendering modes, one renders pages strictly in accordance with the current standards while the other mode tries to render pages similar to legacy browsers. Which rendering mode is chosen is based on the DOCTYPE attribute of the web page being rendered (Upsdell, 2003). In some special cases, for example, strictly following the standard may result in incorrect display of a web page, for example, unwanted vertical gaps in tables containing images (Meyer, 2003).

In summary, being strictly compliant with standards and capable of correct display real web pages are two related, but different things. If the goal is to find out how standard compliant a browser is, then the W3C standard test suite should be run. But on the other hand, if real-world performance is the focus, then maybe a list of popular web sites should be tested to access a browser's capability of rendering pages correctly. The authors perception on Safari is that, while it may be a standard-compliant browser as Apple claims, its beta version does not do a very good job in terms of accurately rendering real world web pages.

2. Security

All three browsers have support for 128-bit security, proxy server, and cookie management. Although there are various online tools available for checking browsers' security features, none of the tools the author found would work with Safari and Camino browsers. Most of them are designed to check the security vulnerabilities of the more popular browsers, such as IE and Netscape. And since both Safari and Camino are still in their Beta versions, it is less meaningful to run a full security test against them. This is because even if a beta browser meets certain security standards, software updates can unknowingly change aspects of how a

program operates in regard to security. Therefore, instead of comparing the actual securities of the three browsers, the focus of the discussion here will be on how configurable the security features of these browsers are.

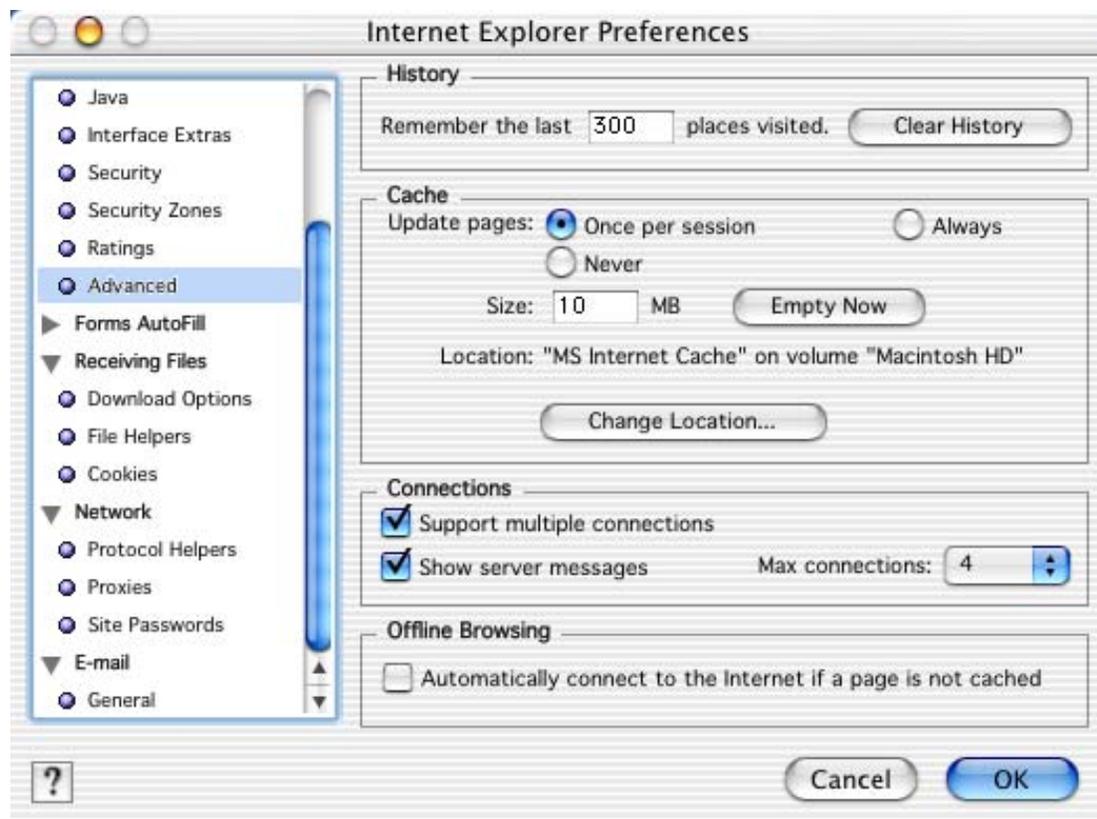


Figure 2. Partial screen shot of the IE Mac 5 security settings. Various security-related settings are located under Java, Security, Security Zones, Ratings, Advanced, Cookies, Proxies and Site Password sections of the Preference window.

IE is clearly the most configurable browser of the three with more than twenty security related user configurable settings (Figure 2). Things covered in IE security settings include active web content, JAVA, security alerts, certificate authority setup, security zone setup, web site content rating setup, cache management, cookie management, proxy server configuration, and site passwords setup.



Figure 3. Screen shot of all the Camino version 0.7 security settings available through the GUI interface.

Camino's user preferences dialog box shows about ten security-related settings covering JAVA, cookie management, site passwords, security alerts, and some active web contents (Figure 3).

A visit to the Camino project web site unveiled many more hidden user-configurable security settings. However, these settings are only available through command line, not the graphical user interface. Similar to the way IE's security zone works, these settings allow user to setup security policies for the browser, and also have different security policies for different Internet sites. And it is made clear in the documentation that these settings are aimed at only programmers familiar with

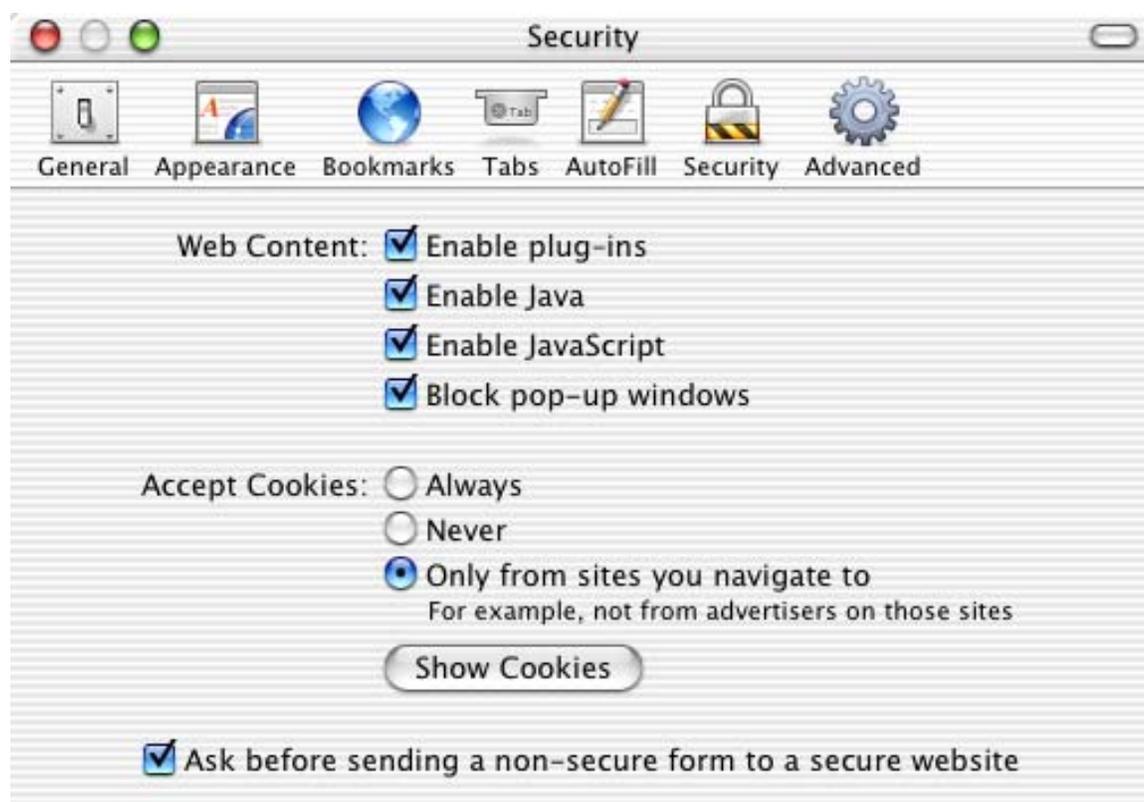


Figure 4. Screen shot of all the Safari version 1.0 Beta 2 (v73) security settings available through the GUI interface.

JavaScript, not the regular users. Finally, Safari, the browser with a very simple user interface, only has five user configurable security settings covering JAVA, some active web content, cookie management and security alerts (Figure 4).

It is important to point out that having less user configurable security settings does not necessarily mean the browser is less secure. And there should not be any hard rule about which way is better, having more configurable settings or having less. It depends on the user's experience and technical expertise, some users may feel more comfortable with more control while others may be more than happy to have the computer take care of things for them. Being simple and having fewer commands means less potential for user errors. In the case of Safari, this may be a design goal and the "power features" are intentionally left out. As Apple's CEO noted in his introduction of the new browser, "it has a very minimal user interface" (Wilcox, 2003)

3. User Interface / Usability

- Intuitiveness

At the first glance, all three browsers use the basic browser interface with a standard set of icons from a long time ago. Looking at them closely, you will find both Safari and Camino have incorporated some refreshing innovative ideas in their simple yet effective user interfaces. Some subtle differences among the three browsers are examined below.

- The “Stop” and “reload” buttons (Figure 5)

Both Camino and IE have two separate buttons for the action “Stop” and “Reload”. And in Camino, the two buttons are never enabled at the same time. Once a page finishes loading, the “stop” buttons is grayed out. And as soon as a page starts loading, the “reload” buttons is grayed out. To really do a reload while a page is loading, the user must press “stop” and then “reload”. IE does this a little differently by having the “Reload” button enabled at all times while disabling the “Stop” button once a page has finished loading. So reloading a page while it is still loading only takes one click, as oppose to two clicks, in IE.

Safari took the concept from Camino a step further. It is the only browser to overload the same button with the “reload” and “stop” function. Therefore the users do not see the unnecessary buttons, and screen real estate is maximized for displaying page contents. One could argue it is a break from the common norm for browsers to display the “reload” and “stop” buttons, and thus a break of the consistency principle of HCI. Consistency within an application is distinct from consistency across applications (perfect

consistency would stymie innovation). However, one could also argue that it is unnecessary to display the disabled buttons, such as the ones in Camino.

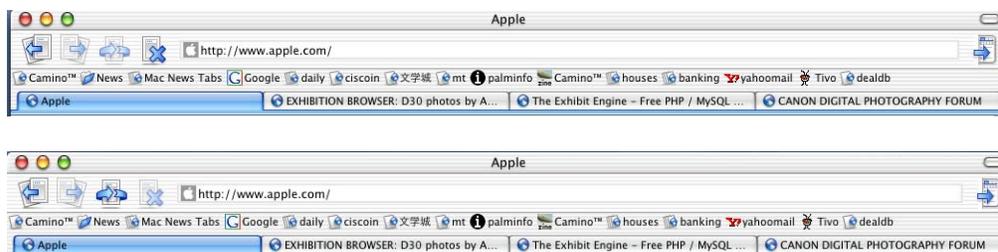


Figure 5-A. Screen shots of the Camino Stop and Reload buttons. The Reload button is disabled while a page is loading and the Stop button is disabled when the page finishes loading.

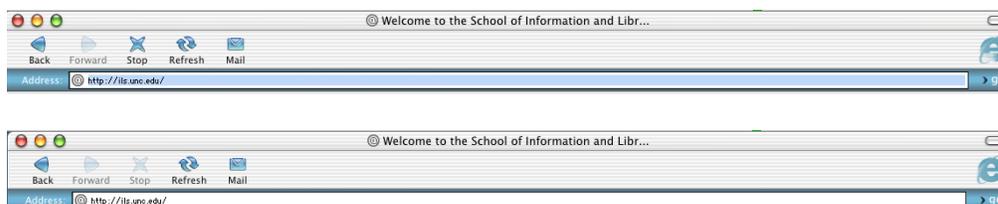


Figure 5-B. Screen shots of the IE Mac Stop and Reload buttons. The Reload button is enabled at all times and the Stop button is disabled when the page finishes loading.

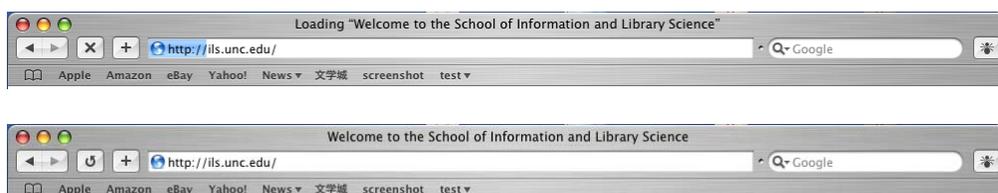


Figure 5-C. Screen shots of the Safari Stop and Reload buttons. While a page is loading, only the Stop button is displayed and once the page completes loading, only the refresh button is displayed.

- Bookmark bar management

All three browsers allow you to drag a link to the bookmark bar to create a new item. IE and Camino use the page title as the name of the item. A user usually prefers a shorter name to save the limited real estate on the bar. Two or three clicks are required to make the name change. Safari apparently paid attention to users. When the user drags a link to the bar, it pops up a small

window with the default name highlighted so the user can type in a shorter name.

Dragging a bookmark to the browser's display area also incurs different behavior. While IE and Camino display the page, Safari deletes the bookmark. Dragging a bookmark to the desktop has different results, too. IE does nothing; Camino creates a URL shortcut on the desktop; and Safari deletes the bookmark. Safari's behavior of deleting when an item is dragged away from the bar is worth noting. It is, in fact, consistent with the behavior of the Mac Dock. When an icon is dragged away from the Dock, it is deleted. As the only natively developed browser for Mac OS X, it is no surprised that Safari retained the Mac flavor.

When a hyperlink from a web page is dragged to the bookmark bar, all three browsers add the link's destination url to the list of bookmarks. However, IE and Camino provide good user feedback by putting a small cursor at the place of bookmark insertion. Safari has a nice graphical presentation of the link being dragged, but provides no feedback if the link is put at a blank space on the bookmark bar. When the link is inserted in between two existing bookmarks, Safari does an animated slide-over with the existing link(s) on the right hand side. However, the graphical presentation of the link being dragged effectively blocks the user's view of what is underneath.

Overall, Safari's default user interface appears to be very simple and straightforward. It would be an effortless transition for a MAC IE user, or even a long time Windows user, like in the author's case, to switch to Safari and start surfing the web. The purposes of most, if not all, Safari-unique browser elements, such as Google search, are obvious enough that it would not even confuse a first time user.

□ **Convenience**

Navigation has always been a very important topic of hypertext systems and it was said that modern WWW browsers are using fewer navigational paths than some of the original hypertext systems (Nielson, 1995). Apparently, the developers of the two more recent MAC browsers, Camino and Safari are trying to address this issue and they a very good job. Both browsers present some innovative navigational aids to the users. These convenience features are discussed below.

- **Tabbed Browsing**

Tabbed browsing is a very useful way to handle multiple browser windows.



Figure 6. Tabbed browsing in Camino. One click on the toolbar bookmark named “daily” opens seven web pages in different tabs inside one browser window.

This is especially helpful on the Macintosh platform because unlike the Window’s taskbar, the Dock does not show an icon for every open window of a running application. Users of IE and Safari often find themselves searching a particular window from a stack of windows, including pop-ups in the case of IE. This may be the single biggest reason that many Camino fans won’t switch to another browser, once got in to the habit of using the tabbed browsing feature, they simply cannot live without it while browsing the web. Moving from Mozilla trunk 1.0 to 1.3, Camino version 0.7’s tabbed bookmark group can now be set as homepage. Similar to the fat link concept mentioned earlier, this feature allows user to open all my daily must-visit web sites inside one browser window on multiple tabs with just one click (Figure 6).

This tabbed browsing feature is probably the most requested feature the Safari developers received from their public beta users. Under this tremendous pressure from the users, Safari is working on adding tabbed browsing capability as well. Its private beta releases were leaked to the public and Apple Computer had to terminate the private beta early. The official Public beta that includes this feature is yet to become available, but what was in the leaked private beta version appeared to be very rudimentary, compared to Camino', although this is not a fair comparison.

- Address bar search

The Windows versions of IE allows user to perform Internet searches directly from the address bar, something started way back when Netscape still held more market share. And IE allows the users to configure which search engine to be used. However, the configuration of the address bar search is buried deep into several levels of menus, and is rarely discovered by users. Even some advanced users settle for the default msn.com search engine. The same feature on the latest version of IE Mac is only worse. For some unspecified reasons, address bar search appears to be broken. An attempt to do a search will result in an error. Attempts of trying to configure the search engine appeared to be successful, but the default one was never changed.

Camino does not do address bar search by default. Some advanced users have got creative and invented ways to perform the equivalent of address bar

search. One such way is to create a bookmark to the intended search engine and assign a keyboard shortcut. The search is performed by combining the keyboard shortcut and the content of the address bar. This is not intuitive for beginners, but does provide flexibility to advanced users because multiple search engines can be configured at the same time and identified by different keyboard shortcuts.

Safari is the most straightforward. It put a separate google.com search box on the address bar. It even has snap-back technology integrated. However, the default search engine is Google, and that is not user configurable.

- Snap back

Snap-back is a new technology inherited from KHTML by Safari. It is a Safari-unique feature that instantly returns you where you started. It works in both the address bar and the Safari Google search bar. In the address bar, it brings you back to the top level of a website after navigating through a series of links. In the Google search bar, it brings you back to the retrieved results of the last search. The snap-back works even if the search is performed from the Google website. The snap-back icon is still presented in the Safari Google search bar.

- Show related links

IE for Mac has a unique feature called "Show Related Links". Once clicked, it will display a list of web sites similar to the one currently displayed in the

browser in a side bar. It also shows additional information, such as contacts, mailing address, telephone, and fax number of the current site. This could be a very useful navigational aid when a user is trying to compare several similar online shopping sites.

One of Shneiderman's golden rules is about supporting *user locus of control*. The authors' hands-on experiences with the three browsers indicate that IE is far superior to the other two browsers in this category. This could be because IE is a much more mature product. IE has many user configurable features while the other two browsers only have a few. It appears that both Safari and Camino browser developers spent considerable effort to make their interface as user friendly as possible, and they seemed to do a very good job. However, sometimes users may like to have a little bit more control over how the tool should work for them instead of having the tool determine what is best for them. Some of the author's favorite IE customizable features that are missing in Camino and Safari are:

- One click select all in URL address field: This is very convenient when you need to type a URL in the address field. To do a select all in the URL field in Safari and Camino, one has to click once to focus in at the address field then double click to select the whole thing. Although all three browsers enable the keyboard shortcut "ctrl-L" to accomplish this task, it is probably more intuitive, for most non-technical users, to click with a mouse than use the keyboard shortcut.
- Ability to set the length of the browser history and cache size: Camino also has this feature but is slightly different and with less options. For example,

IE defines the browser history as the number of web sites, while Camino uses time to determine the length of the browser history. And for cache, Camino only indicates to users that cache is being used and gives an option to clear it. IE has a much more advanced customization for cache, it lets the user decide when to load pages from cache, the size of the cache, the location of the cache folder as well as an option to manually empty the cache. Safari has nothing configurable in terms of browser history and cache management.

- Safari automatically made itself the default web browser without asking the user's permission. Although it can be changed later in the browser's Reference dialog box, it may still make users feeling that this software has too much "power" than it should have.

Safari's lack of customization might have originated from Apple's goal of making a computer that is simple to use. Most Mac users claim that they "just use the computer, not play with it like the PC users."

□ **Accessibility**

A full UAAG 1.0 evaluation of IE 5.0 for Mac can be found at W3C's web site (Celik, 2002). The web site also published a comparison chart that showed UAAG 1.0 implementations in fifteen different browsers. Ideally, a report similar to the UAAG 1.0 evaluation of IE should be done for each browser being assessed. However, for the purpose of writing this paper, there was no time to validate all eighty-three checkpoints outlined in the W3C's UAAG to test the accessibility of the three browsers. Instead a few priority one checkpoints were selected and then their implementations were tested in the three browsers. The test results are

shown in Table 1 below, a checkmark means the feature is implemented while blank indicates its absence:

	Safari	Camino	IE5 MAC
ALT attribute for IMG element (checkpoint 10.1)			Only enabled when image is turned off
TITLE attribute for IMG element (checkpoint 10.1)		✓	✓
LONGDESC attribute for IMG element (checkpoint 10.1)			
Blinking text controlled by proprietary Markup (checkpoint 3.3)	Disabled or not implemented. Cannot enable via interface.	Enabled. Cannot disable via interface	Disabled or not implemented. Cannot enable via interface.
Blinking text controlled by CSS (checkpoint 3.3)	Disabled or not implemented. Cannot enable via interface	Enabled Cannot disable via interface	Disabled or not implemented. Cannot enable via interface
Scrolling text controlled by the proprietary element MARQUEE (checkpoint 3.3)	Disabled or not implemented.	Enabled. Cannot disable via	Enabled by default. Cannot

	Cannot enable via interface	interface	disable via interface
Scrolling text controlled by JAVASCRIPT (checkpoint 3.3)	Disabled or not implemented. Cannot enable via interface	Disabled or not implemented. Cannot enable via interface	Configurable via interface
Apply User style sheets (checkpoint 4.14)			✓
Scroll through the table cells while table header and footer information remains in view (checkpoint 10.1)		✓	
Object element for audio – display text when audio is disabled (checkpoint 2.3)			
Toggle scripts on and off (checkpoint 3.4)	✓	✓	✓
NONSCRIPT conditional content	✓	✓	✓

After running a few tests and reading the IE for Mac evaluation report, the author's perception is that all three browsers are still far from completely implementing everything specified in the UAAG 1.0. IE is probably the most

advanced of the three in this category with the most configurable user options. Safari, on the other hand, appears to be less developed and has the fewest user options. After using the three browsers for about three months, the author feels that there are a few user interface related observations worth noting. These observations are not specific to any single UAAG guideline, but are associated with the accessibility issue in general.

- **Spell check.** As Apple's "native" browser, Safari takes advantage of having the privilege of accessing the OS system resources and provides spell-checking functionality. The spell checking works as the user inputs text in any editable form field. This feature is very useful for users have problem with spelling in English. For non-Apple "native" browsers like IE or Camino, adding this feature, as pointed out by a Camino user on a forum, means the browsers would occupy more memory space and run slower because more code and a database are needed to handle spell checking inside browsers.
- **Foreign language support.** For some of the Chinese web sites the author regularly visits, IE seems to have problems render the content texts correctly. Even worse, it does not seem to support non-English languages in page titles, the bookmark bar and menus. Camino and Safari have done a much superior job in this regard (Figure 7, A - C). Camino always displayed the pages beautifully, while Safari only had some problems when two text-encoding schemes, specifically, Chinese simplified and Chinese traditional, are used together on a single web page.



Figure 7-A. The Chinese page title of this web site renders correctly in Safari's title bar and bookmark bar.



Figure 7-B. The Chinese page title of this web site renders correctly in Camino's title bar and bookmark bar.



Figure 7-C. The Chinese page title of this web site does not render correctly in IE Mac's title bar and bookmark bar.

- Read-a-Page. All browsers on the MAC OS X platform have access to a set of services provided by the operating system. One of the services is speech. Both Safari and Camino integrated this feature and allow user to highlight texts on a web page then let the browser read the text put. This feature is not available in IE. Although this sounds like a great concept, it is not clear whom the target audience would be. If blind people are supposed to benefit from this, it seems that the feature would be impossible for them to use since first they need to

highlight the text and then they need to go three levels down the menu to find the command.

- **Toolbar icons and real estate usage.** IE is the only browser of the three that allow users to choose a toolbar style: text only, icon only or both. Safari and Camino only have graphic icons for their tool bars. All three let user customize their toolbars, but IE and Camino allow users to drag-n-drop any icons anywhere on the tool bar while Safari only allows toggling the display of individual toolbar icons. Some Safari users have complained about the inability to move toolbar icons and the built-in Google search box on a forum. By default, Safari and Camino display the toolbar icon and the address bar in the same line, while the IE address bar has its own line, which takes more screen space. All three browsers allow users to hide the toolbar, address bar and bookmark bar to maximize viewable screen space. IE and Safari goes one-step further, even allowing users to hide the status bar at the bottom. Safari is the most innovative in terms of maximizing screen space. It shows the page loading status as a progressing blue bar inside the address bar as a page loads in the browser. This way, users can turn off the bottom status bar and still be able to monitor how the page loads.
- **Form auto-fill.** Safari is the only browser of the three that does not have the form auto-fill features³. There has been much complaining on several Mac

³ This statement is no longer true with Safari version 1.0 Beta 2 (v73), which has a user configurable form auto-fill feature. The feature works with the standard My Address Book program on Mac OS X platform, which means users don't have to re-entered the information in browser's preference settings.

forums about Safari's lack of this feature. It may be a convenience feature for most users, but for people with memory problems, this may be a critical factor as to how well they complete tasks, such as submitting an online form. IE has done an excellent job on this feature by making it user configurable. Users can enable or disable this feature, as well as adding any words or phrases for the browser to remember. Additionally, users can set an AutoFill Profile, which will be used to fill out forms on web pages when the AutoFill toolbar button, also IE unique, is clicked.

- **Printing.** All three browsers offer a print preview feature. While both Camino and Safari send the page to Acrobat Reader for preview, IE provides a more flexible approach. It not only lets users see what the print-out would look like from inside the browser, but also gives users the ability to "modify" the look of the printed web page by specifying text size, with or without header, background, images, etc... (Figure 8). In addition, IE's print preview is on the first level of the "File" menu, as opposed to being embedded in the print dialog box for the other two browsers, and is thus more intuitive to access, especially for Windows users.

Welcome to the School of Information and Library Science

Close Page Setup Print

Welcome to the School of Information and Library Science 4/22/05 5:05 AM

[Text-only site](#)

Ranked #1 by U.S. News & World Report

UNC SCHOOL OF INFORMATION AND LIBRARY SCIENCE

The University of North Carolina at Chapel Hill

About the School
[Dean's Welcome](#) - [Admissions](#) - [Catalog](#) - [Fast Facts](#) - [Mission](#) - [History](#) - [Accreditation](#)
[People](#) - [Area Info](#) - [Development](#) - [Public Service](#) - [Laptop Requirements](#) - [Contact Us](#)

Academic Information
[Schedules](#) - [Courses](#) - [Degrees](#) - [The Master's Paper](#) - [Undergraduate Minor](#) - [Major](#)
[Continuing Education](#) - [Field Experience](#) - [School Library Media Coordinators](#) - [Intl Programs](#)

Research
[Faculty Research](#) - [Technical Reports](#) - [Doctoral Research](#) - [Dissertations](#)
[Master's Papers](#) - [Carnegie Grants](#)

Labs & Learning
[lib101.org](#) - [Interaction Design Lab \(IDL\)](#) - [Internet 2/OSI](#) - [Plant Information Center](#)
[Open Video Project](#) - [CoRE](#) - [Bioinformatics](#) - [NC Health Info](#) - [Metacata Generation Project](#)
[Digitization for Cultural Heritage Professionals Workshop](#)

Facilities
[SILS Library](#) - [Computer Lab](#) - [Rooms/Resourses](#)

Employment **News & Events**
[Career Services](#) - [Jobs at SILS](#) [Headlines](#) - [Calendar](#)
[Post to SILS Jobs Library](#) [SILS Newsletter](#) - [Commencement](#)

ALUMNI RECEPTION
Join us for an
Alumni Reception
and
a Celebration
of the
25th Anniversary
of the
Doctoral Program
562 Dirksen Senate
Office Building,
Washington, D.C.
Friday, April 25
5:00 - 7:00 pm
[CLICK FOR MORE](#)

[Search](#) [Site Map](#) [UNC Home](#) [UNC Library](#) [Contact Us](#)

http://ils.unc.edu Page 1 of 1

+ —
Larger Smaller

Headers and Footers
 Print Background
 Print Images

+ —
 Shrink Pages to Fit
 Crop Wide Pages
 Print Wide Pages

Figure 8. Print Preview in IE Mac. Note the user configurable printing options at the bottom.

4. Auxiliary Functionalities

Modern browsers are designed more and more to reflect things that web users really want. Both Safari and Camino browsers offer easy-to-use pop-up blocking features that will free users' screens from unwanted, intrusive advertising pop-up windows. The feature appears to be effective in both browsers, the only difference is that in Safari there is a command directly included in the menu to allow users to easily turn the feature on and off on the fly, while in Camino, users have to go into preference to change this setting. The recent nightly build of Camino has added a small icon in the status bar that shows up every time a pop-up is blocked. The user can click on the icon to let a pop-up show up instead.

For people who use online auctions such as EBay, IE has an integrated Auction Manager that can monitor online auctions. One can easily check the status of an auction and visit other web pages without having to return to the original auction site. One can also have the Auction Manager notify him/her when there are high-bid changes – such as when he/she is no longer the highest bidder and when an auction closes. The author does have first hand experience with the feature, but has heard good things about it on one of the MAC forums.

In times of rapid event changes, one may wish to stay on top of the latest news. The subscription feature of IE can help. When a user subscribes to a web page, IE checks and notifies to the user about changes to that web page according to the schedule the user specifies. If preferred, IE can also download the content so the user can read it offline later. Although this seems to be a nice convenient feature for

the web users, some web masters are trying hard to restrict people from subscribing to their site to maximize bandwidth.

Finally, it is worth noting that there is a very simple yet useful feature in Camino called “send link”. It allows users to share any web link with others easily. One click opens a new message in the users’ default mail application, with the URL of the current page already pasted in. IE also has this feature, but it is less intuitive. It only works when the mail button is enabled on the tool bar. To send the link, a user has to point to the mail icon, hold down the mouse button, and then click the Send Link on the pop-up menu.

CONCLUSION

Safari entered a market with two leaders: Internet Explorer, the industry standard from the software giant; and Camino (then Chimera), the valiantly standards-compliant upstart from the massive and well-publicized Mozilla project. Using the framework, the author was able to compare Safari with these two competitors for the Mac OS X platform from many different perspectives. To sum up the detailed findings described above, a simple matrix is developed to outline all major points of the study. This matrix provides a high-level overview of each browser's strength and weakness. The rating is on a 3-point scale, with "1" being the best of the three compared, while "3" being the worst.

		Safari	IE	Camino
Performance	Speed	2	3	1
	Stability	2	1	3
	Compatibility	3	2	1
Security		3	1	2
UI / Usability	Intuitiveness	1	2	3
	Convenience	1	3	2
	Accessibility	3	1	2
Auxiliary Functionalities		2	1	2

Overall, the author's review suggests that although it still is a little pre-mature, Safari is a good browser with a lot of potential for forcing the rest of the field to reshape in its image on the usability criterion. It will attract many users who desire simplicity

and innovative visual appearance. But for power users, who desires ultimate control, IE and Camino will remain strong players in the field.

SUMMARY

Deciding which browser is the best is very much a personal choice. There will never be one best browser for everyone. Everybody has his or her special needs and interests. Some people may be more interested in the latest bells and whistles while others may care more about speed and stability. Using the framework described in this paper allows reviewers to evaluate a browser from all angles and therefore provides users a more complete picture. Further more, using a standard framework to assess all browsers will make the evaluations more comparable, and hence more useful in terms of assisting users to pick the best browser.

BIBLIOGRAPHY

- Bush, V., "As We May Think." *The Atlantic Monthly*, 1945. 176(July): pp. 101-108.
Reprinted and discussed in *interactions*, 3(2), Mar 1996, pp. 35-67.
- Cailliau, R. "A Short History of the Web" Text of a speech delivered at the launching of the European branch of the W3 Consortium Paris, 1995
- Celik, T. "UAAG 1.0 Evaluation of IE 5.0 for Mac", *Web Accessibility Initiative document archive*, August, 2002.
- Engelbart, D. and English, W. "A Research Center for Augmenting Human Intellect."
Reprinted in ACM SIGGRAPH Video Review, 1994., 1968. 106
- Festa, P. "Netscape blares browser reports", *CNET news archives*, August 8, 1997
- Festa, P. "Study: Netscape share below 50%", *CNET news archives*, September 29, 1998
- Franklin, S. "Browsers, Browsers, Browsers! A Strategic Guide to Browser Interoperability", *Dr. Dobb's Software Tools for the Professional Programmer* (<http://www.ddj.com/>), Web Reviews, December 29, 2000
- Goldberg, A., ed. *A History of Personal Workstations*. 1988, Addison-Wesley Publishing Company: New York, NY. 537.
- Heeter, C. "The Look and Feel of Direct Manipulation", *HYPERNEXUS: Journal of Hypermedia and Multimedia Studies*, Fall, 1991.
- Jacobs, I., J. Gunderson and E. Hansen, eds., "User Agent Accessibility Guidelines 1.0" (<http://www.w3.org/TR/2002/REC-UAAG10-20021217/>), *W3C Recommendation*, 2002. Latest version of this document is available at <http://www.w3.org/TR/UAAG10/> .
- Koved, L. and Shneiderman, B., "Embedded menus: Selecting items in context." *Communications of the ACM*, 1986. 4(29): pp. 312-318

Microsoft Corporation. "Using the Discussion Feature in Internet Explorer", *Microsoft Knowledge Base Article – 255085*, September, 2002.

Melton, D. (on behalf of the Safari Team), *Email correspondence to the KDE Development list*. January, 2003.

Meyer, E. A., "Images, Tables, and Mysterious Gaps", *Netscape Devedge* (<http://devedge.netscape.com/viewsource/2002/img-table/>), March, 2003.

Myers, B. A. "A Brief History of Human Computer Interaction Technology." *ACM interactions*. Vol. 5, no. 2, March, 1998. pp. 44-54.

Nelson, T. "A File Structure for the Complex, the Changing, and the Indeterminate," in *Proceedings ACM National Conference*. 1965. pp. 84-100.

Nielsen, J., *Multimedia and Hypertext: the Internet and Beyond*. 1995, Boston: Academic Press Professional.

Nielson, J. "Features for the Next Generation of Web Browsers", *Alert Box*, July, 1995

Robertson, G., Newell, A., and Ramakrishna, K., *ZOG: A Man-Machine Communication Philosophy*. Carnegie Mellon University Technical Report Report, Number, August, 1977.

Shneiderman, B. *Designing the User: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, MA., 1987

Stein, L., J Stewart and Pearlman, L., "The World Wide Web Security FAQ – part 9. Client Side Security", *W3C Documents*, February, 2003.

Upsdell, C., "Browser News (<http://www.upsdell.com/BrowserNews/>)", 238th Edition, April, 2003

van Dam, A., *et al.* "A Hypertext Editing System for the 360," in *Proceedings Conference in Computer Graphics*. 1969. University of Illinois.

W3C Quality Assurance, "The Matrix of W3C specifications", January 15, 2003

Wilcox, J., "Apple releases its own Web browser", *CNET news archives*, January 7, 2003

Appendix 1. Screenshot of the Rendering Speed Test

MSNBC Cover

file:///Users/xtai/Desktop/speed%20test/msnbc.html

Apple Amazon eBay Yahoo! News ▾ screenshot test ▾ 雅虎中国

Business

• Change my business choices • Remove business from my page

Stock data from CNBC on MSN Money

Click symbol for chart, news, and profile

SYMBOL	LAST	CHANGE	PREVIOUS	LOW	HIGH	VOLUME
• CSCO	13.71	+0.07	13.64	13.51	13.89	51,403,073
• EMC	7.80	-0.05	7.85	7.56	7.89	13,311,800
• MSFT	25.09	-0.64	25.73	24.93	25.80	55,755,448

Quotes supplied by S&P Comstock and delayed at least 20 minutes • Investment toolkit
NYSE, AMEX and NASDAQ index data are provided real time

Latest MSNBC and wire stories for your stocks

- CSCO: Network Assoc targets new market, buys Entercept
- MSFT: AOL seeks approval to offer advanced IM service
- View all news stories for your stocks

Sports

• Change my sports choices • Remove sports from my page

NBA for Apr. 4, 2003

Times are in U.S. Eastern Time
Click team name for team page

YESTERDAY / TODAY / TOMORROW

End 3	Final
L.A. Lakers 66	Houston 88
Memphis 89	Philadelphia 79

GAME STORY | BOX SCORE

STEP UP TO THE BIG LEAGUES OF FANTASY! [Sign up Now!](#)

• BACK TO TOP

MSNBC is optimized for
• Microsoft Internet Explorer
• Windows Media Player

• **MSNBC Terms and Conditions**
© 2003

Cover | News | Business | Sports | Local News | Health | Technology & Science | Living | Travel
TV News | Opinions | Weather | Comics

InfoCenter | Newsletters | Search | Help | News Tools | Jobs | Write Us | Terms and Conditions | Privacy

©2003 Microsoft Corporation. All rights reserved. Terms of Use Advertise Truste Approved Privacy Statement GetNetWise

Render started at 38:35:66
Render finished at 38:36:222
Render took 0:1:156



Appendix 2. Javascript Source Code of the Rendering Speed Test

```
<html xmlns:ie="">
<head>
<!-- 10.48.148.25 -->
<title>MSNBC Cover</title>

<script language="JavaScript">
<!--
    ts = new Date();
//-->
</script>

[... page body goes here ... ]

<script language="JavaScript">
<!--
    tf = new Date();
    document.write ("Render started at " + ts.getMinutes() + ":"
+ ts.getSeconds() + ":" + ts.getMilliseconds() + "<br>");
    document.write ("Render finished at " + tf.getMinutes() + ":"
+ tf.getSeconds() + ":" + tf.getMilliseconds() + "<br>");
    document.write ("Render took " + Math.floor((tf.getTime() -
ts.getTime())/(1000*60)) + ":"
+ Math.floor((tf.getTime() -
ts.getTime())/1000) + ":"
+ (tf.getTime() - ts.getTime())%1000);
//-->

</script>
</body>
</html>
```

Appendix 3. Complete results of a rendering speed test for selected MAC OS X browsers⁴.

Tests	Rendering time (Millisecond)		
	IE Mac ⁵	Safari ⁶	Camino ⁷
1	1153	0809	0547
2	1108	0845	0454
3	1141	0835	0391
4	1080	0862	0435
5	1135	0835	0461
6	1121	0835	0489
7	1192	0844	0501
8	1207	0863	0484
9	1138	0875	0336
10	1164	0888	0482
11	1194	0894	0456
12	1223	0896	0322
13	1152	0909	0561
14	1172	0657	0516
15	1219	0920	0454
16	1231	0917	0455
17	1280	0936	0520
18	1227	0931	0409
19	1209	0961	0542
20	1271	0930	0522
21	1245	0964	0324
22	1266	0955	0479
23	1248	0944	0468
24	1264	0966	0536
25	1310	0958	0477
26	1325	1000	0536
27	1314	1015	0476
28	1332	1007	0484
29	1319	1014	0512
30	1130	1020	0456
Min	1080	657	322
Max	1332	1020	561
<i>Average</i>	<i>1212.33</i>	<i>909.50</i>	<i>469.50</i>

⁴ Hardware configuration: PowerBook G4 with 800 MHz PowerPC CPU, 768 MB RAM, 40 GB IDE Hard drive, and Mac OS X 10.2.3.

⁵ IE Mac version 5.2.2

⁶ Safari Beta 1.0 v62

⁷ Camino version 0.7

Appendix 4 A-C. Displays of the ESPN.com 2003 Men's NCAA Tournament bracket in Safari, Camino and IE Mac

ESPN.com: 2003 NCAA Tournament Bracket

2003 Men's NCAA Tournament Bracket

COMPLETE PRINTABLE BRACKET | DOWNLOAD PDF | WOMEN'S BRACKET

Member Name: Password: Sign In

Free Registration for Men's Tournament Challenge

1st ROUND	2nd ROUND	SWEET 16	ELITE EIGHT	FINAL FOUR	CHAMPIONSHIP
March 20-21 Gaylord Center, Nashville, TN	March 22-23 Metrodome, Minneapolis, MN	March 27-28 Metrodome, Minneapolis, MN	March 29-30 Metrodome, Minneapolis, MN	April 5 Superdome, New Orleans, LA	April 7 Superdome, New Orleans, LA
(1) Kentucky 95 (16) UPUJI 64 (8) Oregon 68 (9) Utah 60	(1) Kentucky 74 (9) Utah 54	(1) Kentucky 63 (5) Wisconsin 57	(1) Kentucky 63 (5) Wisconsin 57		
Spokane Arena, Spokane, WA					
(5) Wisconsin 81 (12) Weber St 74 (4) Dayton 71 (13) Tulsa 64	(5) Wisconsin 61 (13) Tulsa 50				
RCA Dome, Indianapolis, IN					
(6) Missouri 72 (11) S Illinois 71 (3) Marquette 72 (14) Holy Cross 68	(6) Missouri 92 (3) Marquette 101	(3) Marquette 64 (2) Pittsburgh 56			
Fleet Center, Boston, MA					
(7) Indiana 67 (10) Alabama 62 (2) Pittsburgh 62 (15) Vermont 51	(7) Indiana 52 (2) Pittsburgh 74				
Huntsman Center, Salt Lake City, UT					
(1) Arizona 80 (16) Vermont 51 (8) Connc 49 (9) Gonzaga 74	(1) Arizona 96 (9) Gonzaga 35	(1) Arizona 88 (5) N Dams 71			
RCA Dome, Indianapolis, IN					
(5) N Dams 70 (12) Wis-Mil 68 (4) Illinois 65 (13) W Kentucky 60	(5) N Dams 68 (4) Illinois 60				

Completed opening the page with 1 error (see Activity window for details)

Appendix 4-A. Screenshot of the bracket displayed in Safari. Note the table cells are misaligned and the MSN page content, which displays perfectly in the right side of the screen in IE Mac, is completely missing.

ESPN.com: 2003 NCAA Tournament Bracket

Address: http://ak-sports.espn.go.com/191778658760s/sports.espn.go.com/nba/ncaatourney03/tbracket

Apple Support | Apple Store | Mac OS X | Mac | Microsoft MacTopia | Office for Macintosh | vegas | CEJ-B2x

Back Forward Stop Refresh Home AutoFill Print Mail

BRACKET PRESENTED BY RUSSELL ATHLETIC

BRACKET PRESENTED BY RUSSELL ATHLETIC

2003 Men's NCAA Tournament Bracket

COMPLETE PRINTABLE BRACKET | DOWNLOAD PDF | WOMEN'S BRACKET

Member Name: Password: Sign In

Free Registration for Men's Tournament Challenge

STATISTICS MESSAGE BOARDS TRIVIA MEN'S COLLEGE BBALL on ESPN ESPN.com

PLAYOFF RACE: Where does your favorite team stand?

NHL.com on MSN

MARTIN ST. LOUIS: A Bolt out of the blue

Game-worn JERSEYS

HOCKEY CARD NEWS

CHAT with hockey fans

SENIOR LEAGUE HOCKEY

IMPACT! MAGAZINE

Watch hockey HIGHLIGHTS

Listen to live game RADIO

Vote for GOAL OF THE WEEK

NHL.com WIRELESS

powered by hp

1st ROUND	2nd ROUND	SWEET 16	ELITE EIGHT	FINAL FOUR	CHAMPIONSHIP
March 20-21	March 22-23	March 27-28	March 29-30	April 5	April 7
Gaylord Center, Nashville, TN		Metrodome, Minneapolis, MN		Superdome, New Orleans, LA	
(1) Kentucky 95 (16) TUPUJ 84 (8) Oregon 58 (9) Utah 80	(1) Kentucky 74 (9) Utah 54	(1) Kentucky 63 (5) Wisconsin 57	(1) Kentucky Sat. Mar 29 - TBA		
Spokane Arena, Spokane, WA	(5) Wisconsin 61 (13) Tulsa 60	(3) Marquette 68 (2) Pittsburgh 53			
(6) Missouri 72 (11) S. Illinois 71 (3) Marquette 72 (14) Holy Cross 68	(6) Missouri 92 (3) Marquette 10.1	(1) Arizona 96 (9) Gonzaga 95			
Fleet Center, Boston, MA	(7) Indiana 67 (6) Arizona 62 (2) Pittsburgh 57 (15) Wagner 61	(1) Arizona 96 (9) Gonzaga 74			
RCA Dome, Indianapolis, IN					
Huntsman Center, Salt Lake City, UT					
RCA Dome, Indianapolis, IN					

Appendix 4-B. Screenshot of the bracket displayed in IE Mac. Note the HP advertisement is misplaced and the two radio buttons seemed to show a white image background instead of being transparent. The fact that IE is the only browser that displays the MSN page content leads people to wonder if the MSN web developers are following some Microsoft's own standards.

ESPN.com: 2003 NCAA Tournament Bracket

http://ak-sports.espn.go.com/f/1917/8668/60s/sports.espn.go.com/nbc/ncaatourney03/bracket

Camino™ News Mac News Tabs Google daily 文学城 mt palminfo ciscoin Camino™ houses banking yahooemail

MSNBC - Your News Loading... Rear Projection Units - AVS Forum 手机印论坛

ESPN TO REGIONS: MIDWEST WEST SOUTH EAST

2003 Men's NCAA Tournament Bracket
[COMPLETE PRINTABLE BRACKET](#) | [DOWNLOAD.PDF](#) | [WOMEN'S BRACKET](#)

TEAMS BRACKET GALLERY SCOREBOARD TOURNAMENT CHALLENGE

STATISTICS MESSAGE BOARDS TRIVIA MEN'S COLLEGE BBALL on ESPN ESPN.com

Member Name: Password: Sign In

Free Registration for Men's Tournament Challenge

powered by

1st ROUND	2nd ROUND	SWEET 16	ELITE EIGHT	FINAL FOUR	CHAMPIONSHIP
March 20-21	March 22-23	March 27-28	March 29-30	April 5	April 7
Gaylord Center, Nashville, TN (1) Kentucky 95 (16) U.P.U. 64 (8) Oregon 56 (9) Utah 60 Spokane Arena, Spokane, WA (5) Wisconsin 81 (12) Weber St 74 (4) Dayton 71 (13) Tulsa 64 FCA Dome, Indianapolis, IN (6) Missouri 72 (11) S. Illinois 71 (3) Marquette 72 (14) Holy Cross 66 Fleet Center, Boston, MA (7) Indiana 67 (10) Alabama 62 (2) Pittsburgh 87 (15) Wagner 61 Huntsman Center, Salt Lake City, UT (1) Arizona 80 (18) Vermont 61	Metrodome, Minneapolis, MN (1) Kentucky 74 (5) Wisconsin 54 (3) Marquette 63 (2) Pittsburgh 57 2nd Half (1) Kentucky 52 (2) Pittsburgh 54 Sat. Mar 29 - TBA	Superdome, New Orleans, LA			

javascript:voic(0)



Appendix 4-C. Screenshot of the bracket displayed in Camino. Other than the missing MSN page content, the page is perfectly displayed.