US 20040148568A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0148568 A1

Springer (43) Pub. Date: Jul. 29, 2004

(54) CHECKER AND FIXER ALGORITHMS FOR ACCESSIBILITY STANDARDS

(76) Inventor: **Timothy Stephen Springer**, San Francisco, CA (US)

Correspondence Address:
FISH & RICHARDSON PC
225 FRANKLIN ST
BOSTON, MA 02110 (US)

(21) Appl. No.: **10/013,886**

(22) Filed: **Oct. 19, 2001**

Related U.S. Application Data

(60) Provisional application No. 60/297,994, filed on Jun. 13, 2001.

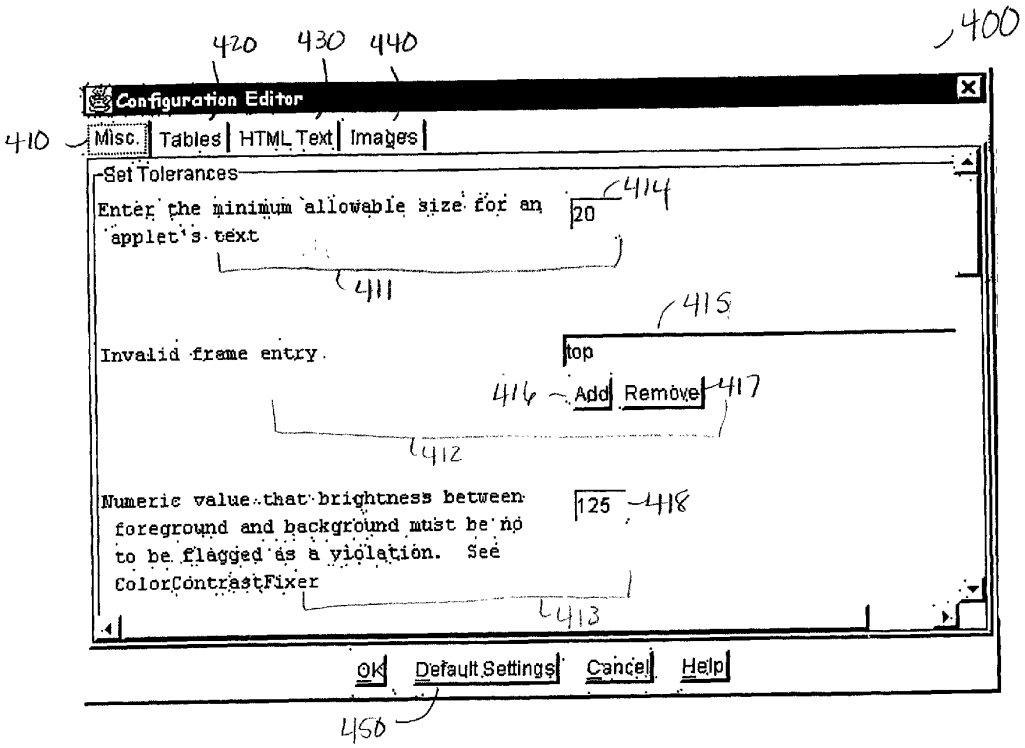
Publication Classification

(51) **Int. Cl.**[7] **G06F 17/00**

(52) **U.S. Cl.** **715/513**

(57) **ABSTRACT**

A method, apparatus and computer program product residing on a computer readable medium are described. The method, apparatus and computer program may use checkers to check a data model of a web page for accessibility, for example compliance with web accessibility requirements codified in 36 CFR § 1194.22. The method, apparatus and computer program may also implement fixers to modify HTML code to ensure compliance with 36 CFR § 1194.22. The method, apparatus and computer program may implement tolerances to allow personalization of checkers.



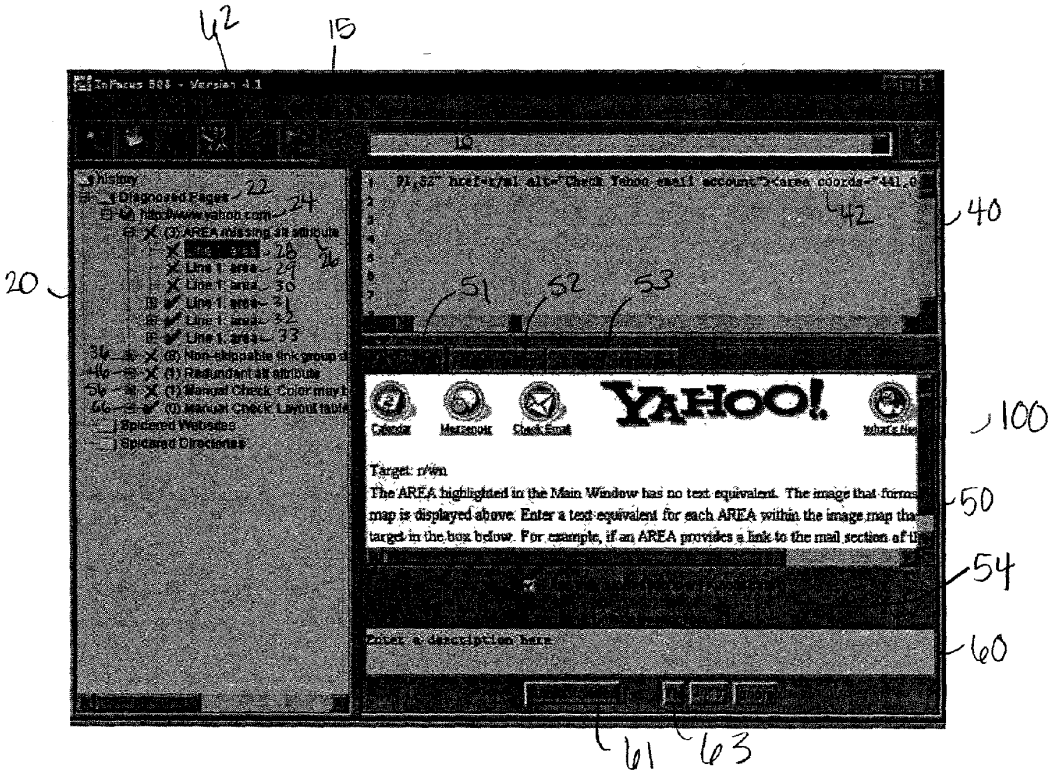


FIGURE 1

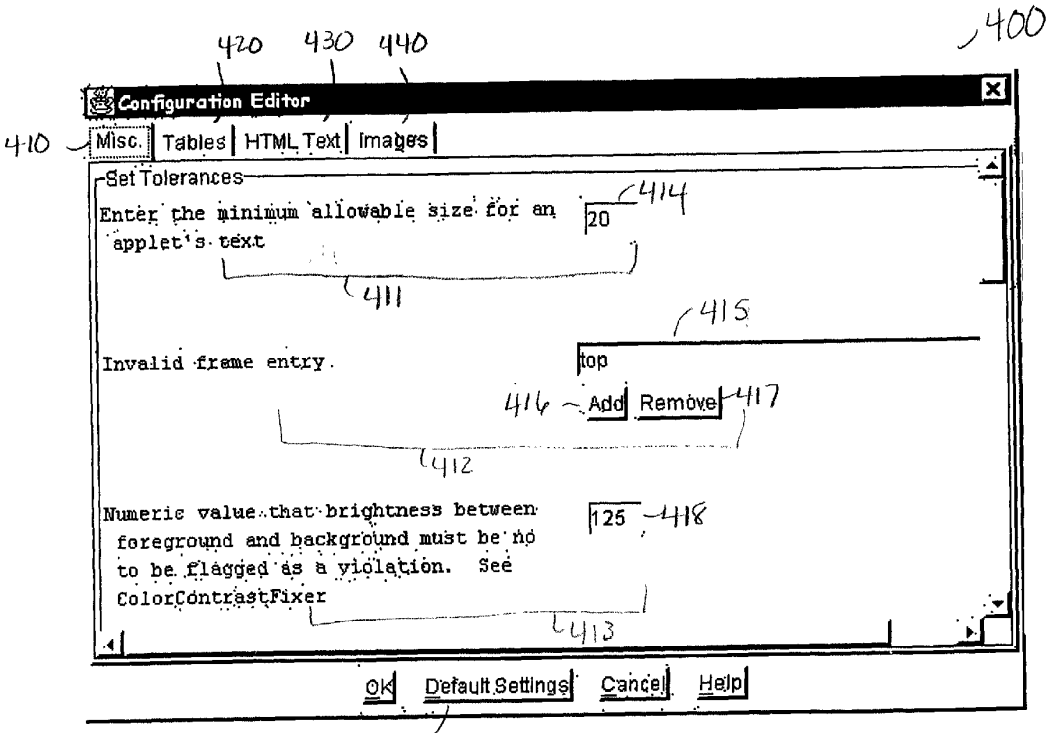


FIGURE 2.

CHECKER AND FIXER ALGORITHMS FOR ACCESSIBILITY STANDARDS

CLAIM OF PRIORITY

[0001] This application claims priority under 35 USC § 119(e) to U.S. patent application Ser. No. 60/297,994, filed on Jun. 13, 2001, the entire contents of which are hereby incorporated by reference.

COMPUTER PROGRAM LISTING APPENDIX

[0002] This application includes a computer program listing appendix in accordance with 37 CFR § 1.96(c), the entire contents of which are hereby incorporated by reference. One compact disk is submitted, with files: configuration.xml, 11 KB, Oct. 19, 2001; FormLabelFixerjava, 22KB, Oct. 19, 2001; LinkSkipCheckerjava, 14KB, Oct. 19, 2001; LinkSkipFixerjava, 14KB, Oct. 19, 2001; and NonVisualTextFilterjava, 2kb, Oct. 19, 2001.

BACKGROUND

[0003] This invention relates to verifying compliance with Section 508 accessibility standards and automatically retrofitting the HTML of web pages to ensure compliance.

[0004] The Architectural and Transportation Barriers Compliance Board (Access Board) has issued accessibility standards for electronic and information technology covered by Section 508 of the Rehabilitation Act of 1973, as amended by the Workforce Investment Act of 1998. The standards set forth a definition of electronic and information technology and the technical and functional performance criteria necessary for technology to comply with Section 508. As explained at http://www.access-board.gov/sec508/508standards.htm, "Section 508 requires that when Federal agencies develop, procure, maintain, or use electronic and information technology, they shall ensure that the electronic and information technology allows Federal employees with disabilities to have access to and use of information and data that is comparable to the access to and use of information and data by Federal employees who are not individuals with disabilities, unless an undue burden would be imposed on the agency. Section 508 also requires that individuals with disabilities, who are members of the public seeking information or services from a Federal agency, have access to and use of information and data that is comparable to that provided to the public who are not individuals with disabilities, unless an undue burden would be imposed on the agency."

[0005] The Section 508 standards cover various products including "web-based intranet and internet information and applications." 36 CFR § 1194.22. Section 1194.22 sets standards for web accessibility, including web accessibility for assistive technologies. (With regard to web accessibility, assistive technology generally refers to software that enables an individual with a certain disability to interact with a computer. For example a sight-impaired individual may be unable to see the screen of a computer and may need the assistance of a screen reader to interact with a computer. A screen reader is assistive technology that reads the contents of a computer screen to the user.) These standards are based in part on the World Wide Web Consortiums' (W3C) Web Accessibility Initiative's (WAI) Web Content Accessibility Guidelines 1.0 (WCAG 1.0), as well as other agency documents on web accessibility and additional recommendations of the Electronic and Information Technology Access Advisory Committee (EITAAC).

[0006] Because the award of government contracts is based in part on compliance with the Section 508 standards, it is prudent for web service providers to find a cost-efficient and effective means for implementing those standards.

SUMMARY

[0007] In one aspect, the invention provides a method and computer program product for checking an HTML document of a web page for compliance with Section 508 of the Rehabilitation Act of 1973. In another aspect, the invention provides a method for fixing and/or checking an HTML document for accessibility. The method may include, and the computer program product may implement, the steps of running a checker against an HTML document; flagging a violation of a requirement of Section 508; and fixing a section of the HTML document containing the flagged violation by modifying HTML code. In one aspect, the method may include, and the computer program product may implement, the steps of providing a user interface operable to display information about the flagged violation and query a user for input relating to the flagged violation; receiving the user input; and, using the user input, fixing the section of the HTML document. In another aspect, the method may include, and the computer program product may implement, the step of offering a recommended solution to the user and/or the step of using previously-saved user input saved at the user's request to fix the section of the HTML document. In one aspect, fixing the section of the HTML document containing the flagged violation is accomplished without input from a user.

[0008] In another aspect of the invention, the method may include, and the computer program product may implement, the step of providing a user interface operable to display information about the flagged violation and provide details as to how the violation may be manually cured and/or the step of displaying the flagged violation on a user interface. In another aspect, the method may include, and the computer program product may implement, the steps of providing a user interface operable to accept user input; and, using the user input, modifying tolerances of one or more checkers. In another aspect, the method may include, and the computer program product may implement, the steps of locating logical groups of links in the document; and determining whether there is a facility for skipping past a located, logical group of links. In one aspect, locating logical groups of links in the document may include producing a psuedo-model of a document that represents the document as a series of text and links; dividing the pseudo-model into groups of links that are separated by a predetermined length of text; and dividing each group based on ancestral differences between adjacent links. In one aspect, determining whether there is a facility for skipping past a located, logical group of links may include checking for inner-document links within each logical group of links that allow a user to skip past the logical group of links.

[0009] In another aspect of the invention, the method may include, and the computer program product may implement, the step of inserting an inner-document link permitting a user to skip past a group of links. In another aspect, the

method may include, and the computer program product may implement, the steps of identifying a located, logical group of links for which there is no facility for skipping past; searching elements near the front of the identified link group for a surrogate anchor element; and inserting an inner-document link at the beginning of the link group, where the inner-document link inserted at the beginning of the link group replaces the surrogate anchor element. In another aspect, the method may include, and the computer program product may implement, the steps of identifying, for a form field not associated with a label, a piece of text that is a candidate for the label; and associating the piece of text with the label. In another aspect, the method may include, and the computer program product may implement, the steps of identifying, for a form field not associated with a label, a piece of text that is a candidate for the label; prompting a user to select the candidate label or insert a label; and, based on the user's selection, associating the piece of text with the candidate label or the inserted label. In one aspect, identifying a piece of text that is a candidate for the label may include identifying a piece of text that is not a child of any of "a", "applet", "script", "noscript", "select", "object", "head", or "label;" and has a predetermined text length.

[**0010**] Aspects of the invention can include one or more of the following advantages. A compliance retrofitter in accordance with the invention can be used to scan the HyperText Mark-up Language (HTML) code that powers the web, identify violations with Section 508 Standards (checkers), and correct those violations by inserting the necessary corrections into the code in real time (fixers). Incorporation into the system of "tolerances" allows the checkers to be customized to individual clients and greatly enhances the value of such tools when distributed on an organization-wide level.

DESCRIPTION OF DRAWINGS

[**0011**] **FIG. 1** is an exemplary screen shot used to implement the compliance retrofitter.

[**0012**] **FIG. 2** is an exemplary screen shot used to implement tolerances in the compliance retrofitter.

[**0013**] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

A. Compliance Checking and Retrofitting Algorithms—Checkers and Fixers

[**0014**] Referring to **FIG. 1, a** compliance retrofitter employing an algorithm designed to check for and/or fix violations of 36 CFR § 1194.22 may produce a graphical user interface GUI, for example a GUI displaying screen shot **100**. A user may check compliance of a document, i.e., a particular web page or file, by entering a URL into the navigation bar **10** or by clicking on the file icon **15** and selecting a URL or entire file. The compliance retrofitter will operate on the HTML of the document and will check compliance with one or more paragraphs of § 1194.22, listed below in Table I(a), by implementing one or more "checkers" and, depending on the paragraph of § 1194.22, by requiring some form of directed manual review. The checkers implemented by the compliance retrofitter are sets of tests that are run against documents, for example files or web pages, to analyze compliance with a standard, e.g., Section 508. A list of exemplary checkers and their functionalities is provided in Table I(b).

[**0015**] One of the checkers ran in the creation of screen shot 100 was the ImageMapChecker, which checked compliance of the web page www.yahoo.com with paragraphs (a) and (f) of § 1194.22. As noted in the sixth row of Table I(b), ImageMapChecker, in part, checked the HTML code of www.yahoo.com to determine if all of the AREAS had valid "alt attributes," i.e., to determine if each AREA had associated with it an alternative textual description. (An AREA element defines a part of an image that functions as a link. Since some users do not use browsers that display images, having valid alt attributes for AREAs allows these users to navigate through the page.) Referring back to **FIG. 1**, the results created by running the checkers are presented in tree format in the history window **20**. As seen in this example, under the folder "Diagnosed Pages"**22**, the first analyzed page is displayed —www.yahoo.com **24**. Under the page www.yahoo.com **24** are the violations **26**, **36**, **46**, **56**, and **66** that were found, for example (3) AREA missing alt attribute **26**, as well as instances Line **1**: area(s) **28-33** (where instances are violations within violations). The instances Line **1**: area(s) **28-33** indicate that the compliance checker algorithm found six HTML elements that were missing alt attribute.

[**0016**] An HTML window **40** provides the HTML code for the currently selected document (www.yahoo.com **24**), and highlights the currently selected instance, in this case Line **1**: area **28**. Below HTML window **40** is fixer window **50** that displays information related to fixers, or algorithms that make changes to HTML documents or other documents to bring the documents into compliance with the standard, e.g., § 1194.22. A list of § 1194.22 fixers are listed in Table II(a) along with the relevant standard and manual fixes (if any). The functionality of each fixer is explained in Table II(b). As can be seen, the ImageMapFixer adds alt attributes to AREA elements that currently do not have valid alt attributes.

[**0017**] Referring again to **FIG. 1**, fixer window **50** has three selectable pages—fix violation **51**, fix information **52**, and violation information **53**. Fix violation page **51** is the graphical display for a fixer, showing the fix for the currently selected instance Line **1**: area **28**. The fix violation page **51** displays a description of the violation as well as instructions as to the change that needs to be implemented to cure the violation. The instructions may explain a fully automatic change that will be implemented or may explain a change that will require user input (interactive change). The fix information page **52** provides information about how the fixer will make a change to the HTML and what that change will look like. The violation information page **53** provides a description of the violation. A user can implement a fix by clicking on the fix button **63**.

[**0018**] Also provided is the option: "Add this information to my Autofix data"**54**. If the user checks the box corresponding to this option, an insertion is made into an Autofix library for later reference by the compliance retrofitter. Selecting the Autofix Violation button **61** will apply the corrections contained in the Autofix library to the current violation. A user can fix multiple pages by selecting the "Autofix" icon **62** next to the Menu Bar. Certain violations,

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.